**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 1**

****

**ĐỒ ÁN**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài:***

“Phân loại ý định người dùng cho dữ liệu trên mạng xã hội”

**Giảng viên hướng dẫn : TS. NGÔ XUÂN BÁCH**

**Sinh viên thực hiện : Vũ Thanh Đạt**

**Lớp : D12CNPM4**

**Khoá** **: 2012-2017**

**Hệ** **: Đại học Chính quy**

**Hà Nội, tháng 12 năm 2016**

**TÓM TẮT**

Sự phát triển nhanh như vũ bão của mạng xã hội ngày nay khiến cho lượng thông tin thu thập được từ thế giới mạng này ngày một lớn. Điều đó nảy sinh ra những nhu cầu cấp thiết về việc lưu trữ và khai phá thông tin. Phân loại thông tin là một trong những giải pháp hợp lý phục vụ cho nhu cầu trên nhưng thực tế là việc phân loại dữ liệu một cách thủ công là điều không tưởng, do khối lượng thông tin là quá lớn. Việc phân loại thông tin thu thập trên mạng xã hội chắc chắn phải dựa trên các nghiên cứu trước đó về phân loại văn bản chính thống, các kỹ thuật phân loại văn bản được phát triển trên toàn thế giới. Nhưng đối với dữ liệu trên mạng xã hội nói riêng không mang tính quy chuẩn về ngữ nghĩa, thì việc phân loại chúng sẽ trở nên khó khăn hơn.

Để có câu trả lời cho những vấn đề trên, đồ án “*Phân loại ý định người dùng cho dữ liệu trên mạng xã hội”* sẽ tập trung nghiên cứu về khai phá dữ liệu văn bản, các đặc trưng của dữ liệu mạng xã hội và đưa ra các phương pháp giúp giải quyết bài toán phân loại ý định người dùng cho dữ liệu trên mạng xã hội bằng tiếng Anh. Đây là bài toán có đầu vào là nội dung trạng thái mà mỗi người dùng đăng trên mạng xã hội chia sẻ về mọi thứ xung quanh. Nhiệm vụ của chúng ta là phân loại ý định người dùng thông qua dòng trạng thái đó, cụ thể hơn là xác định người dùng đó có ý định gì hay không, hay nếu có thì ý định đó về vấn đề gì: **du lịch, đồ ăn và thức uống, nghề nghiệp và giáo dục, hàng hóa và dịch vụ, sự kiện, chuyện vặt**.

Hiện nay có nhiều phương pháp khác nhau được áp dụng cho bài toán phân loại dữ liệu văn bản, trong phạm vi đồ án này, chúng tôi sử dụng phương pháp học máy dựa trên các thuật toán: Máy véc tơ hỗ trợ (SVM), Naive Bayes và K-Láng giềng gần nhất cùng với công cụ Weka trên dữ liệu gồm 2130 dòng trạng thái trên mạng xã hội Twitter, sau đó tổng hợp và đánh giá kết quả thu được. Kết quả đánh giá là tương đối khả quan, với bài toán “Trạng thái người dùng có ý định gì hay không?” có độ chính xác là 85.44%, với bài toán “Ý định cụ thể của người dùng” đạt độ chính xác là 70.50%.

**Từ khóa:** phân loại văn bản, phân loại ý định người dùng trên mạng xã hội, N-grams, máy vec tơ hỗ trợ, SVM, Naive Bayes, KNN, K-láng giềng gần nhất, word vector. Glove vector.

**LỜI CẢM ƠN**

Để có được kết quả học tập đến ngày hôm nay, em xin chân thành gửi đến các thầy cô giáo trong khoa Công Nghệ Thông Tin – Học Viện Công Nghệ Bưu Chính Viễn Thông nói riêng và các thầy cô giáo của trường Học Viện Công Nghệ Bưu Chính Viễn Thông nói chung lời cảm ơn chân thành nhất. Các thầy cô luôn luôn nhiệt huyết truyền đạt cho em những kiến thức về bộ môn và những kinh nghiệm trong thực tế.

Và đặc biệt để hoàn thành đồ án này, em xin chân thành cảm ơn thầy Ngô Xuân Bách đã tận tâm hướng dẫn em thực hiện và tìm hiểu mọi vấn đề. Nếu không có những chỉ dẫn của thầy thì rất khó để em có thể tự mình hoàn thành đồ án này. Một lần nữa em xin gửi lời cảm ơn chân thành nhất tới thầy.

Bên cạnh đó, em xin gửi lời cảm ơn đến gia đình và những người bạn đã luôn tạo điều kiện tốt nhất và luôn hỗ trợ giúp đỡ em khi gặp khó khăn.

Đồ án được thực hiện trong khoảng 3 tháng. Đây cũng là bước đầu em đi sâu tìm hiểu, nghiên cứu và thực nghiệm một đề tài của ngành công nghệ thông tin về xử lý ngôn ngữ tự nhiên. Do vậy, sẽ còn nhiều thiếu sót, em rất mong nhận được những ý kiến đóng góp của quý Thầy Cô và các bạn để em có thể hoàn thiện đồ án một cách tốt nhất.

Em xin chân thành cảm ơn!

Hà Nội, ngày tháng năm 2016

Sinh viên

Vũ Thanh Đạt

**NHẬN XÉT**

**(Của giảng viên phản biện)**

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**

**(Của Người hướng dẫn)**

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..

…………………………………………………………………………………………..…………………………………………………………………………………………..

**Điểm:** …………………….………(bằng chữ: …..…………… ….)

**Đồng ý/Không đồng ý** cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp**?**.

*…………, ngày tháng năm 20*

**CÁN BỘ - GIẢNG VIÊN HƯỚNG DẪN**

*(ký, họ tên)*

MỤC LỤC

[**DANH MỤC BẢNG** ix](#_Toc469566775)

[**DANH MỤC HÌNH VẼ** x](#_Toc469566776)

[**BẢNG DANH MỤC THUẬT NGỮ** xi](#_Toc469566777)

[**LỜI NÓI ĐẦU** xii](#_Toc469566778)

[**CHƯƠNG 1: TỔNG QUAN BÀI TOÁN PHÂN LOẠI Ý ĐỊNH NGƯỜI DÙNG MẠNG XÃ HỘI** 1](#_Toc469566779)

[**1.1.** **Tổng quan khai phá dữ liệu** 1](#_Toc469566780)

[**1.2.** **Khai phá ý định người dùng trên mạng xã hội** 3](#_Toc469566781)

[**1.3.** **Các nghiên cứu liên quan** 5](#_Toc469566782)

[**1.4.** **Đóng góp của đồ án** 6](#_Toc469566783)

[**1.5.** **Tổng kết chương** 6](#_Toc469566784)

[**CHƯƠNG 2: PHƯƠNG PHÁP PHÂN LOẠI Ý ĐỊNH NGƯỜI DÙNG MẠNG XÃ HỘI TIẾNG ANH** 7](#_Toc469566785)

[**2.1.** **Vấn đề phân loại ý định cho dòng trạng thái tiếng Anh** 7](#_Toc469566786)

[**2.2.** **Trích chọn đặc trưng** 8](#_Toc469566787)

[**2.2.1.** **Đặc trưng N-Grams** 8](#_Toc469566788)

[**a.** **Biểu diễn đặc trưng với loại UniGram** 9](#_Toc469566789)

[**b.** **Biểu diễn đặc trưng với loại BiGram** 10](#_Toc469566790)

[**c.** **Biểu diễn đặc trưng với loại TriGram** 11](#_Toc469566791)

[**2.2.2.** **Đặc trưng Word vector** 11](#_Toc469566792)

[**2.2.3.** **Đặc trưng Log count ratios** 13](#_Toc469566793)

[**2.3.** **Các thuật toán học máy** 14](#_Toc469566794)

[**2.3.1.** **Máy véc tơ hỗ trợ SVM (Support Vector Machine)** 15](#_Toc469566795)

[**a)** **Ý tưởng** 15](#_Toc469566796)

[**b)** **Cơ sở lý thuyết** 16](#_Toc469566797)

[**c)** **Các bước chính của phương pháp SVM** 18](#_Toc469566798)

[**2.3.2.** **Naïve Bayes** 18](#_Toc469566799)

[**a)** **Giới thiệu** 18](#_Toc469566800)

[**b)** **Cơ sở lý thuyết** 18](#_Toc469566801)

[**c)** **Áp dụng cho bài toán phân loại văn bản** 20](#_Toc469566802)

[**2.3.3.** **K láng giềng gần nhất (K-nearest neighbors)** 22](#_Toc469566803)

[**a)** **Giới thiệu** 22](#_Toc469566804)

[**b)** **Cơ sở lý thuyết** 22](#_Toc469566805)

[**c)** **Mô tả thuật toán** 23](#_Toc469566806)

[**d)** **Nhận xét thuật toán** 23](#_Toc469566807)

[**e)** **Áp dụng cho bài toán phân loại văn bản** 24](#_Toc469566808)

[**CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ** 27](#_Toc469566809)

[**3.1.** **Dữ liệu thực nghiệm** 27](#_Toc469566810)

[**3.1.1** **Chuẩn bị dữ liệu** 27](#_Toc469566811)

[**3.1.2.** **Tiền xử lý dữ liệu** 28](#_Toc469566812)

[**3.2.** **Thiết lập thực nghiệm** 29](#_Toc469566813)

[**3.2.1.** **Yêu cầu cho thực nghiệm** 29](#_Toc469566814)

[**3.2.2.** **Cách thức thực nghiệm** 29](#_Toc469566815)

[**3.2.3.** **Cách thức đánh giá** 29](#_Toc469566816)

[**3.3.** **Công cụ và dữ liệu thực nghiệm** 31](#_Toc469566817)

[**3.3.1.** **Giới thiệu công cụ Weka** 31](#_Toc469566818)

[**3.3.2.** **Giới thiệu chuẩn dữ liệu đầu vào cho thực nghiệm (ARFF)** 33](#_Toc469566819)

[**3.6.** **Tiến hành thực nghiệm** 35](#_Toc469566820)

[**3.3.1.** **Chuẩn bị dữ liệu cho các phương pháp thực nghiệm** 35](#_Toc469566821)

[**3.4.** **Kết quả thực nghiệm** 40](#_Toc469566822)

[**3.4.1.** **Theo các thuật toán khác nhau** 40](#_Toc469566823)

[**3.4.2.** **Theo các phương pháp lấy đặc trưng khác nhau** 42](#_Toc469566824)

[**3.4.3.** **Đề xuất phương pháp lấy đặc trưng mới** 46](#_Toc469566825)

[**3.4.4.** **Tiến hành kết hợp các phương pháp lấy đặc trưng** 52](#_Toc469566826)

[**KẾT LUẬN** 54](#_Toc469566827)

[PHỤ LỤC 56](#_Toc469566828)

[**A.** **Công cụ sử dụng** 56](#_Toc469566829)

[**B.** **Chương trình Demo** 56](#_Toc469566830)

**DANH MỤC BẢNG**

[*Bảng 3-1: Số lượng tweet của mỗi nhãn* 27](#_Toc469565873)

[*Bảng 3-2: Kết quả thực nghiệm sử dụng thuật toán SVM* 40](#_Toc469565874)

[*Bảng 3-3: Kết quả thực nghiệm sử dụng thuật toán KNN (k=1)* 41](#_Toc469565875)

[*Bảng 3-4: Kết quả thực nghiệm sử dụng thuật toán Naive Bayes* 41](#_Toc469565876)

[*Bảng 3-5: Kết quả thực nghiệm sử dụng n-grams (thuật toán SVM)* 43](#_Toc469565877)

[*Bảng 3-6: Kết quả thực nghiệm Glove Vector (thuật toán SVM)* 43](#_Toc469565878)

[*Bảng 3-7: Kết quả thực nghiệm Log count ratio sử dụng n-grams (thuật toán SVM)* 43](#_Toc469565879)

[*Bảng 3-8: Kết quả thực nghiệm đặc trưng New Naive Bayes (thuật toán SVM)* 49](#_Toc469565880)

[*Bảng 3-9: Kết quả thực nghiệm kết hợp Glove Vector và UniGram (thuật toán* *SVM).* 52](#_Toc469565881)

[*Bảng 3-10: Kết quả thực nghiệm kết hợp Glove Vector và TriGram (thuật toán SVM).* 52](#_Toc469565882)

[*Bảng 3-11 : Kết quả thực nghiệm TriGram + Glove Vector 300D + NaiveBayes Trigram (thuật toán SVM)* 53](#_Toc469565883)

[*Bảng A-1:* *Các công cụ sử dụng* 56](#_Toc469565884)

**DANH MỤC HÌNH VẼ**

[*Hình 1-1: Quy trình phát hiện tri thức [10]* 2](#_Toc469566193)

[*Hình 2-1: Mô hình giai đoạn huấn luyện [9].* 7](#_Toc469566194)

[*Hình 2-2: Mô hình giai đoạn phân lớp [9].* 8](#_Toc469566195)

[*Hình 2-3: Siêu phẳng phân chia dữ liệu học thành 2 lớp + và – với khoảng cách biên lớn nhất. [12]* 16](#_Toc469566196)

[*Hình 2-4: Minh họa bài toán phân 2 lớp bằng phương pháp SVM* [12] 17](#_Toc469566197)

[*Hình 3-1: File data.txt (\*)* 27](#_Toc469566198)

[*Hình 3-2: File datasetver2.txt* 29](#_Toc469566199)

[*Hình 3-3: Giao diện màn hình chính của Weka.* 31](#_Toc469566200)

[*Hình 3-4: Giao diện khi chọn một tập tin để thực nghiệm* 32](#_Toc469566201)

[*Hình 3-5: Giao diện khi tiến hành thực nghiệm* 32](#_Toc469566202)

[*Hình 3-6: Ví dụ minh họa một phần biểu diễn ARFF mềm dẻo* 35](#_Toc469566203)

[*Hình 3-7: File từ điển dictionary.txt* 36](#_Toc469566204)

[*Hình 3-8: Một phần tập tin định dạng dữ liệu cho Weka* 37](#_Toc469566205)

[*Hình 3-9: Một phần tập tin glove6B300d.txt* 37](#_Toc469566206)

[*Hình 3-10: Một phần file “resultGlove300D7.txt”* 38](#_Toc469566207)

[*Hình 3-11: Một phần file “resultGlove300D7.txt”* 38](#_Toc469566208)

[*Hình 3-12: File dữ liệu NaiveBayes2Categories.txt* 48](#_Toc469566209)

[*Hình 3-13: File dữ liệu NaiveBayes7Categories.txt* 49](#_Toc469566210)

[*Hình B-1: Giao diện khi khởi chạy Demo* 57](#_Toc469566211)

[*Hình B-2: Giao diện khi sử dụng lựa chọn nhập bằng tay* 58](#_Toc469566212)

[*Hình B-3: Giao diện khi chọn chức năng nhập bằng tập tin* 59](#_Toc469566213)

[*Hình B-4: Giao diện khi lựa chọn huấn luyện bằng tập dữ liệu khác* 60](#_Toc469566214)

**BẢNG DANH MỤC THUẬT NGỮ**

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Ý nghĩa |
| NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| SVM | Support Vector Machine | Máy vector hỗ trợ |
| NB | Naïve Bayes | Thuật toán Nave Bayes |
| KNN | K-Nearest Neighbors | Thuật toán K-Láng giềng gần nhất |
| ARFF | Atrribute Regation File Format | [Định dạng tập tin thuộc tính liên quan](http://www.abbreviationfinder.org/vi/acronyms/arff_attribute-relation-file-format.html) |
|  | Tweet | Tráng thái người dùng trên mạng xã hội Twitter |

# **LỜI NÓI ĐẦU**

Bài toán phân loại ý định người dùng trên mạng xã hội là một bài toán khá phức tạp. Đây là bài toán có đầu vào là dòng trạng thái được mỗi cá nhân đăng lên mạng xã hội, đưa ra các thông tin của họ về mọi thứ xung quanh và chúng ta cần phải xác định xem dòng trạng thái của họ có ý định gì hay không và cụ thể, ý định của họ khi đó là về vấn đề gì: **du lịch, đồ ăn và thức uống, nghề nghiệp và giáo dục, hàng hóa và dịch vụ, sự kiện, chuyện vặt**. Thách thức lớn nhất của bài toán ở đây là do dữ liệu đầu vào không thể đảm bảo tính quy chuẩn ngữ pháp và chính tả bởi vì đó là thói quen của mỗi người khi sử dụng mạng xã hội. Do đó để đưa ra một kết quả ý định của người dùng sẽ gặp rất nhiều khó khăn nhưng đó cũng là động lực cho nhiệm vụ phân tích ý định người dùng ở mức thực thể. Hiện nay, có rất nhiều ứng dụng cần đến việc phân tích xác định người dùng mạng xã hội như: hệ thống quảng cáo sản phẩm đến người dùng, dựa trên dòng trạng thái của họ mà đưa ra những sản phẩm, những gợi ý hợp lý nhất, v.v.

Đồ án “*Phân loại ý định người dùng cho dữ liệu trên mạng xã hội*” thực hiện khảo sát, nghiên cứu các phương pháp xây dựng hệ thống phân loại ý định được quan tâm nhất hiện nay. Từ đó đưa ra phương pháp phân loại ý định phù hợp nhất cho hệ thống phân loại ý định tiếng Anh. Những nghiên cứu trong đồ án có thể coi là tiền đề cũng như tham khảo cho các nghiên cứu tiếp theo về phân loại ý định người dùng hoàn thiện hơn cho tiếng Anh cũng như là tiếng Việt sau này.

Các thuật toán được áp dụng làm thực nghiệm dựa trên kết hợp một số đặc trưng ngôn ngữ tiếng Anh. Các đặc trưng này sẽ được biểu diễm dưới dạng vector và làm đầu vào cho các thuật toán. Kết quả thực nghiệm tốt nhất đạt được khi sử dụng thuật toán SVM trên tập đặc trưng kết hợp. Kết quả thực nghiệm cho kết quả tốt nhất với bài toán “Trạng thái người dùng có ý định gì hay không?” có độ chính xác là 85.44%, với bài toán “Ý định cụ thể của người dùng” đạt độ chính xác là 70.50%.

Nội dung đồ án chúng tôi thực hiện bao gồm 3 chương, bố cục chi tiết bao gồm:

**Chương 1: Tổng quan về bài toán phân loại ý định người dùng**

Nội dung của chương là đưa ra cái nhìn một cách tổng quan nhất về ý định người dùng trên mạng xã hội và bài toán phân loại ý định người dùng với dữ liệu mạng xã hội Twitter (tiếng Anh).

**Chương 2: Phương pháp phân loại ý định người dùng cho dữ liệu mạng xã hội Tiếng Anh**

Nội dung của chương là trình bày một số phương pháp lấy đặc trưng ngôn ngữ tiếng Anh cho bài toán phân loại ý định người dùng dựa trên dữ liệu mạng xã hội Twitter và các thuật toán học máy thông kê được sử dụng để tiến hành thực nghiệm trên đó.

**Chương 3: Thực nghiệm và đánh giá**

Trong chương 3, đồ án trình bày chi tiết về quá trình làm thực nghiệm gồm có cách thu thập dữ liệu, cách xử lý dữ liệu, chi tiết từng bước làm thực nghiệm, đánh giá kết quả thực nghiệm.

**CHƯƠNG 1: TỔNG QUAN BÀI TOÁN PHÂN LOẠI Ý ĐỊNH NGƯỜI DÙNG MẠNG XÃ HỘI**

Trong chương này, đồ án trình bày cái nhìn tổng quan về khai phá dữ liệu cũng như bài toán phân loại ý định người dùng, các cách tiếp cận bài toán, các nghiên cứu liên quan và những gì mà đồ án đã đóng góp được.

* 1. **Tổng quan khai phá dữ liệu**

Khai phá dữ liệu là một lĩnh vực khoa học mới xuất hiện, nhằm tự động hóa khai phá những thông tin, tri thức hữu ích, tiềm ẩn trong các CSDL cho các tổ chức, doanh nghiệp, v.v từ đó thúc đẩy khả năng sản xuất, kinh doanh, cạnh tranh của tổ chức, doanh nghiệp này. Hiện nay, khai phá dữ liệu được ứng dụng rộng rãi trong các lĩnh vực như: Phân tích dữ liệu hỗ trợ ra quyết định, điều trị y học, tin-sinh học, thương mại, tài chính, bảo hiểm, text mining, web mining..v.v.

Quá trình khám phá tri thức có thể phân thành các công đoạn sau:

- Trích lọc dữ liệu: Là bước tuyển chọn những tập dữ liệu cần được khai phá từ các tập dữ liệu lớn (databases, data warehouses, data repositories) ban đầu theo một số tiêu chí nhất định

- Tiền xử lý dữ liệu: Là bước làm sạch dữ liệu (xử lý dữ liệu không đầy đủ, dữ liệu nhiễu, dữ liệu không nhất quán), tổng hợp dữ liệu (nén, nhóm dữ liệu, tính tổng, xâydựng các histograms, lấy mẫu,…), rời rạc hóa dữ liệu (rời rạ hóa dựa vào histograms, entropy, phân khoảng,...). Sau bước tiền sử lý này, dữ liệu sẽ nhất quán, đầy đủ, được rút gọn và rời rạc hóa.

- Biến đổi dữ liệu: Là bước chuẩn hóa và làm mịn dữ liệu để đưa dữ liệu về dạng thuận lợi nhất nhằm phục vụ việc áp dụng các kỹ thuật khai phá ở bước sau.

- Khai phá dữ liệu: Là bước áp dụng những kỹ thuật phân tích (phần nhiều là các kỹ thuật học máy) nhằm khai thác dữ liệu, trích lọc những mẫu tin (information patterns), những mối quan hệ đặc biệt trong dữ liệu. Đây được xem là bước quan trọng và tiêu tốn thời gian nhất của toàn bộ quá trình KDD.

- Đánh giá và biểu diễn tri thức: Những mẫu thông tin và mối quan hệ trong dữ liệu đã được phát hiện ở bước khai phá dữ liệu được chuyển sang và biểu diễn ở dạng gần gũi với người sử dụng như đồ thị, cây, bảng biểu, luật,..v.v. Đồng thời bước này cũng đánh giá những tri thức khai phá được theo những tiêu chí nhất định.



*Hình 1-1: Quy trình phát hiện tri thức [10]*

Khai phá dữ liệu tuy là một lĩnh vực mới nhưng đã thu hút được sự quan thâm của rất nhiều nhà nghiên cứu, nhờ có những ứng dụng trong thực tiễn, các ứng dụng điển hình có thể liệt kê như sau:

* Phân tích dữ liệu và hỗ trợ ra quyết định.
* Điều trị trong y học: mối liên hệ giữa triệu chứng, chuẩn đoán và phương pháp điều trị.
* Phân lớp văn bản, tóm tắt văn bản và phân lớp các trang web.
* Tin sinh học: tìm kiếm, đối sánh các hệ gen và thông tin di truyền, mối liên hệ giữa một số hệ gen và một số bệnh di truyền.
* Nhận dạng.
* Tài chính và thị trường chứng khoán: phân tích tình hình tài chính và dự đoán giá cổ phiếu.
  1. **Khai phá ý định người dùng trên mạng xã hội**

Dữ liệu trên mạng xã hội hiện nay là một nguồn dữ liệu vô tận và việc quản lý cũng như khai thác được nó là một vấn đề khá khó khăn. Những người quản lý thường đặt ra văn bản “*Sở thích của người dùng là gì?”,* hoặc những nhà bán hàng sẽ đặt ra câu hỏi “*Người dùng đang cần những sản phẩm nào?”.* Phân loại ý định người dùng đã và đang trở thành một chủ đề nghiên cứu hấp dẫn trong xử lý ngôn ngữ tự nhiên và cộng đồng khai phá dữ liệu trong những năm gần đây. Một hệ thống có thể phân tích được ý định người dùng sẽ cung cấp những thông tin chính xác cho những nhà quản lý, những nhà tiếp thị mà còn cho chính người dùng bởi những tính năng hữu ích sau này.

Phân loại ý định của người dùng với mục đích là xác định được cụ thể dòng trạng thái của họ có ý định hay không, và nếu có thì ý định đó là gì. Ví dụ:

***Đầu vào:*** “*I must look like a right weirdo*”

(*Tôi trông giống như một kẻ lập dị*)

***Đầu ra:*** Không có ý định

***Đầu vào:*** “*I wanna go to Japan this summer.”*

(*Tôi muốn đi Nhật Bản vào mùa hè này*)

***Đầu ra:*** Có ý định du lịch

Phân loại ý định có thể thực hiện ở cấp câu cũng như cấp đoạn văn bản, mỗi câu sẽ được xác định là có ý định hay không có ý định, cụ thể được chia làm 7 loại ý định như sau:

**Thực phẩm & Đồ uống:** người dùng lên kế hoạch để có một số thực phẩm hoặc uống. Có thể đó cũng là sở thích ăn uống của họ và các quan điểm về các món ăn, đồ uống. Ví dụ:

“*I want pancakes , should I make some ???”*

(*Tôi muốn vài cái bánh, có ai cho tôi không??)*

**Ý định du lịch:** người dùng đăng trạng thái quan tâm đến thăm một số địa điểm, địa danh cụ thể. Ví dụ:

“*I wanna go to Japan this summer.”*

(*Tôi muốn đi Nhật Bản vào mùa hè này*).

**Hướng nghiệp & Giáo dục:** người dùng muốn có được một công việc hoặc một mong muốn một trình độ học vấn nào đó. Ví dụ:

“*I wanna be a model*”

(*Tôi muốn trở thành một người mẫu*).

**Hàng hóa & Dịch vụ ý:** người dùng quan tâm hoặc muốn có một số sản phẩm (không phải đồ ăn và thức uống) hoặc dịch vụ đáp ứng nhu cầu bản thân. Ví dụ:

“*I wanna buy an iPhone for my mother*”

(*Tôi muốn mua một chiếc iPhone cho mẹ tôi*).

**Sự kiện & Hoạt động:** người dùng muốn tham gia một sự kiện hay một hoạt động nào đó. Ví dụ:

“I might be going to the festival again later. Hopefully .”

(Tôi có thể đến lễ hội đó một lần nữa. Hy vọng là như vậy.)

**Chuyện vặt:** Thể loại này tweet những cuộc nói chuyện phiếm về thói quen hàng ngày, hay một số chuyện lặt vặt về tâm trạng. Ví dụ:

“I don't love you but I might like you a little bit”

(Tôi không yêu bạn nhưng tôi có thích bạn một chút)

**Không có ý định:** Đây được coi là những dòng tweet không cho ta bất cứ thông tin nào về ý định của người dùng.

“I must look like a right weirdo”

(Tôi trông giống như một kẻ lập dị).

Việc khai thác ý định từ nguồn dữ liệu mạng xã hội Twitter chắc chắn sẽ phải dựa vào những kết quả nghiên cứu về văn bản nói chung, về dữ liệu văn bản và các kỹ thuật xử lý đã được phát triển trước đó.

* 1. **Các nghiên cứu liên quan**
* *An Empirical Study on Sentiment Analysis for Vietnamese:* Nghiên cứu của TS. Ngô Xuân Bách, TS. Từ Minh Phương, Nguyễn Thị Duyên sử dụng phương pháp học có giám sát, sử dụng các tập đặc trưng trong câu để phân loại câu.
* *Khai phá dữ liệu từ các mạng xã hội để khảo sát ý kiến đánh giá các địa điểm du lịch tại Đà Nẵng:* Nghiên cứu của Phùng Đức Đoàn và các cộng sự sử dụng công cụ SVM cùng với các tập đặc trưng từ quan trọng n-grams được kết quả cao nhất là 74.79%.
* *Nghiên cứu của J.Bollen về Dự báo thị trường chứng khoán dựa trênTwitter:* Công trình này đã nêu và giải quyết bài toán dự báo thị trường chứng khoándựa trên các thông tin mà người dùng để lại trên Tweeter, dùng kĩ thuật khai phá dữ liệu để lấy tất cả các thông tin này sau đó dùng phương pháp phân loại văn bản SVM kết hợp với kNN để phân tích cảm xúc của người dùng từ đó đưa ra những dự đoán về giá chứng khoán trên thị trường.
* *Nghiên cứu của Jinpeng Wang và các cộng sự về phương pháp học máy bán giám sát, phân loại ý định cho dữ liệu trên Twitter.* Công trình nghiên cứu này dựa trên phương pháp học máy bán giám sát, sử dụng các tập đặc trưng để phân loại ý định với kết quả cao nhất là 58.64%.
* *Phát hiện cộng đồng sử dụng thuật toán CONGA và khai phá quan điểm cộng đồng:* Công trình đã trình bày cách phát hiện cộng đồng dựa trên thuật toán CONGA do SteveGrogery đề xuất năm 2007 và sử dụng kỹ thuật học máy xác suất Naïve Bayes để phân lớp quan điểm cộng đồng.
  1. **Đóng góp của đồ án**

Đồ án đã góp phần mang lại những đóng góp cơ bản sau:

* Nghiên cứu bài toán Phân loại ý định người dùng cho dữ liệu trên mạng xã hội.
* Nghiên cứu phương pháp học máy cho bài toán.
* Đưa ra các giải pháp nhằm phân loại ý định người dùng trên mạng xã hội.
* Đánh giá thực nghiệm cho dữ liệu trên mạng xã hội Twitter.
* Xây dựng hệ thống thực nghiệm cho dữ liệu tiếng Anh.
  1. **Tổng kết chương**

Trong chương này chúng tôi giới thiệu khái quát bài toán, các vấn đề liên quan và cách phân loại ý định người dùng. Trong chương tiếp theo, chúng tôi sẽ làm rõ các vấn đề của bài toán phân loại ý định người dùng và các phương pháp để phân loại ý định người dùng.

**CHƯƠNG 2: PHƯƠNG PHÁP PHÂN LOẠI Ý ĐỊNH NGƯỜI DÙNG MẠNG XÃ HỘI TIẾNG ANH**

Trong chương này, đồ án trình bày một số phương pháp lấy đặc trưng cho phân loại văn bản tiếng Anh gồm có đặc trưng từ vựng, N-grams, đặc trưng Word Vector, đặc trưng cú pháp. Ngoài ra, đồ án còn trình bày các thuật toán được sử dụng khi làm thực nghiệm gồm có Máy véc tơ hỗ trợ, Naïve Bayes và K-láng giềng gần nhất.

* 1. **Vấn đề phân loại ý định cho dòng trạng thái tiếng Anh**

Chúng ta có một tập các nội dung trạng thái của người dùng trên mạng xã hội, mục đích của chúng ta là xác định nội dung đó có ý định hay không, và nếu có ý định thì ý định đó là gì: du lịch, thực phẩm và đồ uống, học tập và sự nghiệp, sự kiện và hoạt động, hàng hóa và dịch vụ, chuyện phiếm.

Đầu vào: một đoạn gồm 1 hay nhiều câu nói tiếng Anh.

Đầu ra: nhãn phân loại ý định cho nội dung đó của người dùng.

**Ví dụ:** một người dùng cập nhật trạng thái của mình trên mạng xã hội: “*I really look forward to Egypt next week ! Its gonna be awesome.”*(Tôi thật sự chờ đợi chuyến đi Ai Cập tuần tới. Nó sẽ rất tuyệt vời) thì sẽ có nhãn ý định là “travel” (du lịch).

Phân loại ý định gồm 2 giai đoạn: giai đoạn huấn luyện và giai đoạn phân lớp.

* Giai đoạn huấn luyện:

Vector hóa

Dữ liệu huấn luyện

Tiền xử lý

Mô hình phân loại

Thuật toán huấn luyện

Trích chọn đặc trưng

*Hình 2-1: Mô hình giai đoạn huấn luyện [9].*

Giai đoạn huấn luyện nhận đầu vào là tập ngữ liệu huấn luyện gồm các nội dung dưới dạng văn bản đã được gán nhãn, sau khi xử lý tập ngữ liệu và áp dụng các thuật toán huấn luyện sẽ cho ra đầu ra là một mô hình phân loại.

* Giai đoạn phân lớp:

Trích chọn đặc trưng

Thuật toán học máy

Đầu vào

Đầu ra

Nội dung - N-grams - SVM Loại ý định

tiếng Anh - Word Vector - NB

- Log-count ratios - KNN

*Hình 2-2: Mô hình giai đoạn phân lớp [9].*

Giai đoạn phân lớp nhận đầu vào là nội dung trạng thái của người dùng dưới dạng ngôn ngữ tự nhiên, sau quá trình xử lý và áp dụng mô hình phân loại sẽ cho ra nhãn phân loại của văn bản đầu vào. Từ đó chúng ta có kiến trúc hệ thống phân loại ý định người dùng tiếng Anh gồm có 2 thành phần chính:

* **Bộ trích chọn đặc trưng:** trích xuất ra các đặc trưng từ nội dung trạng thái của người dùng như đặc trưng từ vựng N-grams, đặc trưng Word vector, đặc trưng Log-count ratios
* **Bộ phân loại:** có thể sử dụng các thuật toán khác nhau, trong phạm vi đồ án chúng tôi sử dụng 3 thuật toán học máy gồm: Máy véc tơ hỗ trợ (SVM), Naïve Bayes (NB) và K-láng giềng gần nhất (KNN).

Sau đây chúng tôi sẽ giới thiệu chi tiết 2 thành phần quan trọng nhất trong bài toán phân loại ý định người dùng nói riêng và bài toán phân loại văn bản nói chúng.

* 1. **Trích chọn đặc trưng**

Trích chọn đặc trưng có ý nghĩa quan trọng, ảnh hưởng trực tiếp đến kết quả phân lớp. Các loại đặc trưng chính thường được sử dụng là tập từ N-grams. Ngoài ra trong đồ án này, chúng tôi còn sử dụng phương pháp biểu diễn từ bằng Vector và lấy đặc trưng Log-count ratios.

* + 1. **Đặc trưng N-Grams**

Gram ở đây là đơn vị nhỏ nhất – hay nói cách khác trong câu thì nó chỉ bao gồm một từ. Một cụm n-grams là một dãy con gồm n-yếu tố liên tiếp nhau của một dãy các yếu tố cho trước. Yếu tố ở đây có thể là âm tiết, chữ cái hoặc từ vựng v.v. Nhãn từ loại và các n-grams thường được thu thập từ một văn bản hoặc lời nói. Số phần tử trong một n-grams được gọi là bậc của n-grams, thông thường n-grams có bậc từ 1 tới 4. N-gram bậc 1 được gọi là unigram, bậc 2 được gọi là bigram, bậc 3 được gọi là trigram, bậc 4 được gọi là quadrigram,v.v. N-grams được dùng để ước lượng xác suất xuất hiện của một yếu tố dựa vào các yếu tố xung quanh nó trong câu. Do đó, n-grams có thể áp dụng cho các hệ thống tách từ, gán nhãn từ loại, phát hiện lỗi chú giải từ loại, v.v.

Ví dụ:

**Câu 1 :** “*I really wanna go to Japan this summer*”

(Tôi thực sự muốn đi Nhật Bản vào mùa hè này)

* UniGram: I, really, wanna, go, to, Japan, this, summer.
* BiGram: I really, really wanna, wanna go, go to, to Japan, Japan this, this summer.
* TriGram: I really wanna, really wanna go, wanna go to, go to Japan, to Japan this, Japan this summer.

Để có thể sử dụng được các thuật toán học máy, việc biểu diễn các đặc trưng theo đúng chuẩn sẽ là một bước rất quan trọng sau khi chúng tôi đã lựa chọn được các đặc trưng.

Ví dụ:

**Câu 1:** “*I really wanna go to Japan this summer*”

(Tôi thực sự muốn đi Nhật Bản vào mùa hè này)

**Câu 2:** “*I need a car. Anybody selling a car ?”*

(Tôi cần một chiếc xe ô tô. Có ai bán ô tô không?)

1. **Biểu diễn đặc trưng với loại UniGram**

Xây dựng từ điển:

{

1: I, 2: really, 3: wanna, 4: go, 5: to, 6: Japan, 7: this, 8: summer, 9: need, 10: a, 11: car, 12: anybody, 13: selling

}

Biểu diễn 2 câu trên dưới dạng vector đặc trưng, mỗi phần tử của vector có dạng: <*vị trí của từ trong từ điển> : <số lần xuất hiện của từ trong câu>*, dựa vào chỉ số trong từ điển ta có 2 vector:

Câu 1: [1:1, 2:1, 3:1, 4:1, 5:1, 6:1, 7:1, 8:1, 9:0, 10:0, 11:0, 12:0, 13:0]

Câu 2: [1:1, 2:0, 3:0, 4:0, 5:0, 6:0, 7:0, 8:0, 9:1, 10:2, 11:2, 12:1, 13:1]

Khi ở bộ dữ liệu thực sự, số lượng từ trong từ điển là vô cùng lớn, khi đó độ dài của vector (tương ứng với độ lớn của từ điển) là rất lớn, việc xử lý sẽ vô cùng khó khăn. Mà ở đây, chúng tôi có thể để ý, những từ trong từ điển không xuất hiện trong câu (<số lần xuất hiện trong câu> bằng 0) thì hầu không có ý nghĩa trong việc xử lý thuật toán, vì thế chúng tôi sẽ loại bỏ chúng đi trong vector, chỉ giữ lại những từ xuất hiện trong câu. Cụ thể, 2 câu trên sau khi được rút gọn sẽ còn lại:

Câu 1: [1:1, 2:1, 3:1, 4:1, 5:1, 6:1, 7:1, 8:1]

Câu 2: [1:1, 9:1, 10:2, 11:2, 12:1, 13:1]

1. **Biểu diễn đặc trưng với loại BiGram**

Xây dựng từ điển:Ở đây, việc xây dựng từ điển sẽ bao gồm các UniGram (gồm 2 từ liên tiếp nhau), cụ thể:

{

1: I, 2: really, 3: wanna, 4: go, 5: to, 6: Japan, 7: this, 8: summer, 9: need, 10: a, 11: car, 12: anybody, 13: selling, 14: I\_really, 15:really\_wanna, 16:wanna\_go, 17:go\_to, 18:to\_Japan, 19:Japan\_this, 20:this\_summer, 21:I\_need, 22:need\_a, 23:a\_car, 24:car\_anybody, 25:anybody\_selling, 26:selling\_a

}

Biểu diễn câu trên dưới dạng vector đặc trưng, mỗi phần tử của vector có dạng: <*vị trí của từ trong từ điển> : <số lần xuất hiện của từ trong câu>*, chỉ tính các từ xuất hiện trong câu đang xét, ta có vector:

[1:1, 2:1, 3:1, 4:1, 5:1, 6:1, 7:1, 8:1,14:1, 15:!, 16:1, 17:1,18:1, 19:1, 20:1]

1. **Biểu diễn đặc trưng với loại TriGram**

Xây dựng từ điển: Ở đây, việc xây dựng từ điển sẽ bao gồm các TriGram (gồm 3 từ liên tiếp nhau), cụ thể:

{

1: I, 2: really, 3: wanna, 4: go, 5: to, 6: Japan, 7: this, 8: summer, 9: need, 10: a, 11: car, 12: anybody, 13: selling, 14: I\_really, 15:really\_wanna, 16:wanna\_go, 17:go\_to, 18:to\_Japan, 19:Japan\_this, 20:this\_summer, 21:I\_need, 22:need\_a, 23:a\_car, 24:car\_anybody, 25:anybody\_selling, 26:selling\_a, 27:I\_really\_wanna, 28:really\_wanna\_go, 29:wanna\_go\_to, 30:go\_to\_Japan, 31:to\_Japan\_this, 32:Japan\_this\_summer, 33:I\_need\_a, 34:need\_a\_car, 35: a\_car\_anybody, 36:car\_anybody\_selling, 37:anybody\_selling\_a, 38: selling\_a\_car

}

Biểu diễn câu trên dưới dạng vector đặc trưng, mỗi phần tử của vector có dạng: <*vị trí của từ trong từ điển> : <số lần xuất hiện của từ trong câu>*, chỉ tính các từ xuất hiện trong câu đang xét, ta có vector:

[1:1, 2:1, 3:1, 4:1, 5:1, 6:1, 7:1, 8:1,14:1, 15:!, 16:1, 17:1,18:1, 19:1, 20:1, 27:`, 28:1, 29:1, 30:1, 31:1, 32:1]

* + 1. **Đặc trưng Word vector**

Về cơ bản, **Word vector** chỉ là một vector trọng số. Ví dụ, 1-of-N sẽ mã hoá các từ trong từ điển thành một vector có chiều dài N (tổng số lượng các từ trong từ điển). Trong đó, có một phần tử mang giá trị 1 tương ứng với thứ tự của từ đó trong từ điển (ta có thể sắp từ điển tăng dần a-z, A-Z, 0-9) và các phần tử khác đều mang giá trị 0. Giả sử từ điển của chúng ta chỉ có 5 từ: King, Queen, Man, Woman, và Child. Ta có thể biểu diễn các từ như bên dưới:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| King | 1 | 0 | 0 | 0 | 0 |
| Queen | 0 | 1 | 0 | 0 | 0 |
| Man | 0 | 0 | 1 | 0 | 0 |
| Woman | 0 | 0 | 0 | 1 | 0 |
| Child | 0 | 0 | 0 | 0 | 1 |

Với cách mã hoá đơn giản như vậy, ta không thu được nhiều ý nghĩa trong việc so sánh các từ với nhau ngoại trừ so sánh bằng. Với ý tưởng biểu diễn từ thành một vector như vậy, Glove vector [16] biểu diễn các từ dưới dạng một phân bố quan hệ với các từ còn lại (*distributed* representation). Giả sử ta có một vector có số chiều N. Khi đó, mỗi từ được biểu diễn bằng một vector có các phần tử mang giá trị là phân bố quan hệ của từ này đối với các từ khác trong từ điển. Glove [16] là một thuật toán học không giám sát để biểu diễn cho vector các từ. Huấn luyện được thực hiện trên số liệu thống kê từ các từ đồng xảy tổng hợp từ corpus, và kết quả biểu diễn là không gian vector từ N chiều.

**Ví dụ:**

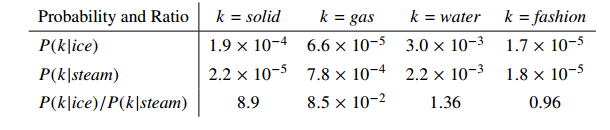
Câu: "*King is to queen as man is to woman*" được mã hóa trong không gian vector với một phương pháp đủ tốt sẽ xấp xỉ bằng phương trình vector king - queen = man - woman.

Chương trình đánh giá này thuận lợi cho mô hình sản sinh chiều của ý nghĩa, qua đó nắm bắt được ý tưởng multi-clustering của biểu diễn phân phối.

Số lần xuất hiện của từ - từ đồng xảy ra được biểu thị bởi ma trận X, trong đó  biểu diễn số lần từ  xảy ra trong ngữ cảnh chứa từ .

Khi đó: là số lần bất kỳ từ nào xuất hiện trong ngữ cảnh chứa từ .

Và  là xác suất của từ xuất hiện trong ngữ cảnh chứa từ .

**Ví dụ:** Xác suất đồng xảy ra với các từ ice và steam với những từ được chọn trong corpus. 

Bắt đầu với một ví dụ đơn giản mà giới thiệu cách khía cạnh nhất định của ý nghĩa có thể được chiết xuất trực tiếp từ xác suất đồng xảy ra. Xem xét hai chữ i và j mà biểu lộ một khía cạnh cụ thể. Giả sử đang xét khái niệm về giai đoạn nhiệt động lực học, lấy i = ice và j = steam. Các mối quan hệ của những từ này có thể được kiểm tra bằng cách nghiên cứu các tỷ lệ xác suất đồng xảy ra với các từ khác nhau, k. Đối với những từ k liên quan đến ice nhưng không liên quan đến steam, k = rắn, dự kiến tỷ lệ là rất lớn. Tương tự như vậy, các từ liên quan đến steam nhưng không liên quan đến ice, nói k = gas, tỷ lệ phải nhỏ.

Đối với những từ k = water hay fashion, một trong hai liên quan đến cả hai băng và hơi nước, nên tỷ lệ gần một. So với các xác suất ban đầu, tỷ lệ này có khả năng tốt hơn để phân biệt có liên quan từ (solid và gas) từ những từ không liên quan (water và fashion) và nó cũng có thể tốt hơn để phân biệt giữa hai từ có liên quan. Khi đó để biểu diễn câu s chứa các từ w1,w2,..,w|s| với |s| là độ dài câu, ta cộng tương ứng các Vector của từng từ w1,w2,..,w|s|.

* + 1. **Đặc trưng Log count ratios**

Là đặc trưng dựa vào tỉ lệ của từ thuộc câu có nhãn là Không có ý định so với từ thuộc câu có nhãn là Có ý định. Phương pháp Naive Bayes log-count ratios đã được chứng minh là có hiệu quả và ổn định trên dữ liệu thuộc nhiều loại khác nhau. Sau đây là cách mô tả để tính toán tỉ lệ Naive Bayes log-count ratios từ dữ liệu đầu vào [1].

Ta gọi là các vector đặc trưng tương ứng với câu  với nhãn .

Để tăng thêm độ chính xác thì chúng ta cần gia tăng thêm mối liên hệ giữa các từ trong câu và nhãn loại của câu đó. Phương pháp lấy đặc trưng này có thể được trình bày như sau, giả sử ta có 2 nhãn là 1 và -1:

* Đặt 2 vector  và có độ dài đúng bằng độ dài của vector đặc trưng.
* dicNon là từ điển từ của các câu có nhãn là “non-intent”.
* dicUnnon là từ điển từ của các câu có nhãn là “unnon-intent”.

Ở đây xây dựng 2 ma trận  và có độ dài bằng độ dài của từ điển và gán như sau:

*  nếu từ thứ  trong từ điển xuất hiện trong câu có nhãn -1.
*  nếu từ thứ  không trong từ điển xuất hiện trong câu có nhãn -1.
*  nếu từ thứ  trong từ điển xuất hiện trong câu có nhãn 1.
*  nếu từ thứ  không trong từ điển xuất hiện trong câu có nhãn 1.

Sau đó tiến hành tính lại 2 vector và theo công thức:

 và  với  ,  là độ dài của từ điển.

Khi đó, vector  được tính theo công thức sau:



Như vậy vector đặc trưng sẽ được tính lại theo công thức:



* 1. **Các thuật toán học máy**

Có nhiều thuật toán khác nhau cho phân lớp như Naïve Bayes, K-láng giềng gần nhất, cây quyết định (Decision Tree), Máy véc tơ hỗ trợ (Support Vector Machine), Mạng lọc thưa (Sparse Network of Winnows - SNoW), Mô hình Entropy cực đại, v.v. Tuy nhiên phần tiếp theo của đồ án, chúng tôi chỉ trình bày 3 thuật toán là Máy véc tơ hỗ trợ, Naïve Bayes và K-láng giềng gần nhất, đây là các thuật toán được sử dụng nhiều trong phân loại văn bản và cũng sẽ được sử dụng trong phần thực nghiệm ở chương 3.

* + 1. **Máy véc tơ hỗ trợ SVM (Support Vector Machine)**

Support Vector Machine [12] viết tắt là SVM, là một phương pháp trong thống kê và khoa học máy tính cho một tập hợp các phương pháp học có giám sát liên quan đến nhau để phân loại và phân tích hồi quy. SVM dạng chuẩn nhận dữ liệu vào và phân loại chúng vào 2 lớp khác nhau.

1. **Ý tưởng**

Cho trước một tập huấn luyện, được biểu diễn trong không gian vector, trong đó mỗi tài liệu là một điểm, phương pháp này tìm ra một siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt tương ứng là lớp + và lớp -. Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này. Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, đồng thời việc phân loại càng chính xác.

Mục đích của phương pháp SVM là tìm được khoảng cách biên lớn nhất, điều này được minh họa như sau:



*Hình 2-3: Siêu phẳng phân chia dữ liệu học thành 2 lớp + và – với khoảng cách biên lớn nhất. [12]*

1. **Cơ sở lý thuyết**

SVM thực chất là một bài toán tối ưu, mục tiêu của thuật toán này là tìm được một không gian F và siêu phẳng quyết định f trên F sao cho sai số phân loại là thấp nhất.

Cho tập mẫu (x1, y1), (x2, y2), … (xf, yf)} với xi ∈ Rn, thuộc vào hai lớp nhãn: yi ∈ {-1,1} là nhãn lớp tương ứng của các xi (-1 biểu thị lớp I, 1 biểu thị lớp II).

Ta có, phương trình siêu phẳng chứa vectơ trong không gian:

Như vậy, f(Xi) biểu diễn sự phân lớp của Xi vào hai lớp như đã nêu. Ta nói yi = +1 nếu Xi thuộc lớp I và yi = -1 nếu Xi thuộc lớp II. Khi đó, để có siêu phẳng f ta sẽ phải giải bài toán sau: Tìm min w với W thỏa mãn điều kiện sau:

Bài toán SVM có thể giải bằng kỹ thuật sử dụng toán tử Lagrange để biến đổi về thành dạng đẳng thức. Một đặc điểm thú vị của SVM là mặt phẳng quyết định chỉ phụ thuộc các Support Vector và nó có khoảng cách đến mặt phẳng quyết định là . Cho dù các điểm khác bị xóa đi thì thuật toán vẫn cho kết quả giống như ban đầu. Đây chính là điểm nổi bật của phương pháp SVM so với các phương pháp khác vì tất cả các dữ liệu trong tập huấn luyện đều được dùng để tối ưu hóa kết quả.

**Bài toán phân 2 lớp với SVM**

Bài toán đặt ra là: Xác định hàm phân lớp để phân lớp các mẫu trong tương lai, nghĩa là với một mẫu dữ liệu mới xi thì cần phải xác định xi được phân vào lớp +1 hay lớp -1

*Hình 2-4: Minh họa bài toán phân 2 lớp bằng phương pháp SVM* [12]

Để xác định hàm phân lớp dựa trên phương pháp SVM, ta sẽ tiến hành tìm hai siêu phẳng song song sao cho khoảng cách y giữa chúng là lớn nhất có thể để phân tách hai lớp này ra làm hai phía. Hàm phân tách tương ứng với phương trình siêu phẳng nằm giữa hai siêu phẳng tìm được.

Các điểm mà nằm trên hai siêu phẳng phân tách được gọi là các Support Vector. Các điểm này sẽ quyết định đến hàm phân tách dữ liệu.

**Bài toán nhiều phân lớp với SVM**

Để phân nhiều lớp thì kỹ thuật SVM nguyên thủy sẽ chia không gian dữ liệu thành 2 phần và quá trình này lặp lại nhiều lần. Khi đó hàm quyết định phân dữ liệu vào lớp thứ i của tập n, 2-lớp sẽ là:

Những phần tử x là support vector sẽ thỏa điều kiện:

Như vậy, bài toán phân nhiều lớp sử dụng phương pháp SVM hoàn toàn có thể thực hiện giống như bài toán 2 lớp. Bằng cách sử dụng cách thức “một - đối - một”. Giả sử bài toán cần phân loại có k lớp (k > 2), chiến lược "một-đối-một”sẽ tiến hành lần phân lớp nhị phân sử dụng phương pháp SVM. Mỗi lớp sẽ tiến hành phân tách với k-1 lớp còn lại để xác định k-1 hàm phân tách dựa vào bài toán phân hai lớp bằng phương pháp SVM.

1. **Các bước chính của phương pháp SVM**

* Phương pháp SVM yêu cầu dữ liệu được biểu diễn như các vector của các số thực. Như vậy nếu đầu vào chưa phải là số thì ta cần phải tìm cách chuyển chúng về dạng số của SVM.
* Tiền xử lý dữ liệu: Thực hiện biến đổi dữ liệu phù hợp cho quá trình tính toán, tránh các số quá lớn mô tả các thuộc tính. Thường nên co giãn (scaling) dữ liệu để chuyển về đoạn [-1, 1] hoặc [0, 1].
* Chọn hàm hạt nhân: Lựa chọn hàm hạt nhân phù hợp tương ứng cho từng bài toán cụ thể để đạt được độ chính xác cao trong quá trình phân lớp.
* Thực hiện việc kiểm tra chéo để xác định các tham số cho ứng đụng. Điều này cũng quyết định đến tính chính xác của quá trình phân lớp.
* Sử dụng các tham số cho việc huấn luyện với tập mẫu. Trong quá trình huấn luyện sẽ sử dụng thuật toán tối ưu hóa khoảng cách giữa các siêu phẳng trong quá trình phân lớp, xác định hàm phân lớp trong không gian đặc trưng nhờ việc ánh xạ dữ liệu vào không gian đặc trưng bằng cách mô tả hạt nhân, giải quyết cho cả hai trường hợp dữ liệu là phân tách và không phân tách tuyến tính trong không gian đặc trưng.
* Kiểm thử tập dữ liệu Test.
  + 1. **Naïve Bayes**

1. **Giới thiệu**

Naïve Bayes (NB) [12] là phương pháp phân loại có giám sát dựa vào xác suất được sử dụng rộng rãi trong lĩnh vực máy học [Mitchell, 1996] [Joachims, 1997] [Jason, 2001], được sử dụng lần đầu tiên trong lĩnh vực phân loại bởi Maron vào năm 1961 [Maron, 1961] sau đó trở nên phổ biến dùng trong nhiều lĩnh vực như trong các công cụ tìm kiếm [Rijsbergen et al, 1970], các bộ lọc email [Sahami et al, 1998] v.v.

1. **Cơ sở lý thuyết**

Thuật toán Naïve Bayes dựa trên định lý Bayes được phát biểu như sau:

Áp dụng trong bài toán phân loại, các dữ kiện gồm có:

* D: tập dữ liệu huấn luyện đã được vector hóa dưới dạng .
* Ci: phân lớp i, với i = {1,2,…,m}.
* Các thuộc tính độc lập điều kiện đôi một với nhau.

Theo định lý Bayes:

Theo tính chất độc lập điều kiện:

Trong đó:

* là xác suất thuộc phân lớp i khi biết trước mẫu X.
* xác suất là phân lớp i.
* xác suất thuộc tính thứ k mang giá trị xk khi đã biết X thuộc phân lớp i.

Các bước thực hiện thuật toán Naïve Bayes:

Bước 1: Huấn luyện Naïve Bayes (dựa vào tập dữ liệu), tính và .

Bước 2: Phân lớp , ta cần tính xác suất thuộc từng phân lớp khi đã biết trước Xnew. Xnew được gán vào lớp có xác suất lớn nhất theo công thức:

1. **Áp dụng cho bài toán phân loại văn bản**

*Ý tưởng:*

Ý tưởng cơ bản của cách tiếp cận Naïve Bayes là sử dụng xác suất có điều kiện giữa từ và chủ đề để dự đoán xác suất chủ đề của một văn bản cần phân loại. Điểm quan trọng của phương pháp này chính là ở chỗ giả định rằng sự xuất hiện của tất cả các từ trong văn bản đều độc lập với nhau. Giả định đó làm cho việc tính toán NB hiệu quả và nhanh chóng hơn các phương pháp khác vì không sử dụng việc kết hợp các từ để đưa ra phán đoán chủ đề. Kết quả dự đoán bị ảnh hưởng bởi kích thước tập dữ liệu, chất lượng của không gian đặc trưng.

*Hướng dẫn cài đặt:*

Mô tả vector đặc trưng của văn bản: là vector có số chiều là số đặc trưng trong toàn tập dữ liệu, các đặc trưng này đôi một khác nhau. Nếu văn bản có chứa đặc trưng đó sẽ có giá trị 1, ngược lại là 0.

Thuật toán gồm 2 giai đoạn huấn luyện và phân lớp:

* **Huấn luyện: tính và**

Đầu vào:

* Các vector đặc trưng của văn bản trong tập huấn luyện (Ma trận MxN, với M là số vector đặc trưng trong tập huấn luyện, N là số đặc trưng của vector).
* Tập nhãn/lớp cho từng vector đặc trưng của tập huấn luyện.

Đầu ra:

* Các giá trị xác suất và .

Công thức tính đã làm trơn Laplace:

Trong đó:

* |docsi|: số văn bản của tập huấn luyện thuộc phân lớp i.
* |total docs|: số văn bản trong tập huấn luyện.
* M: số phân lớp

Cài đặt:

* Khởi tạo mảng A, B có kích thước m.
* Duyệt qua các văn bản trong tập dữ liệu, đếm số văn bản trong mỗi phân lớp lưu vào A.
* Tính xác suất cho từng phân lớp theo công thức trên và lưu vào mảng B.

Công thức tính đã làm trơn Laplace:

Trong đó:

* : Số văn bản trong trong phân lớp i có đặc trưng thứ k mang giá trị xk. (hay số văn bản trong lớp i, có xuất hiện/không xuất hiện đặc trưng k)
* : Số văn bản của tập huấn luyện thuộc phân lớp i.
* Số giá trị có thể có của đặc trưng thứ k

Cài đặt:

* Với vector đặc trưng như mô tả bên trên, dk ở đây mang giá trị là 2, tương ứng với xuất hiện và không xuất hiện. Do chỉ có 2 giá trị, ta có thể tính nhanh xác suất không xuất hiện theo công thức .
* Khởi tạo mảng 3 chiều C, chiều 1 có kích thước là m (số phân lớp), chiều 2 có kích thước là N (số đặc trưng), chiều 3 có kích là 2 (dk) để lưu các giá trị .
* Duyệt qua các văn bản trong tập dữ liệu, tiến hành thống kê các chỉ số cần thiết để tính xác suất theo công thức trên và lưu vào mảng C.
* **Phân lớp**

Đầu vào:

* Vector đặc trưng của văn bản cần phân lớp
* Các giá trị xác suất tính và .

Đầu ra:

* Giá trị xác suất thuộc phân lớp I khi biết trước mẫu X

Dựa vào vector đặc trưng của văn bản cần phân lớp, áp dụng công thức trên tính xác suất thuộc từng phân lớp cho văn bản, và chọn ra lớp có xác suất cao nhất.

* + 1. **K láng giềng gần nhất (K-nearest neighbors)**

1. **Giới thiệu**

K-Nearest Neighbors (KNN) [12] được sử dụng rất phổ biến trong lĩnh vực Data Mining. KNN là phương pháp để phân lớp các đối tượng dựa vào khoảng cách gần nhất giữa đối tượng cần xếp lớp với tất cả các đối tượng trong Training Data.

Một đối tượng được phân lớp dựa vào k láng giềng của nó. K là số nguyên dương được xác định trước khi thực hiện thuật toán. Người ta thường dùng khoảng cách Euclide để tính khoảng cách giữa các đối tượng.

1. **Cơ sở lý thuyết**

Bộ phân lớp dựa trên thuật toán K người láng giềng gần nhất là một bộ phân lớp dựa trên bộ nhớ, đơn giản vì nó được xây dựng bằng cách lưu trữ tất cả các đối tượng trong tập huấn luyện. Để phân lớp cho một điểm dữ liệu mới x, trước hết bộ phân lớp sẽ tính khoảng cách từ điểm dữ liệu trong tập huấn luyện. Qua đó tìm được tập N(x, D, k) gồm k điểm dữ liệu mẫu có khoảng cách đến x là gần nhất.

Ví dụ nếu các dữ liệu mẫu được biểu diễn bởi không gian vector thì chúng tôi có thể sử dụng khoảng cách Euclide để tính khoảng cách giữa các điểm dữ liệu với nhau. Sau khi xác định được tập N(x, D, k), bộ phân lớp sẽ gán nhãn cho điểm dữ liệu x bằng lớp chiếm đại đa số trong tập N(x, D, k). Mặc dù rất đơn giản, nhưng thuật toán K người láng giềng gần nhất đã cho kết quả tốt trong nhiều ứng dụng thực tế.

Để áp dụng thuật toán K người láng giềng vào tài liệu văn bản, chúng tôi sử dụng hàm tính trọng số cho mỗi lớp theo biểu thức:

Trong đó Nc(x, D, k) là tập con chỉ chứa các đối tượng thuộc lớp c của tập N(x, D, k). Khi đó tài liệu x sẽ được phân vào lớp c0 nếu:

Phương pháp K người láng giềng gần nhất là một phương pháp đơn giản. Tuy nhiên, thuật toán này ổn định và sai sót thấp khi số văn bản trong tập văn bản láng giềng phải lớn.

1. **Mô tả thuật toán**

* Xác định giá trị tham số K (số láng giềng gần nhất).
* Tính khoảng cách giữa đối tượng cần phân lớp với tất cả các đối tượng trong training data (thường sử dụng khoảng cách Euclide, Cosine…).
* Sắp xếp khoảng cách theo thứ tự tăng dần và xác định k láng giềng gần nhất với đối tượng cần phần lớp.
* Lấy tất cả các lớp của k láng giềng gần nhất đã xác định.

Dựa vào phần lớn lớp của láng giềng gần nhất để xác định lớp cho đối tượng

1. **Nhận xét thuật toán**

* Rất đơn giản, không có quá trình học
* Khi phân loại mất nhiều thời gian, do quá trình tìm kiếm k dữ liệu lân cận, sau đó phân loại dựa trên majority vote (hồi quy dựa trên giá trị trung bình)
* Kết quả phụ thuộc vài việc chọn khoảng cách sử dụng
* Có thể làm việc trên nhiều loại dữ liệu khác nhau
* Giải quyết các vấn đề về phân loại, hồi quy, gom nhóm,...
* Cho kết quả tốt, tuy nhiên độ phức tạp của quá trình phân loại khá lớn
* Được ứng dụng thành công trong hầu hết các lãnh vực tìm kiếm thông tin, nhận dạng, phân tích dữ liệu, etc.

1. **Áp dụng cho bài toán phân loại văn bản**

Ý tưởng:

Khi cần phân loại một văn bản mới, thuật toán sẽ tính khoảng cách (khoảng cách Euclide, Cosine…) của tất cả các văn bản trong tập huấn luyện đến văn bản này để tìm ra k văn bản gần nhất (gọi là k “láng giềng”), sau đó dùng các khoảng cách này đánh trọng số cho tất cả chủ đề. Trọng số của một chủ đề chính là tổng tất cả các văn bản trong k láng giềng có cùng chủ đề, chủ đề nào không xuất hiện trong k láng giềng sẽ có trọng số bằng 0. Sau đó các chủ đề sẽ được sắp xếp theo mức độ giảm dần và các chủ đề có trọng số cao sẽ được chọn là chủ đề của văn bản cần phân loại.

Khoảng cách giữa 2 văn bản chính là độ tương tự giữa 2 văn bản đó, 2 văn bản có giá trị độ tương tự càng lớn thì khoảng cách càng gần nhau.

Ví dụ: Dùng công thức Cosine để tính độ tương tự giữa 2 văn bản:

Văn bản A: I am a student.

Văn Bản B: I am a teacher.

Văn bản C: I am a doctor.

Biểu diễn văn bản theo vector:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | I | Am | A | Student | Teacher | doctor |
| Văn bản A | 1 | 1 | 1 | 1 | 0 | 0 |
| Văn bản B | 1 | 1 | 1 | 0 | 1 | 0 |
| Văn bản C | 1 | 1 | 1 | 0 | 0 | 1 |

Vector A = (1,1,1,1,0,0)

Vector B = (1,1,0,1,1,0)

Vector C = (1,1,0,0,1,1)

Điều đó cho thấy văn bản A tương tự văn bản B hơn so với C.

Hướng dẫn cài đặt:

Thông thường các thuật toán sẽ gồm 2 giai đoạn huấn luyện và phân lớp, riêng đối với thuật toán KNN do thuật toán này không cần tạo ra mô hình khi làm trên tập huấn luyện các văn bản đã có nhãn/lớp sẵn, nên không cần giai đoạn huấn luyện (giai đoạn huấn luyện của KNN là gán nhãn cho các văn bản trong tập huấn luyện bằng cách gom nhóm các văn bản có vector đặc trưng giống nhau thành cùng 1 nhóm).

Mô tả vector đặc trưng của văn bản: là vector có số chiều là số đặc trưng trong toàn tập dữ liệu, các đặc trưng này đôi một khác nhau. Nếu văn bản có chứa đặc trưng đó sẽ có giá trị 1, ngược lại là 0.

**Đầu vào:**

* Vector đặc trưng của văn bản cần phân lớp.
* Các vector đặc trưng của văn bản trong tập huấn luyện (Ma trận MxN, với M là số vector đặc trưng trong tập huấn luyện, N là số đặc trưng của vector).
* Tập nhãn/lớp cho từng vector đặc trưng của tập huấn luyện.

**Đầu ra:**

* Nhãn/lớp của văn bản cần phân loại.

**Quá trình phân lớp gồm các bước sau:**

* Xác định giá trị tham số K (số láng giềng gần nhất). Tùy vào mỗi tập huấn luyện (số lượng mẫu trong tập huấn luyện, không gian tập mẫu có phủ hết các trường hợp…) mà việc chọn số K sẽ ảnh hưởng đến kết quả phân lớp.
* Lần lượt duyệt qua các văn bản (được đại diện bằng vector đặc trưng của văn bản) trong tập huấn luyện và tính độ tương tự của văn bản đó với văn bản cần phân lớp.
* Sau khi đã có mảng các giá trị lưu độ tương tự của văn bản cần phân lớp với các văn bản trong tập huấn luyện, ta sắp xếp độ tương tự các văn bản theo thứ tự giảm dần (lưu ý đây là độ tương tự, độ tương tự càng lớn tức là khoảng cách càng gần) và lấy ra k văn bản đầu tiên trong mảng (tức là k văn bản gần với văn bản cần phân lớp nhất).
* Khởi tạo mảng A có độ dài bằng số phân lớp để lưu số văn bản của mỗi lớp. Duyệt qua k văn bản, đếm số văn bản trong từng phân lớp và lưu vào mảng.
* Duyệt qua mảng A, tìm lớp có số văn bản nhiều nhất và chọn là lớp cho văn bản mới.

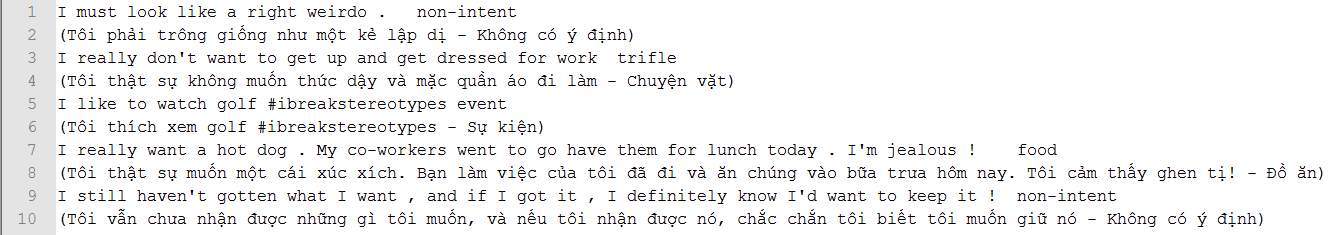
**CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ**

Sau khi tìm hiểu các phương pháp giải quyết bài toán, trong chương này, đồ án trình bày chi tiết thực nghiệm gồm có quá trình thu thập và xử lý dữ liệu, các phương pháp làm thực nghiệm, kết quả và đánh giá.

* 1. **Dữ liệu thực nghiệm**
     1. **Chuẩn bị dữ liệu**

Bộ dữ liệu chúng tôi tiến hành thực nghiệm bao gồm 2130 tweets của người dùng mạng xã hội Twitter bao gồm 7 loại ý định được gán nhãn cụ thể [2]. Mỗi tweet là một dòng trong file dữ liệu “*data.txt”* sẽ có dạng:

<nội dung tweet>\_dấu tab\_<nhãn>



*Hình 3-1: File data.txt (\*)*

Sau đây là bảng số lượng cụ thể của mỗi nhãn loại.

*Bảng 3-1: Số lượng tweet của mỗi nhãn*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nhãn tiếng Anh | Nhãn tiếng Việt | Số lượng | Tỉ lệ (%) |
| 1 | Food | Đồ ăn & Thức uống | 245 | 11.50% |
| 2 | Travel | Du lịch | 187 | 8.78% |
| 3 | Career | Nghề nghiệp và giáo dục | 159 | 7.46% |
| 4 | Goods | Hàng hóa và dịch vụ | 251 | 11.78% |
| 5 | Event | Sự kiện và các hoạt động | 321 | 15.07% |
| 6 | Trifle | Chuyện vặt | 436 | 20.47% |
| 7 | Non-intent | Không có ý định | 531 | 24.92% |
|  |  | Tổng | 2130 |  |

* + 1. **Tiền xử lý dữ liệu**

Trước khi tiến hành xây dựng dữ liệu thực nghiệm, chúng tôi sẽ tiến hành lọc và loại bỏ một số dữ liệu không cần thiết.

1. **Loại bỏ các từ không mang nghĩa**

Các từ không có nghĩa ở đây là các con số, các ký tự đặc biệt và không mang nghĩa. Ví dụ: “@@”, “#1234&”, “eat2123”,...

Sau khi loại bỏ các từ không mang nghĩa và các ký tự đặc biêt, chúng tôi sẽ tiến hành bước tiếp theo.

1. **Loại bỏ các stop word**

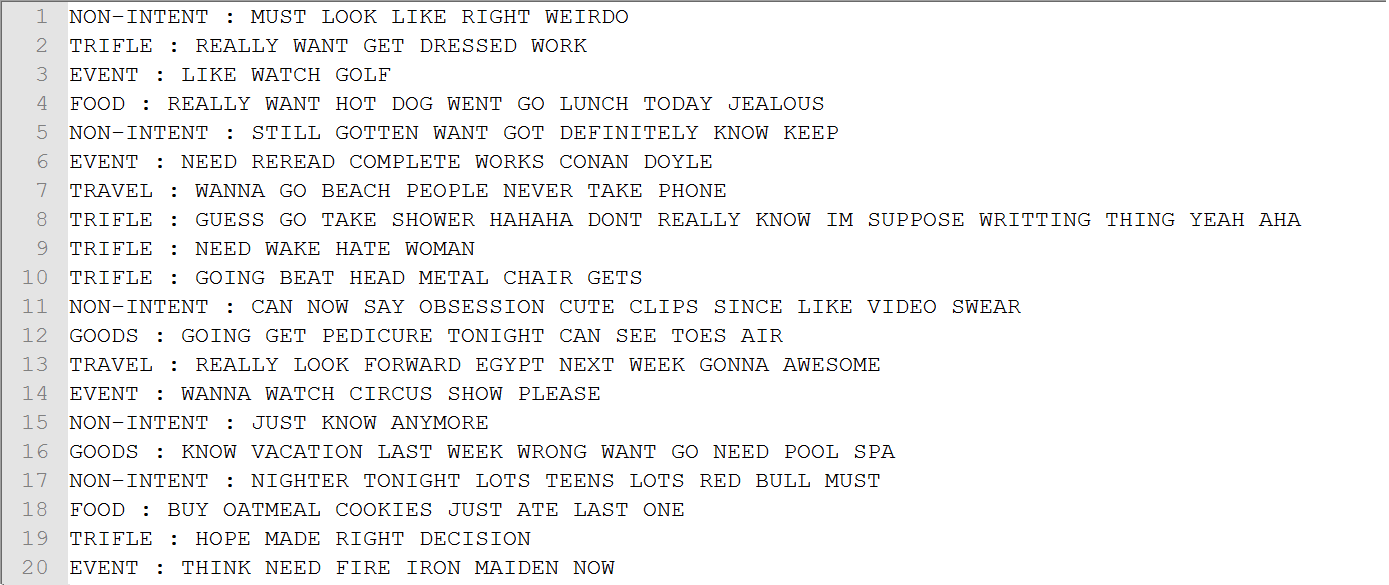
Để tiết kiệm không gian lưu trữ và gia tăng tốc độ xử lý, chúng tôi sẽ không ghi nhận lại những từ quá phổ biến, quá chung chung và những từ này gọi là stop word. Xem ví dụ sau:

“The way to the school is long and hard when walking in the rain.”

Từ "The" xuất hiện 3 lần. Để tiết kiệm không gian lưu trữ cũng như thời gian xử lý, chúng tôi có thể loại bỏ các từ đó đi và thu được kết quả như sau:

“way to school is long and hard when walking in rain”

Đó chỉ là cách mô tả cơ bản của việc loại bỏ các từ trùng lặp (một phần trong danh sách các từ stop words). Để việc lược bỏ được một cách triệt để, chính xác nhất mà vẫn giữ nguyên được tính đúng đắn của câu, ở đây chúng tôi sử dụng danh sách các stop words tiếng Anh [17] chuẩn được lưu tại file “topwords.txt” gồm 175 stop word. Kết quả của việc lọc các từ không cần thiết, chúng tôi sẽ tiến hành lưu trữ và ghi vào một file dữ liệu mới có tên là ”datasetver2.txt” với cấu trúc của mỗi dòng: <nhãn> \_:\_<nội dung câu sau khi lọc>. Cụ thể dữ liệu như sau:



*Hình 3-2: File datasetver2.txt*

Kết quả của việc lọc dữ liệu, chúng tôi thu được file datasetver2.txt và vẫn sử dụng file data.txt để tiếp tục tiến hành xử lý các bước tiếp theo.

* 1. **Thiết lập thực nghiệm**
     1. **Yêu cầu cho thực nghiệm**
* Môi trường cài đặt cần thiết là JRE (Java Runtime Enviroment) phiên bản 1.6 trở lên.
* Phần mềm sử dụng: Weka phiên bản 3.8
* Dữ liệu chuẩn bị: Các file định dạng đuôi .arff được tính toán dựa vào các phương pháp lấy đặc trưng: N-grams, Glove Vector và Log count ratio.
  + 1. **Cách thức thực nghiệm**

Chia ngẫu nhiên mỗi tệp thành 10 phần bằng nhau, sử dụng 9 phần để huấn luyện và 1 phần còn lại để kiểm thử và việc này sẽ được thực hiện 10 lần.

* + 1. **Cách thức đánh giá**

Sau khi đã thực nghiệm xong, việc tiếp theo là cần phải đánh giá xem kết quả thực nghiệm có tốt không. Để đánh giá kết quả phân loại văn bản, ta sử dụng một số chỉ số đánh giá:

* Recall: độ chính xác của dự đoán cho từng nhãn

Recall =

* Precision: độ chính xác của mỗi lần dự đoán

Precision =

Từ thực nghiệm chúng tôi có 3 giá trị: Human, Model, Match

Trong đó:

* Human: Số văn bản thực tế
* Model: Số văn bản mà máy dự đoán
* Match: Số văn bản được máy dự đoán đúng

Chúng tôi có công thức:

Rec = =

Prec = =

* F1

Tiêu chí đánh giá F1 là sự kết hợp của 2 tiêu chí đánh giá Precision và Recall

F1 =

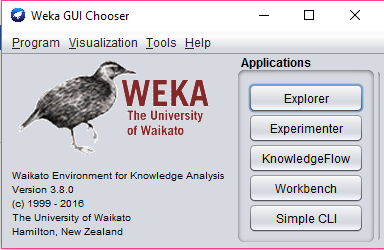
F1 là một giá trị trung bình điều hòa của các tiêu chí Precision và Recall

* F1 có xu hướng lấy giá trị gần với giá trị nào nhỏ hơn giữa 2 giá trị Precision và Recall
* F1 có giá trị lớn nếu cả 2 giá trị Precision và Recall đều lớn.

Để chuẩn bị tiến hành thực nghiệm, sau đây chúng tôi sẽ giới thiệu công cụ thực nghiệm và cách xây dựng dữ liệu thực nghiệm.

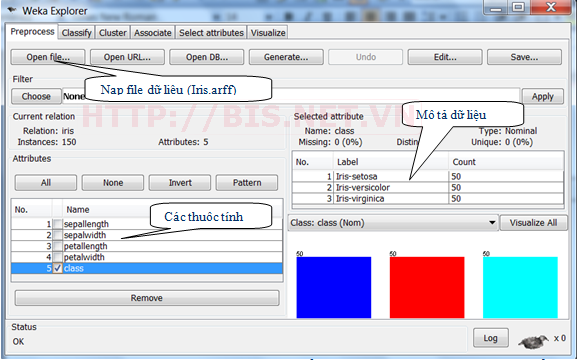
* 1. **Công cụ và dữ liệu thực nghiệm**
     1. **Giới thiệu công cụ Weka**

Weka là một công cụ phần mềm viết bằng Java phục vụ lĩnh vực học máy và khai phá dữ liệu. Weka cung cấp: Một tập các công cụ tiền xử lý dữ liệu, các giải thuật học máy, khai phá dữ liệu, và các phương pháp thí nghiệm đánh giá; giao diện đồ họa dễ dàng thực hiện; và một môi trường cho phép so sánh các giải thuật học máy và khai phá dữ liệu.



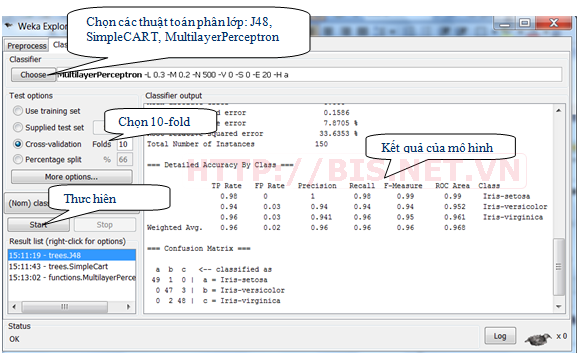
*Hình 3-3: Giao diện màn hình chính của Weka.*

Để sử dụng được các tính năng trong việc sử dụng dữ liệu thực nghiệm thì chúng ta sẽ nhất chọn “Explorer” và nhấn “Open file” để chọn tập tin dữ liệu. Tập tin ở đây sẽ phải đảm bảo được đưa về đúng định dạng mà Weka có thể hiểu được như .csv, .arff, v.v. Weka sẽ đưa ra hiển thị các thống số cơ bản như: số câu, số nhãn và số thuộc tính, v.v.



*Hình 3-4: Giao diện khi chọn một tập tin để thực nghiệm*

Để tiến hành chọn các thuật toán và các lựa chọn thực nghiệm thì chúng ta sẽ nhấn vào tab “Classify”. Sau khi tiến hành lựa chọn và nhấn OK thì kết quả thực nghiệm chi tiết sẽ được hiển thị cụ thể.



*Hình 3-5: Giao diện khi tiến hành thực nghiệm*

Việc lựa chọn thuật toán nào để có một model tốt phụ thuộc rất nhiều yếu tố, trong đó cấu trúc của dataset có ý nghĩa quan trọng đến việc lựa chọn thuật toán. Ví dụ thuật toán cây hồi qui phân loại (CART – Classification And Regression Tree) và J48 cho kết quả tốt trên các dữ liệu kiểu số (Numerical Data), trong khi đó thuật toán ID3 cho kết quả tốt đối với dữ liệu định danh (nominal Data) trong khi đó thuật toán SMO lại cho kết quả tốt đối với dữ liệu kiểu text được gán nhãn (label Data).

* + 1. **Giới thiệu chuẩn dữ liệu đầu vào cho thực nghiệm (ARFF)**

Để có thể tiến hành thực nghiệm trên Weka thì điều bắt buộc là phải đưa dữ liệu về một trong các định dạng được Weka quy định sẵn như .csv, .arff. Trong đồ án này, chúng tôi sẽ sử dụng tập tin dưới mô hình ARFF (Atrribute-Regation File Format), tất cả các dữ liệu sẽ được chuyển sang chuẩn dữ liệu ARFF trước khi tiến hành thực nghiệm.

Mô hình ARFF có hai phần riêng biệt Phần đầu tiên là thông tin **header**, theo sau đó là các thông tin **dữ** **liệu**. **Header** của file ARFF chứa tên các mối quan hệ, một danh sách các thuộc tính và loại của chúng.

* Các kiểu dữ liệu:

Numeric: dữ liệu dạng số Ví dụ: @ATTRIBUTE name numeric

Nominal: dữ liệu rời rạc Ví dụ: @ATTRIBUTE class {sport, count}

String: dữ liệu chuỗi Ví dụ: @ATTRIBUTE name string Date

Date: dữ liệu kiểu ngày Ví dụ: @ATTRIBUTE discovered date

Ví dụ như sau:

@RELATION iris

@ATTRIBUTE sepallength NUMERIC

@ATTRIBUTE sepalwidth NUMERIC

@ATTRIBUTE petallength NUMERIC

@ATTRIBUTE petalwidth NUMERIC

@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

Ở đây:

* 1. Tên quan hệ là: iris
  2. Có 5 thuộc tính và loại tương ứng:
     1. sepallength có loại là kiểu số
     2. sepalwidth có loại là kiểu số.
     3. petallength có loại là kiểu số.
     4. petalwidth có loại là kiểu số.
     5. Thuộc tính class bao gồm 3 loại: Iris-setosa,Iris-versicolor, Iris-virginica.

Các **dữ liệu** của tập tin ARFF trông giống như sau:

@DATA

5.1,3.5,1.4,0.2,Iris-setosa

4.9,3.0,1.4,0.2,Iris-setosa

4.7,3.2,1.3,0.2,Iris-setosa

4.6,3.1,1.5,0.2,Iris-setosa

5.0,3.6,1.4,0.2,Iris-setosa

5.4,3.9,1.7,0.4,Iris-setosa

4.6,3.4,1.4,0.3,Iris-setosa

5.0,3.4,1.5,0.2,Iris-setosa

4.4,2.9,1.4,0.2,Iris-setosa

4.9,3.1,1.5,0.1,Iris-setosa

Nhưng trong thực tế, các thuộc tính có thể xuất hiện rất nhiều giá trị 0. Thế nên, những giá trị bằng 0 thì chúng tôi lược bớt đi và sẽ không thể hiện trong file mô hình ARFF. Thay vào đó, những giá trị còn lại sẽ được biểu diễn kèm theo thứ tự thuộc tính của chúng. Ví dụ:

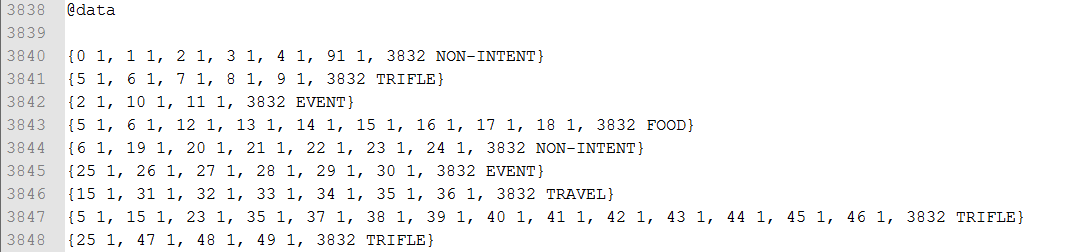
Với mô hình ARFF ban đầu, giả sử như đang sử dụng thuộc tính mô tả giống như ví dụ trong định nghĩa trên, chúng tôi có biểu diễn sau:

5.1,0.0,1.4,0.2,Iris-setosa

Ta thấy có một giá trị bằng 0, có thể lược bớt giá trị đó trong cách biểu diễn thành như sau:

{0 5.1, 2 1.4, 3 0.2, 4 Iris-setosa}

Có nghĩa là tất cả các thuộc tính còn lại sẽ được biểu diễn kèm thuộc tính của nó. Người ta gọi đó là mô hình ARFF mềm dẻo, và chắc chắn, định dạng tệp vẫn là .arff. Mô hình này vẫn giữ nguyên yêu cầu đúng chuẩn mà Weka đọc hiểu được và giảm được đáng kể dung lượng lưu trữ cũng như thời gian thực nghiệm.



*Hình 3-6: Ví dụ minh họa một phần biểu diễn ARFF mềm dẻo*

* 1. **Tiến hành thực nghiệm**
     1. **Chuẩn bị dữ liệu cho các phương pháp thực nghiệm**

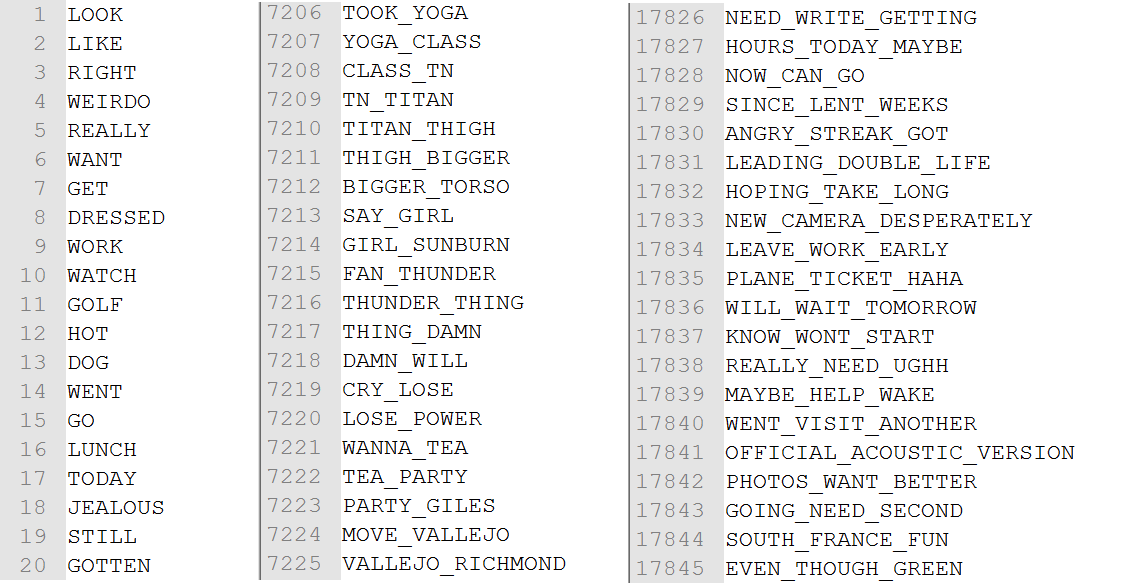
1. **SVM**

Trong thực nghiệm với thuật toán SVM, chúng tôi sử dụng công cụ Weka để phân loại ý định tweet với các đặc trưng n-gram.

* 1. **Bước 1: Xây dựng từ điển**

Sử dụng dữ liệu trong tập tin “datasetver2.txt” xây dựng bộ từ điển chưa các đặc trưng âm tiết được biểu diễn theo 1 gram, 1+2 grams và 1+2+3 grams. Mỗi dòng trong từ điển có định dạng: *<từ trong từ điển>.* Về vị trí của từ trong từ điển, chúng tôi sẽ không lưu vào trong file lưu trữ, do khi gọi đến và sử dụng trong Arraylist, thì vị trí của nó chính là số thứ tự trong mảng.

Từ 1-gram (UniGram) thì chúng tôi biểu diễn như bình thường, đối với các từ 2-gram (BiGram) và 3-gram (TriGram) thì giữa các gram, chúng tôi nối liền bằng dấu “\_”. Cụ thể như sau:

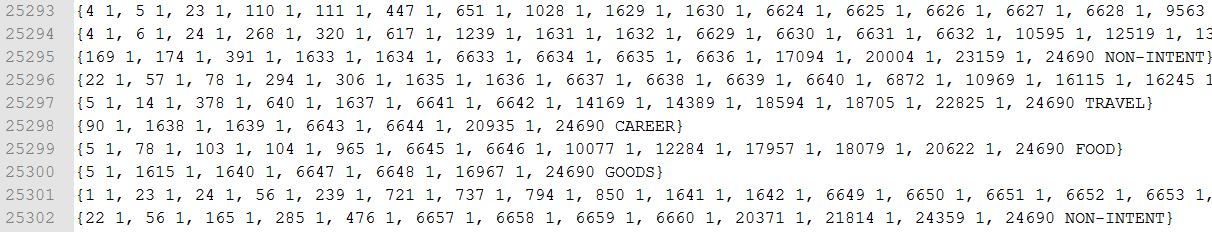


*Hình 3-7: File từ điển dictionary.txt*

* 1. **Bước 2: Tạo bộ dữ liệu thực nghiệm**

Dựa vào bộ từ điển vừa xây dựng, chuyển đổi thành các đặc trưng theo đúng định dạng ARFF mềm dẻo. 

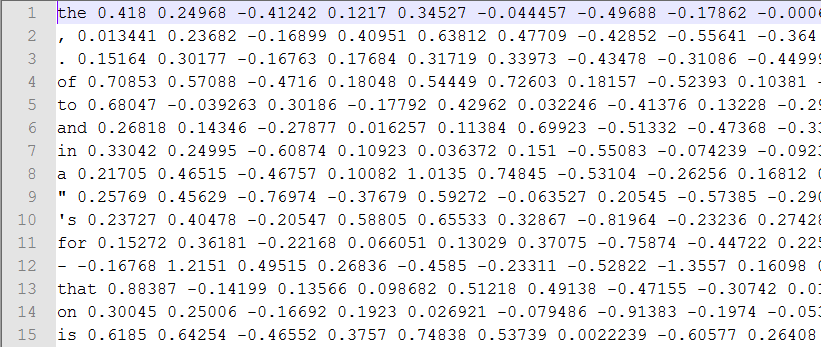
Trong đó:

*  là thứ tự của từ thứ  trong câu xuất hiện trong từ điển.
*  là số lần xuất hiện của từ thứ  trong câu.
*  là thuộc tính chứa class, ở đây sẽ là số ngay sau  với  là số từ trong từ điển.
* **** là nhãn ý định của câu đó.

*Hình 3-8: Một phần tập tin định dạng dữ liệu cho Weka*

1. **Glove Vector**

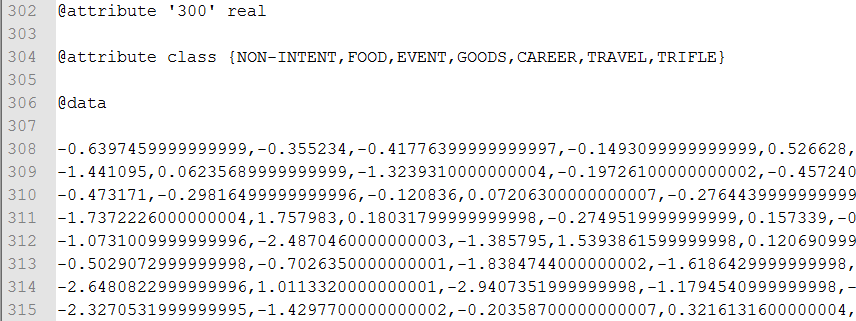
**Bước 1: Xây dựng từ điển**

****Mỗi ký tự sẽ được đi kèm với một ma trận vector cố định (ở đây chúng tôi sử dụng file từ điển có độ dài của mỗi vector là 300 để đạt kết quả tốt nhất)[16]. Cụ thể:

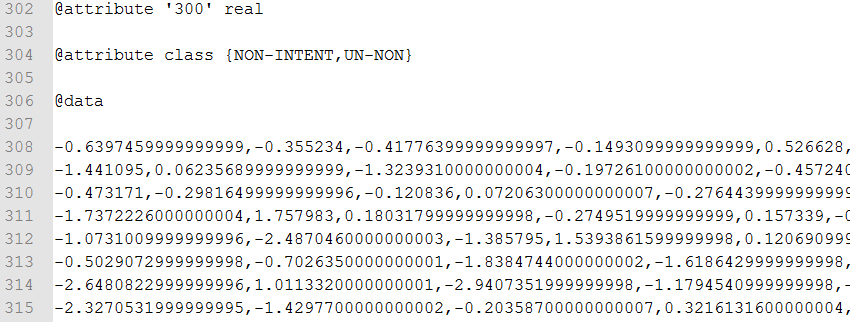
*Hình 3-9: Một phần tập tin glove6B300d.txt*

**Bước 2: Tạo bộ dữ liệu thực nghiệm**

Dựa vào bộ từ điển vừa xây dựng, chuyển đổi thành các đặc trưng theo đúng định dạng ARFF**.** Với mỗi nội dung tweet chúng tôi lấy được từ file dữ liệu “data.txt”, chúng tôi sẽ tiến hành tách từ, với mỗi từ, chúng tôi sẽ lấy vector Golve tương ứng của chúng trong từ điển Glove đã cho, sau đó tiến hành cộng dồn vào sẽ thu được một vector có độ dài là 300 là vector Glove của câu tweet đó đi kèm nhãn. Với một nội dung như vậy, việc cuối cùng chúng tôi cần làm là tiến hành biểu diễn chúng dưới mô hình ARFF rồi lưu dữ liệu vào file “resultGlove300D7” (300 chiều với 7 nhãn ý định) và file “resultGlove300D2” (300 chiều với 2 nhãn ý định) để chuẩn bị cho việc thực nghiệm trên weka. Cụ thể:

****

*Hình 3-10: Một phần file “resultGlove300D7.txt”*

**

*Hình 3-11: Một phần file “resultGlove300D7.txt”*

1. **Log count ratio**

Phương pháp lấy đặc trưng này sử dụng chung từ điển với phương pháp N-grams, chúng tôi vẫn sử dụng tập dữ liệu sau khi lọc là “datasetver2.txt” để tiến hành xử lý.

Ở đây, chúng tôi cần nhắc lại lý thuyết một chút. Điểm khác biệt duy nhất ở phương pháp này, là chúng tôi cần tìm được vector  được tính như sau:

****

Để dễ hình dung, chúng tôi tưởng tượng như sau:

* Dữ liệu được chia thành 7 nhãn, nhưng ở đây chúng tôi thấy  và  nên chúng tôi sẽ gán tạm thành 2 nhãn “non-intent” và “unnon-intent”.
* Cách chia lại rất đơn giản, nhãn “non-intent” vẫn giữ nguyên, nhãn “unnon-intent” sẽ bao gồm 6 nhãn còn lại.

Chúng tôi sẽ xây dựng từ điển riêng của 2 nhãn:

* dicNon là từ điển từ của các câu có nhãn là “non-intent”.
* dicUnnon là từ điển từ của các câu có nhãn là “unnon-intent”.

Ở đây xây dựng 2 ma trận  và có độ dài bằng độ dài của từ điển và gán như sau:

*  nếu từ thứ  trong từ điển xuất hiện trong dicNon.
*  nếu từ thứ  trong từ điển xuất hiện trong dicNon.
*  nếu từ thứ  trong từ điển xuất hiện trong dicUnnon.
*  nếu từ thứ  trong từ điển xuất hiện trong dicUnnon.

Sau đó tiến hành tính lại 2 vector và theo công thức:

 và  với  ,  là độ dài của từ điển.

Khi đó, vector  được tính theo công thức sau:



Giả sử hệ vector thu được sau khi xây dựng dữ liệu của phương pháp SVM là  . Thì ta sẽ tính lại u theo công thức sau:



Kết quả thu được lưu vào file dữ liệu để chuẩn bị tiến hành thực nghiệm bằng weka.

* 1. **Kết quả thực nghiệm**

Chúng tôi sẽ tiến hành thực nghiệm và đo kết quả trên cả việc phân loại 7 nhãn và 2 nhãn để có cái nhìn tổng thể về mỗi phương pháp.

* + 1. **Theo các thuật toán khác nhau**

Trong đồ án này, chúng tôi tiến hành tìm hiểu 3 thuật toán: Máy véc tơ hỗ trợ (Support Vector Machine), K láng giềng gần nhất (K-nearest neighbors), Naive Bayes.

Để kiểm tra xem thuật toán nào sẽ cho kết quả dự đoán tốt nhất, chúng tôi sẽ sử dụng 1 trong 3 từ điển đã được xây dựng ở phần trên. Ở đây, chúng tôi xin chọn từ điển của phương pháp lấy đặc trưng N-Grams và được kết quả như sau:

*Bảng 3-2: Kết quả thực nghiệm sử dụng thuật toán SVM*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | 83.14 | 82.64 | 83.14 | 82.89 |
|  | BiGram | 83.87 | 83.21 | 83.87 | 83.54 |
|  | TriGram | **84.18** | 83.45 | 84.18 | 83.81 |
| 7 nhãn | UniGram | **64.21** | 66.46 | 64.21 | 65.31 |
|  | BiGRam | 63.35 | 67.25 | 63.35 | 65.24 |
|  | TriGRam | 62.47 | 67.34 | 62.47 | 64.81 |

Với thuật toán SVM, phương pháp trích chọn đặc trưng với TriGram cho chúng ta kết quả tốt nhất với độ chính xác là 84.18% đối với 7 nhãn, nhưng đối với 2 nhãn thì trích chọn đặc trưng UniGram mới đưa ra kết quả cao nhất với 64.21% chính xác.

*Bảng 3-3: Kết quả thực nghiệm sử dụng thuật toán KNN (k=1)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | **78.04** | 77.79 | 78.04 | 77.89 |
|  | BiGram | 77.09 | 76.81 | 77.09 | 76.93 |
|  | TriGram | 77.70 | 77.48 | 77.70 | 77.57 |
| 7 nhãn | UniGram | 49.31 | 49.82 | 49.31 | 49.39 |
|  | BiGRam | 49.60 | 50.01 | 49.60 | 49.68 |
|  | TriGRam | **49.81** | 50.32 | 49.81 | 50.02 |

Với thuật toán KNN (k=1) thì kết quả là thấp hơn rõ ràng so với thuật toán SVM. Ngược lại với thuật toán SVM, đặc trưng UniGram cho khả năng dự đoán tốt nhất với 7 nhãn (78.04%) còn đặc trưng TriGram lại cho độ chính xác cao nhất với phân loại 7 nhãn (49.21%).

*Bảng 3-4: Kết quả thực nghiệm sử dụng thuật toán Naive Bayes*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | 82.14 | 81.16 | 82.14 | 81.65 |
|  | BiGram | 82.31 | 81.51 | 82.31 | 81.72 |
|  | TriGram | **82.51** | 82.69 | 82.51 | 82.60 |
| 7 nhãn | UniGram | **39.72** | 38.77 | 39.72 | 39.02 |
|  | BiGRam | 39.78 | 39.09 | 39.78 | 39.14 |
|  | TriGRam | 39.51 | 38.80 | 39.51 | 39.21 |

Thuật toán này dự đoán khá tốt với 2 nhãn ý định khi độ chính xác tốt nhất là 82.51% cho đặc trưng TriGram. Nhưng với 7 nhãn ý định thì độ chính xác rất thấp (tốt nhất là 39.72% với đặc trưng UniGram). Như vậy cho thấy, thuật toán này xử lý tương đối tốt với các bài toán phân biệt đối lập (có 2 nhãn).

Để dễ hình dung hơn thì chúng ta xem biểu đồ sau:

Có thể dễ dàng nhận thấy, thuật toán Máy véc tơ hỗ trợ (Support Vector Machine - SVM) đạt độ chính xác cao nhất. Đây cũng là thuật toán hiện nay thường được dùng cho các bài toán phân nhãn dữ liệu.

Như vậy, chúng tôi sẽ chọn thuật toán SVM để tiến hành phân tích cũng như thực nghiệm đánh giá các phương pháp trích đặc trưng.

* + 1. **Theo các phương pháp lấy đặc trưng khác nhau**

Ở phần này, chúng tôi sẽ đều sử dụng thuật toán SVM cho 3 phương pháp lấy đặc trưng khác nhau.

*Bảng 3-5: Kết quả thực nghiệm sử dụng n-grams (thuật toán SVM)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | 83.14 | 82.64 | 83.14 | 82.89 |
|  | BiGram | 83.87 | 83.21 | 83.87 | 83.54 |
|  | TriGram | **84.18** | 83.45 | 84.18 | 83.81 |
| 7 nhãn | UniGram | **64.21** | 66.46 | 64.21 | 65.31 |
|  | BiGRam | 63.35 | 67.25 | 63.35 | 65.24 |
|  | TriGRam | 62.47 | 67.34 | 62.47 | 64.81 |

*Bảng 3-6: Kết quả thực nghiệm Glove Vector (thuật toán SVM)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | **83.27** | 82.52 | 83.27 | 82.89 |
| 7 nhãn | UniGram | **67.87** | 68.63 | 67.87 | 68.25 |

*Bảng 3-7: Kết quả thực nghiệm Log count ratio sử dụng n-grams (thuật toán SVM)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | **81.35** | 80.12 | 81.35 | 80.73 |
|  | BiGram | 80.28 | 80.43 | 80.28 | 80.35 |
|  | TriGram | 79.73 | 79.30 | 79.73 | 79.43 |
| 7 nhãn | UniGram | **51.69** | 53.12 | 51.69 | 52.03 |
|  | BiGRam | 48.45 | 51.97 | 48.45 | 48.72 |
|  | TriGRam | 48.91 | 52.42 | 48.91 | 49.17 |

Để dễ hiểu, chúng tôi có thể nhìn biểu đồ so sánh sau:

Kết quả thực nghiệm với các phương pháp lấy đặc trưng khác nhau cũng cho các kết quả dự đoán khác nhau:

* Việc dự đoán chính xác nếu có 2 nhãn là dễ dàng hơn rất nhiều so với dự đoán có 7 nhãn.
* Với 2 nhãn, khả năng dự đoán cao nhất thuộc về phương pháp lấy đặc trưng n-grams với Acc = 84.18% với lấy đặc trưng TriGram.
* Với 7 nhãn, khả năng dự đoán cao nhất thuộc về phương pháp lấy đặc trưng Glove Vector với Acc = 67.87%.
* Tựu chung lại, 2 phương pháp N-grams và Glove Vector có khả năng dự đoán cao hơn lấy đặc trưng theo phương pháp Log count ratio.
* Không hẳn đặc trưng TriGram đã cho kết quả tốt hơn so với UniGram. Điều này còn phụ thuộc vào dữ liệu đầu vào và thuật toán được lựa chọn.

Kết quả thực nghiệm cho thấy, để giải quyết bài toán hiện tại chúng tôi đang đặt ra, thì phương pháp Glove Vector đang đạt kết quả dự đoán tốt nhất (67.87%). Nên chúng tôi sẽ đi sâu hơn một chút về kết quả của phương pháp này và tìm cách bổ sung các đặc trưng để có thể kết quả dự đoán sẽ tốt hơn.

Nhìn vào biểu đồ ta có thể thấy, với phương pháp lấy đặc trưng theo Glove Vector thì: 2 nhãn dự đoán đúng nhiều nhất là Food và Travel. Các nhãn còn lại có độ chính xác ở mức tạm chấp nhận.

Chúng tôi có thể kiểm tra thêm một phương pháp nữa, là đặc trưng UniGram có độ chính xác là 64.21% xem nhãn nào có độ chính xác cao nhất.

Như vậy, phương pháp N-Grams thì nhãn Non-intent và Food được dự đoán khá tốt. Như vậy, chúng tôi có thể kết hợp 2 cách lấy đặc trưng này lại với nhau để tăng độ chính xác của bài toán. Cách thức kết hợp cũng rất đơn giản, chỉ cần nối các thuộc tính của hai phương pháp lại với nhau (độ dài vector đặc trưng sẽ tăng lên).

* + 1. **Đề xuất phương pháp lấy đặc trưng mới**

Bằng những kiến thức học được từ môn xác suất thống kê cũng như trí tuệ nhân tạo, chúng tôi xin phép được thử cũng như đề xuất một phương pháp lấy đặc trưng mà tôi nghĩ là phần nào áp dụng được những kiến thức đó một cách cơ bản nhất mà không phải sử dụng lại lối mòn các phương pháp lấy đặc trưng đã được giới thiệu.

**Xây dựng từ điển**

Từ điển ở phương pháp này, chúng tôi sẽ sử dụng từ file “datasetver2.txt”, nghĩa là đã loại bỏ các từ không mang nghĩa cũng như là stop words.

**Cơ sở lý thuyết**

Lý thuyết của cách lấy đặc trưng này dựa trên những điều rút ra về xác suất thống kê như sau: “**ngựa quen đường cũ**”. Đây là một câu thành ngữ quen thuộc của ông cha ta nhưng hôm nay xin áp dụng vào lý thuyết phân loại, có nghĩa là: xác suất của một câu mang nhãn gì (1 hoặc -1) phụ thuộc vào xác suất của từng từ trong câu đó mang nhãn gì.

Đây chính là cơ sở lý thuyết của thuật toán Naive Bayes trong việc phân nhãn từ loại. Điều quan trọng nhất ở đây là lựa chọn được các thuộc tính quan trọng nhất để xem sự phụ thuộc của chúng vào nhãn từ loại.

Chúng tôi xem xét và đề xuất một số thuộc tính sau:

* Số lượng từ trong câu
* Số lượng từ không mang nghĩa trong câu
* Số lượng từ mang nghĩa trong câu
* Tổng xác suất mỗi từ mang nghĩa trong câu xuất hiện trong từ điển của nhãn -1
* Tổng xác suất mỗi từ mang nghĩa trong câu xuất hiện trong từ điển của nhãn 1
* Tích xác suất mỗi từ mang nghĩa trong câu xuất hiện trong từ điển của nhãn -1
* Tích xác suất mỗi từ mang nghĩa trong câu xuất hiện trong từ điển của nhãn 1

Sau khi tiến hành thực nghiệm và kiểm tra, lấy kết quả tốt nhất và chốt được các thuộc tính sau:

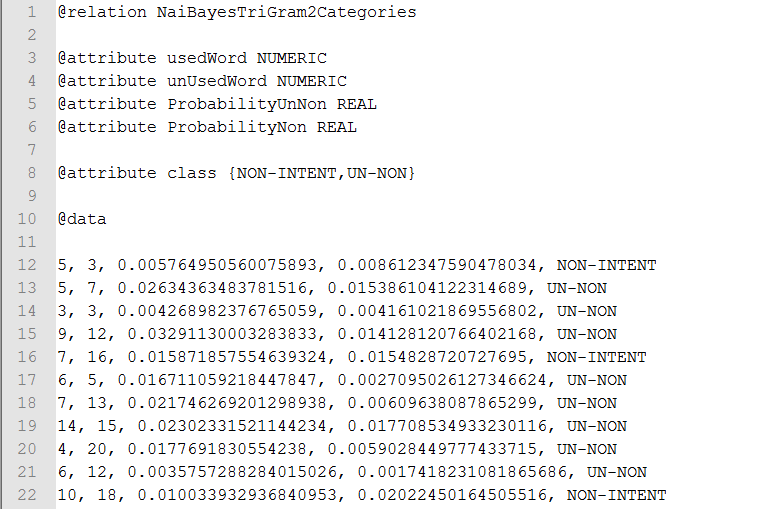
* Số lượng từ trong câu
* Số lượng từ không mang nghĩa trong câu
* Số lượng từ mang nghĩa trong câu
* Tổng xác suất mỗi từ mang nghĩa trong câu xuất hiện trong từ điển của nhãn -1
* Tổng xác suất mỗi từ mang nghĩa trong câu xuất hiện trong từ điển của nhãn 1

**Chuẩn bị dữ liệu**

Với những lý thuyết phía trên, từ dữ liệu thô ban đầu, chúng ta bắt đầu xây dựng ra file để huấn luyện theo mô hình ARFF để tiến hành thực nghiệm trên Weka.

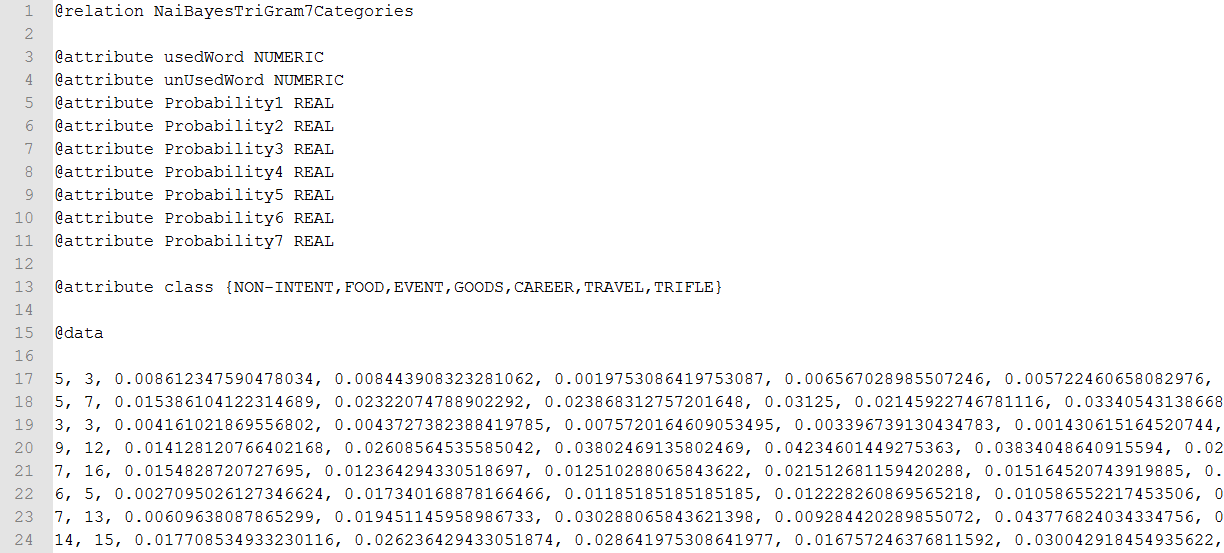
Với 2 nhãn, chúng ta xây dựng các thuộc tính:

* Số lượng từ không mang nghĩa trong câu
* Số lượng từ mang nghĩa trong câu
* Tổng xác suất mỗi từ (mang nghĩa trong câu xuất hiện trong từ điển của nhãn non-intent)
* Tổng xác suất mỗi từ (mang nghĩa trong câu xuất hiện trong từ điển của nhãn unnon-intent)

Cụ thể:

*Hình 3-12: File dữ liệu NaiveBayes2Categories.txt*

Với 7 nhãn, chúng ta xây dựng các thuộc tính

* Số lượng từ không mang nghĩa trong câu
* Số lượng từ mang nghĩa trong câu
* Tổng xác suất mỗi từ (mang nghĩa trong câu xuất hiện trong từ điển của từng nhãn).
* Cụ thể:

*Hình 3-13: File dữ liệu NaiveBayes7Categories.txt*

**Kết quả thực nghiệm**

Sử dụng phần mềm Weka, thuật toán thực hiện là SVM, chúng tôi thu được kết quả sau:

*Bảng 3-8: Kết quả thực nghiệm đặc trưng New Naive Bayes (thuật toán SVM)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | 82.14 | 81.16 | 82.14 | 81.65 |
|  | BiGram | 82.31 | 81.51 | 82.31 | 81.72 |
|  | TriGram | **82.51** | 82.69 | 82.51 | 82.60 |
| 7 nhãn | UniGram | 39.72 | 38.77 | 39.72 | 39.02 |
|  | BiGRam | **39.78** | 39.09 | 39.78 | 39.14 |
|  | TriGRam | 39.51 | 38.80 | 39.51 | 39.21 |

* **Đánh giá kết quả**

Để so sánh phương pháp này với 3 phương pháp quen thuộc đã có, chúng tôi xây dựng biểu đồ so sánh sau:

Như vậy có thể thấy với phân loại 2 nhãn, phương pháp này là tốt hơn phương pháp Log-count ratio và kém một chút so với 2 phương pháp còn lại.

Còn với phân loại 7 nhãn thì sao?

Nhưng đối với 7 nhãn, phương pháp này kém hiệu quả hơn hẳn so với các phương pháp còn lại.

**Đánh giá ưu, nhược điểm**

Ưu điểm lớn nhất của cách lấy đặc trưng này là số thuộc tính ít, dễ dàng xây dựng và không cần có công thức phức tạp. Rõ ràng, trong các phương pháp trước, số thuộc tính có thể lên đến hàng trăm nghìn do độ dài của từ điển là quá lớn, vì thế phương pháp này phần nào đã cải thiện được điều đó.Tuy có số lượng thuộc tính rất ít nhưng không phải vì thế mà độ chính xác của nó là kém. Với 2 nhãn thì khả năng phân loại của chúng khá tốt. Chúng còn phụ thuộc vào cách chúng ta chọn các thuộc tính, ở đây chúng tôi chỉ mới lựa chọn 2 thuộc tính rất cơ bản là số lượng từ và xác suất của từ. Như vậy nếu sau này lựa chọn thêm được những thuộc tính tốt hơn thì có thể cải thiện.

Nhưng bên cạnh đó thì phương pháp trích chọn đặc trưng này cũng sẽ có những nhược điểm. Thứ nhất là khả năng dự đoán với số nhãn lớn hơn là khá thấp, ví dụ như từ kết quả trên có thể thấy, với 7 nhãn thì khả năng đoán đúng của nó chưa đến 50%. Thứ 2, là tuy số lượng thuộc tính là rất ít nhưng cách xây dựng giá trị mỗi thuộc tính đó là khá phức tạp. Nhìn chung, cách trích chọn đặc trưng này cũng khá tốt và cần cải thiện hơn trong thời gian tới.

* + 1. **Tiến hành kết hợp các phương pháp lấy đặc trưng**

Đầu tiên chúng tôi tiến hành kết hợp 2 phương pháp hiện đang đạt độ chính xác cao nhất là N-grams và Glove Vector. Phương pháp Log-count ratio chung tôi sẽ không tiến hành kết hợp vì 2 điều:

* Thứ nhất: quá nhiều thuộc tính, tương đương với độ dài từ điển nên khi kết hợp vào sẽ mất thời gian xử lý và tính toán.
* Thứ hai: độ chính xác không cao, thua các phương pháp còn lại khá nhiều.

Sau khi thực nghiệm được kết quả sau:

*Bảng 3-9: Kết quả thực nghiệm kết hợp Glove Vector và UniGram (thuật toán* *SVM).*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | UniGram | **83.14** | 82.56 | 83.14 | 82.82 |
| 7 nhãn | UniGram | **64.58** | 66.53 | 64.58 | 65.08 |

*Bảng 3-10: Kết quả thực nghiệm kết hợp Glove Vector và TriGram (thuật toán SVM).*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | TriGram | **85.20** | 84.74 | 85.20 | 84.88 |
| 7 nhãn | TriGram | **69.89** | 71.49 | 69.89 | 70.16 |

Tiến hành thử nghiệm, chúng tôi có thể nhận thấy sự thay đổi độ chính xác của bài toán:

* Glove Vector + TriGram = Glove Vector + **1.02%** (2 nhãn)
* Glove Vector + TriGram = Glove Vector + **2.02%** (7 nhãn)

Kết quả kết hợp giữa Glove Vector và TriGram đang cho kết quả tốt nhất. Vì thế chúng tôi sẽ chọn nó để kết hợp thêm phương pháp mới. Sau khi tiến hành thực nghiệm thu được kết quả sau:

*Bảng 3-11 : Kết quả thực nghiệm TriGram + Glove Vector 300D + NaiveBayes Trigram (thuật toán SVM)*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Số nhãn | Số gram | Acc(%) | Precision(%) | Recall(%) | F(%) |
| 2 nhãn | TriGram | **85.44** | 85.01 | 85.44 | 85.09 |
| 7 nhãn | TriGram | **70.50** | 71.89 | 70.05 | 70.82 |

Như vậy, với 2 nhãn thì độ chính xác là 85.44% (tăng thêm 0.24%) và với 7 nhãn thì là 70.50% (tăng thêm 0.61%). Dù không tăng thêm nhiều nhưng phương pháp cũng đã giúp cải thiện độ chính xác của việc phân loại.

**KẾT LUẬN**

Phân loại ý định người dùng thông qua những nội dung mà họ đăng tải trên mạng xã hội hiện nay là một vấn đề rất nóng hổi bởi bì thế giời online đang phát triển một cách không ngừng. Từ việc giải quyết bài toán phân loại ý định người dùng giúp cho chúng ta tiến gần hơn đến sự thông minh của thế giới ảo, giúp quản lý tốt hơn hệ thống thông tin ngập tràn những nội dung. Ngoài ra, dựa vào nghiên cứu này, chúng ta có thể phát triển trí tuệ nhân tạo cho các mạng xã hội, giúp chúng có thể hiểu được nội dung đăng tải của người dùng từ đó đưa ra các lời khuyên, các mặt hàng sản phẩm hay các quảng cáo hữu ích. Từ đó giúp việc sử dụng mạng xã hội tưởng chừng là ảo nhưng lại rất thật và thiết thực với đời sống con người.

Nhìn chung, đồ án đã đạt được một số thành tựu như:

* Trình bày một cách khái quát, tổng quan nhất và nêu lên ý nghĩa, vai trò quan trọng của bài toán phân loại ý định người dùng trên mạng xã hội.
* Khảo sát nghiên cứu 3 loại đặc trưng khác nhau cho bài toán phân loại ý định người dùng.
* Đưa ra một các chọn đặc trưng mới cho bài toán phân loại ý định người dùng nói riêng và bài toàn phân nhãn cho dữ liệu nói chung.
* Nghiên cứu và làm thực nghiệm với 3 thuật toán học máy khác nhau.
* So sánh và phân tích các kết quả thực nghiệm và chúng tôi đã tìm ra được trường hợp cho kết quả tốt nhất.
* Từ kết quả thực nghiệm, chúng tôi xây dựng thành công một ứng dụng demo giúp phân loại ý định người dùng dựa trên các đặc trưng khác nhau.

Đồ án vẫn còn một số hạn chế như:

* Nghiên cứu dựa trên số lượng dữ liệu còn ít và chưa đầy đủ.
* Kết quả thực nghiệm đạt được vẫn chưa thực sự tốt so với kỳ vọng.
* Phần mềm demo vẫn còn hạn chế, tốc độ xử lý vẫn còn chậm, đặc biệt là khi tiến hành huấn luyện dữ liệu mới.

Về hướng phát triển tương lai, chúng tôi sẽ tiến hành thu thập và phát triển trên một tập dữ liệu lớn hơn và dựa trên nhiều đặc trưng hơn để góp phần cải thiện khả năng phân loại. Ngoài ra chúng tôi cũng sẽ nghiên cứu và thử nghiệm với một số thuật toán khác để tìm ra thuật toán phù hợp nhất với bài toán phân loại ý định người dùng mạng xã hội tiếng Anh.

# PHỤ LỤC

1. **Công cụ sử dụng**

*Bảng A-1:* *Các công cụ sử dụng*

|  |  |
| --- | --- |
| Công cụ | Trang chủ |
| Java SE JDK 7u21 | <http://www.java.sun.com> |
| Netbean 8.0.1 | https://netbeans.org/ |
| Notepad++ | http://notepad-plus-plus.org/ |
| Inteliji 2016 | https://www.jetbrains.com/idea/ |
| Weka | http://www.cs.waikato.ac.nz/ml/weka/ |

1. **Chương trình Demo**

Phần này trình bày hệ thống phân loại ý định người dùng của chúng tôi. Hệ thống nhận đầu vào là văn bản tiếng Anh – tương ứng với nội dung tweets nào đó của người dùng và trả về nhãn phân loại cho từng nội dung nhập vào.

**Yêu cầu cấu hình hệ thống:**

* **Hệ điều hành:** Windows 10 / 8.1 / 8 / 7 SP1 / Vista SP2
* **Vi xử lý:** Intel Core i3 530
* **Bộ nhớ RAM:** 4 GB
* **Bộ nhớ máy:** còn trống trên 2GB
* **Môi trường:** JRE (Java Runtime Enviroment) phiên bản 1.6 trở lên.

**Chức năng:**

* Phân loại ý định cho dữ liệu nhập trực tiếp
* Phân loại ý định cho dữ liệu nhập từ tập tin.
* Huấn luyện dữ liệu mới theo yêu cầu của người dùng.
* Phân loại ý định cho đầu vào theo dữ liệu mới được huấn luyện của người dùng.

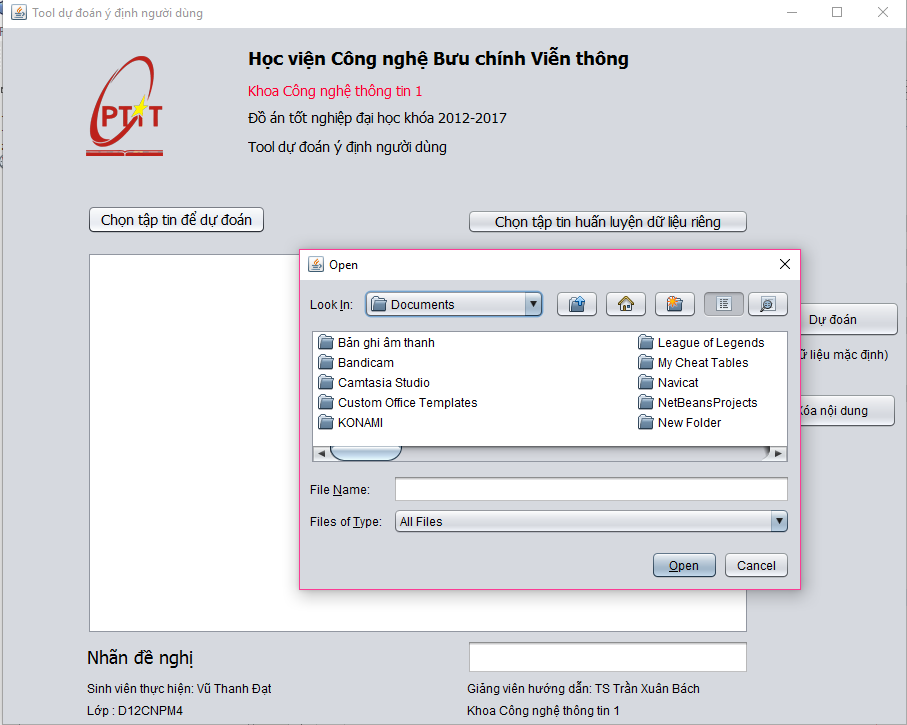


*Hình B-1: Giao diện khi khởi chạy Demo*



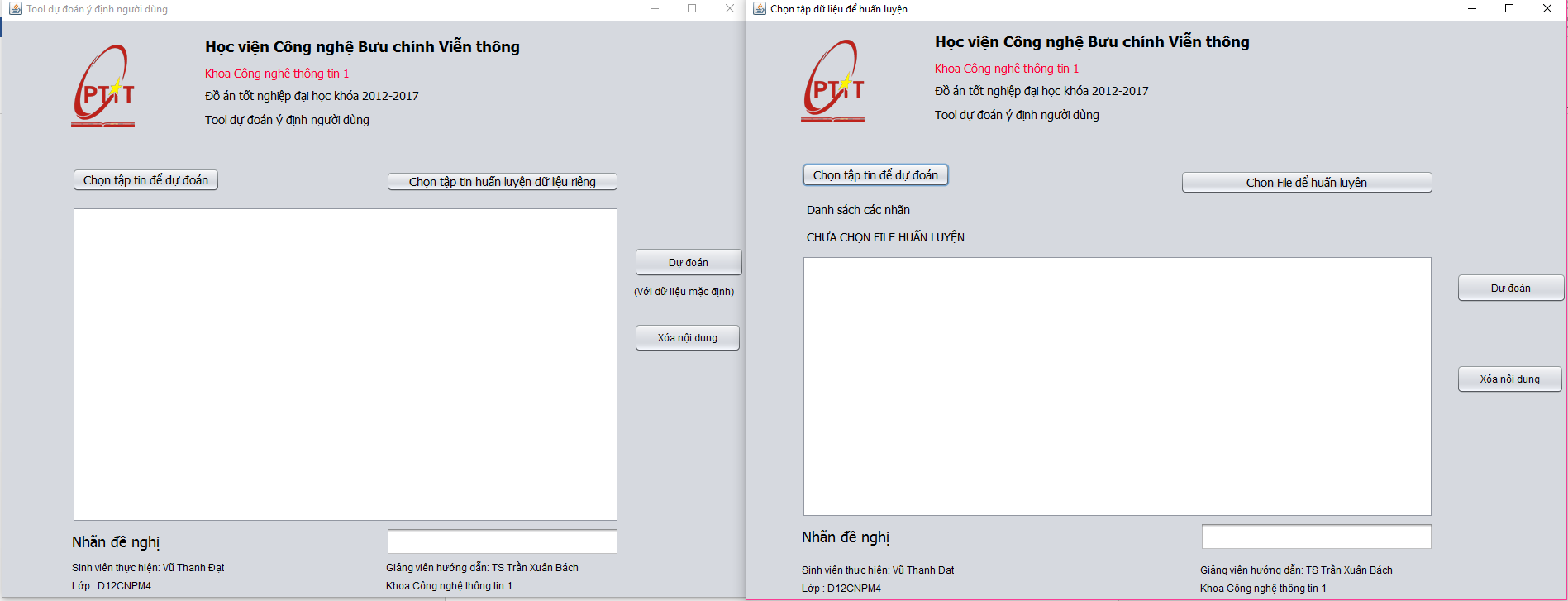
*Hình B-2: Giao diện khi sử dụng lựa chọn nhập bằng tay*

Sau khi nhập văn bản, người dùng nhấn nút “Dự đoán” để hệ thống tính toán và đưa ra kết quả phân loại và được in ra tại ô kết quả bên dưới. Nút “Xóa nội dung” có chức năng xóa nội dung vừa nhập và kết quả để có thể tiến hành các lần tiếp theo.



*Hình B-3: Giao diện khi chọn chức năng nhập bằng tập tin*

Sau khi nhập bằng tập tin, hệ thống sẽ đọc nội dung trong tập tin và dự đoán theo từng dòng và tiến hành xử lý. Sau đó kết quả sẽ được ghi ra tại vùng text lớn trong giao diện.



*Hình B-4: Giao diện khi lựa chọn huấn luyện bằng tập dữ liệu khác*

Khi chọn huấn luyện bằng tập dữ liệu khác, có nghĩa vẫn là bài toán phân loại dữ liệu nhưng nội dung dữ liệu và nhãn dữ liệu hoàn toàn khác. Hệ thống sẽ đọc tập tin và tiến hành phân tích xử lý để học, sau đó sẽ có các chức năng dự đoán giống y hệt các chức năng của bài toàn Phân loại ý định người dùng mạng xã hội.

**TÀI LIỆU THAM KHẢO**

**Tài liệu tiếng Anh:**

1. SidaWang and Christopher D. Manning (2012). *Baselines and Bigrams: Simple, Good Sentiment and Topic Classification.* Stanford University, USA.
2. Jinpeng Wang, Gao Cong,Wayne Xin Zhao, Xiaoming Li (2015). *Mining User Intents in Twitter: A Semi-Supervised - Approach to Inferring Intent Categories for Tweets.*
3. Phuong Le-Hong, Xuan-Hieu Pham and Tien-Dung Nguyen, *Using dependency analysis to improve question classification*, Knowledge and Systems Enginneering, Springer International Publishing, 2015, pages 653-655.
4. Zhiheng Huang, Marcus Thint and Zengchang Qin, *Question Classification using Head Words and their Hypernyms*, Proceedings of the 2008 Conference on Empirical Methods in Natural Languae Processing, pages 927-936, Honolulu, October 2008.
5. Boris Katz. *Annotating the World Wide Web Using Natural Language*. Proceedings of the 5th RIAO Conference on Computer Assisted Information Searching on the Internet (RIAO '97), 1997, pages 136-139.

**Tài liệu tiếng Việt:**

1. Trần Cao Đệ, Phạm Nguyên Khang, *Phân loại văn bản với máy học vector hỗ trợ và cây quyết định*, Tạp chí Khoa học 2012:21a 52-63, Trường Đại học Cần Thơ.
2. Nguyễn Minh Thành, *Phân loại văn bản*, Đồ án môn học Xử lý ngôn ngữ tự nhiên, Đại học quốc gia Thành phố Hồ Chí Minh, 01/2011.
3. Ngô Xuân Bách. *Bài giảng Xử lý ngôn ngữ tự nhiên cho cao học. Học viện Công nghệ Bưu chính Viễn thông*, 2016.
4. Nguyễn Minh Thành, *Phân loại văn bản*, Đồ án môn học Xử lý ngôn ngữ tự nhiên, Đại học quốc gia Thành phố Hồ Chí Minh, 01/2011.
5. Trần Cao Đệ, Phạm Nguyên Khang, *Phân loại văn bản với máy học vector hỗ trợ và cây quyết định*, Tạp chí Khoa học 2012:21a 52-63, Trường Đại học Cần Thơ.

**Danh mục các Website tham khảo:**

1. Google : <https://www.google.com>
2. Wikipedia: <http://www.wikipedia.org>
3. Stack Overflow: http://stackoverflow.com/
4. Natural Language Processing: <http://viet.jnlp.org/>
5. Ông Xuân Hồng – chia sẻ kết thức về Machine Learning: <https://ongxuanhong.wordpress.com/>
6. Glove : Global Vectors for Word Representation: <http://nlp.stanford.edu/projects/glove/>
7. Danh sách stop word tiêu chuẩn

http://www.ranks.nl/stopwords