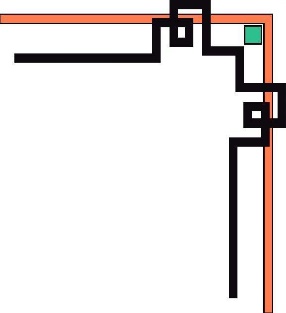
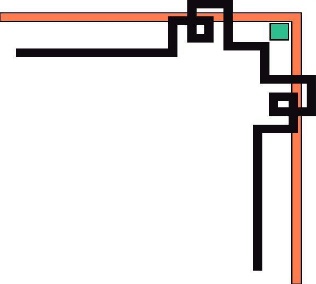
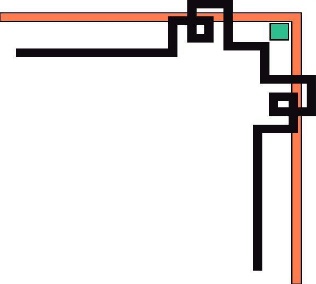
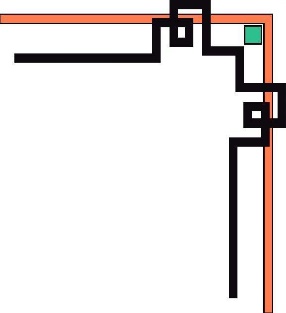
**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

-----

**BÁO CÁO NGÔN NGỮ LẬP TRÌNH JAVA**

**ĐỀ TÀI: LỌC SPAM EMAIL**



**GVHD: hUỲNH TUẤN ANH**

**nGUYỄN TẤN TOÀN**

**lớp: SE330.I21**

**svth: Trần Văn Tịnh – 14520965**

**Trần thanh trà - 14520983**

# NHẬN XÉT CỦA GIÁO VIÊN



# LỜI CẢM ƠN

Đầu tiên, chúng tôi xin gửi lời cảm ơn sâu sắc nhất tới thầy Huỳnh Tuấn Anh, người hướng dẫn khoa học, đã tận tình chỉ bảo, giúp đỡ và tạo điều kiện cho tôi hoàn thành đồ án này.

Chúng tôi xin chân thành cảm ơn các thầy cô trường Đại học Công nghệ Thông Tin, Đại học Quốc gia Hồ Chí Minh đã giảng dạy và truyền đạt kiến thức cho tôi.

Cuối cùng, chúng tôi xin cảm ơn những người thân và các bạn bè đã chia sẻ, giúp đỡ tôi hoàn thành đồ án này.

Mặc dù đã hết sức cố gắng với tất cả sự nỗ lực của bản thân, nhưng chắc đồ án vẫn còn những thiếu sót. Kính mong nhận được những ý kiến đóng góp của quý Thầy, Cô và bạn bè.

Tôi xin chân thành cảm ơn!

HCM, ngày 18 tháng 6 năm 2018

# LỜI MỞ ĐẦU

Từ những năm cuối của thế kỷ XX, những nền móng đầu tiên của nghành công nghệ thông tin (CNTT) đã ra đời. Từ đó đến nay, CNTT đã trở thành một trong nghành có tốc độ phát triển nhanh và ảnh hưởng sâu rộng nhất đến đời sống của con người. CNTT đã và đang thay đổi và cả thiện tất cả các mặt của cuộc sống, từ các nghành nghiên cứu khoa học cho đến hoạt động sản xuất kinh tế và cả chính trị, giải trí. Nhờ có CNTT mà cuộc sống ngày càng dễ dàng hơn, con người từ khắp mọi nơi trên thế giới có thể kết nối với nhau dễ hơn bao giờ hết. Nghành CNTT đã trở thành một trong những nghành kinh tế tạo ra nhiều giá trị nhất của các quốc gia.

Tuy nhiên, cùng với sự phát triển đó thì nhu cầu của con người và xã hội ngày một cao hơn. Con người không chỉ muốn điều khiển máy móc thực hiện mục tiêu của mình mà còn muốn các hệ thống ngày càng thông minh hơn, làm việc hiệu quả hơn. Những phần mềm, robot không chỉ là công cụ làm việc nữa mà trở thành một đồng nghiệp, một người bạn của con người hay thậm chí thay thế con người đưa ra quyết định giải quyết các vấn đề.

Đứng trước nhu cầu đó, nghành Khoa học máy tính nói chung và Trí tuệ nhân tạo nói riêng là một bộ phận đảm nhận trọng trách lớn nhất để tìm ra những giải pháp, những mô hình mới có thể hiện thực hóa những ý tưởng nhằm đáp ứng nhu cầu trên. Những sản phẩm tạo ra phải mang tính “thông minh”, có khả năng học hỏi, suy luận và “gần gũi” với con người hơn nữa.

Không chỉ có nhu cầu, hiện nay vấn đề ứng dụng CNTT vẫn còn phải được quan tâm giải quyết nhiều hơn. Nhà nước ta tuy đã chú trọng vào phát triển nguồn nhân lực CNTT nhiều và chất lượng hơn, nhưng nguồn nhân lực đó đa số vẫn sử dụng cho việc sản xuất kinh tế nhiều hơn cho việc nghiên cứu, giáo dục và thành tựu vẫn còn chưa đồng đều. Trong khi nhu cầu của học sinh hiện nay đang cần có một công cụ giúp hỗ trợ xử lý ngôn ngữ tự nhiên vào máy tính.

Do đó, chúng em đã xây dựng chương trình phân tích cú pháp câu trong bộ ngữ liệu huấn luyện để xử lý ngôn ngữ tự nhiên. Với mục tiêu là giúp học sinh và giáo viên có thể phân tích ngôn ngữ tự nhiên trên máy tính.

# MỤC LỤC

Contents

[NHẬN XÉT CỦA GIÁO VIÊN 2](#_Toc517101300)

[LỜI CẢM ƠN 3](#_Toc517101301)

[LỜI MỞ ĐẦU 4](#_Toc517101302)

[MỤC LỤC 5](#_Toc517101303)

[Chương 1 TỔNG QUAN 6](#_Toc517101304)

[1.1 Hiện trạng vấn đề 6](#_Toc517101305)

[1.1.1 Vấn đề 6](#_Toc517101306)

[1.1.2 Phương pháp giải quyết 8](#_Toc517101307)

[1.2 Hiện trạng cơ sở vật chất và con người 9](#_Toc517101308)

[1.2.1 Cơ sở vật chất 9](#_Toc517101309)

[1.2.2 Con người 9](#_Toc517101310)

[1.3 Yêu cầu sơ bộ về phần mềm 9](#_Toc517101311)

[1.3.1 Tin học 9](#_Toc517101312)

[1.3.2 Con người 10](#_Toc517101313)

[1.4 Yêu cầu đồ án 10](#_Toc517101314)

[Chương 2 THIẾT KẾ 11](#_Toc517101315)

[2.1 Thiết kế giao diện chính 11](#_Toc517101316)

[2.1.1 Giao diện ban đầu 11](#_Toc517101317)

[2.1.2 Giao diện chạy chương trình 12](#_Toc517101318)

[2.2 Các nút lệnh và thao tác 12](#_Toc517101319)

[2.2.1 Phần Load dữ liệu 12](#_Toc517101320)

[2.2.2 Phần Kiểm tra mail 13](#_Toc517101321)

[2.2.3 Phần Rút trích đặc trưng 13](#_Toc517101322)

[2.2.4 Phần Đánh giá 14](#_Toc517101323)

[Chương 3 CÀI ĐẶT PHẦN MỀM 15](#_Toc517101324)

[3.1 Ngôn ngữ lập trình Java 15](#_Toc517101325)

[3.1.1 Sự phổ biến 15](#_Toc517101326)

[3.1.2 Khả năng của ngôn ngữ Java 15](#_Toc517101327)

[3.1.3 Những đặc điểm của ngôn ngữ Java 15](#_Toc517101328)

[3.1.4 Máy ảo Java (JMV - Java Virtual Machine) 16](#_Toc517101329)

[3.1.5 Hai kiểu ứng dụng dưới ngôn ngữ java 16](#_Toc517101330)

[3.1.6 Bộ phát triển ứng dụng Java (JDK- Java Development Kit) 17](#_Toc517101331)

[3.2 Cài đặt 17](#_Toc517101332)

[3.2.1 Phương pháp phân tích e-mail 17](#_Toc517101333)

[3.2.2 Phương pháp lựa chọn đặc trưng 18](#_Toc517101334)

[Chương 4 KẾT QUẢ lọc spam mail bằng phương pháp phân lớp naïve bayes 19](#_Toc517101335)

[4.1 Ví dụ 1: Phân tích Mail thường 19](#_Toc517101336)

[4.2 Ví dụ 2: Phân tích Mail rác 20](#_Toc517101337)

[Chương 5 TỔNG KẾT 21](#_Toc517101338)

[5.1 Kết quả đạt được 21](#_Toc517101339)

[5.2 Hạn chế 21](#_Toc517101340)

[5.3 Hướng phát triển 21](#_Toc517101341)

[TÀI LIỆU THAM KHẢO 22](#_Toc517101342)

# TỔNG QUAN

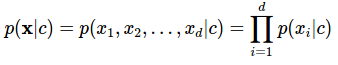
## Hiện trạng vấn đề

### Vấn đề

* Xử lý ngôn ngữ tự nhiên (*natural language processing* - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ-công cụ hoàn hảo nhất của tư duy và giao tiếp.
* Phân tích hình thái - Trong bước này từng từ sẽ được phân tích và các ký tự không phải chữ (như các dấu câu) sẽ được tách ra khỏi các từ. Trong [tiếng Anh](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Anh) và nhiều ngôn ngữ khác, các từ được phân tách với nhau bằng dấu cách. Tuy nhiên trong [tiếng Việt](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Vi%E1%BB%87t), dấu cách được dùng để phân tách các tiếng (âm tiết) chứ không phải từ. Cùng với các ngôn ngữ như [tiếng Trung](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Trung_Qu%E1%BB%91c), [tiếng Hàn](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Tri%E1%BB%81u_Ti%C3%AAn),[tiếng Nhật](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Nh%E1%BA%ADt), [phân tách từ](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_t%C3%A1ch_t%E1%BB%AB&action=edit&redlink=1) trong tiếng Việt là một công việc không hề đơn giản.
* Phân tích cú pháp - Dãy các từ sẽ được biến đổi thành các cấu trúc thể hiện sự liên kết giữa các từ này. Sẽ có những dãy từ bị loại do vi phạm các luật văn phạm.
* Phân tích ngữ nghĩa - Thêm ngữ nghĩa vào các cấu trúc được tạo ra bởi bộ phân tích cú pháp.
* Tích hợp văn bản - Ngữ nghĩa của một câu riêng biệt có thể phụ thuộc vào những câu đứng trước, đồng thời nó cũng có thể ảnh hưởng đến các câu phía sau.
* Phân tích thực nghĩa - Cấu trúc thể hiện điều được phát ngôn sẽ được thông dịch lại để xác định nó thật sự có nghĩa là gì.
* Thời đại hiện hay máy tính ngày càng thông minh, có thể giải quyết các vấn đề con người đặt ra. Trong đó mảng xử lý ngôn ngữ tự nhiên ngày càng được quan tâm.
* [Nhận dạng chữ viết](https://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%ADn_d%E1%BA%A1ng_ch%E1%BB%AF_vi%E1%BA%BFt&action=edit&redlink=1): Có hai kiểu nhận dạng, thứ nhất là nhận dạng chữ in, ví dụ nhận dạng chữ trên sách giáo khoa rồi chuyển nó thành dạng văn bản điện tử như dưới định dạng doc của [Microsoft Word](https://vi.wikipedia.org/wiki/Microsoft_Word) chẳng hạn. Phức tạp hơn là [nhận dạng chữ viết tay](https://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%ADn_d%E1%BA%A1ng_ch%E1%BB%AF_vi%E1%BA%BFt_tay&action=edit&redlink=1), có khó khăn bởi vì chữ viết tay không có khuôn dạng rõ ràng và thay đổi từ người này sang người khác. Với chương trình nhận dạng chữ viết in có thể chuyển hàng ngàn đầu sách trong thư viện thành văn bản điện tử trong thời gian ngắn. Nhận dạng chữ viết của con người có ứng dụng trong khoa học hình sự và bảo mật thông tin (nhận dạng chữ ký điện tử).
* [Nhận dạng tiếng nói](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADn_d%E1%BA%A1ng_ti%E1%BA%BFng_n%C3%B3i): Nhận dạng tiếng nói rồi chuyển chúng thành văn bản tương ứng. Giúp thao tác của con người trên các thiết bị nhanh hơn và đơn giản hơn, chẳng hạn thay vì gõ một tài liệu nào đó bạn đọc nó lên và trình soạn thảo sẽ tự ghi nó ra. Đây cũng là bước đầu tiên cần phải thực hiện trong ước mơ thực hiện giao tiếp giữa con người với robot. Nhận dạng tiếng nói có khả năng trợ giúp người khiếm thị rất nhiều.
* [Tổng hợp tiếng nói](https://vi.wikipedia.org/wiki/T%E1%BB%95ng_h%E1%BB%A3p_gi%E1%BB%8Dng_n%C3%B3i): Từ một văn bản tự động tổng hợp thành tiếng nói. Thay vì phải tự đọc một cuốn sách hay nội dung một trang [web](https://vi.wikipedia.org/wiki/World_Wide_Web), nó tự động đọc cho chúng ta. Giống như nhận dạng tiếng nói, tổng hợp tiếng nói là sự trợ giúp tốt cho [người khiếm thị](https://vi.wikipedia.org/wiki/Ng%C6%B0%E1%BB%9Di_khi%E1%BA%BFm_th%E1%BB%8B), nhưng ngược lại nó là bước cuối cùng trong giao tiếp giữa [robot](https://vi.wikipedia.org/wiki/Robot) với người.
* [Dịch tự động](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_t%E1%BB%B1_%C4%91%E1%BB%99ng) (*machine translate*): Như tên gọi đây là chương trình dịch tự động từ ngôn ngữ này sang ngôn ngữ khác. Một phần mềm điển hình về tiếng Việt của chương trình này là [Evtrans](https://vi.wikipedia.org/w/index.php?title=Evtrans&action=edit&redlink=1) của Softex, dịch tự động từ tiếng Anh sang tiếng Việt và ngược lại, phần mềm từng được trang web vdict.com mua bản quyền, đây cũng là trang đầu tiên đưa ứng dụng này lên mạng. Tháng 10 năm 2008 có hai công ty tham gia vào lĩnh vực này cho ngôn ngữ tiếng Việt là công ty Lạc Việt (công ty phát hành từ điển Lạc Việt) và [Google](https://vi.wikipedia.org/wiki/Google), một thời gian sau đó Xalo\_vn cũng đưa ra dịch vụ tương tự.
* [Tìm kiếm thông tin](https://vi.wikipedia.org/wiki/T%C3%ACm_ki%E1%BA%BFm_th%C3%B4ng_tin) (*information retrieval*): Đặt câu hỏi và chương trình tự tìm ra nội dung phù hợp nhất. Thông tin ngày càng đầy lên theo cấp số nhân, đặc biệt với sự trợ giúp của [internet](https://vi.wikipedia.org/wiki/Internet) việc tiếp cận thông tin trở lên dễ dàng hơn bao giờ hết. Việc khó khăn lúc này là tìm đúng nhất thông tin mình cần giữa bề bộn tri thức và đặc biệt thông tin đó phải đáng tin cậy. Các [máy tìm kiếm](https://vi.wikipedia.org/wiki/M%C3%A1y_truy_t%C3%ACm_d%E1%BB%AF_li%E1%BB%87u) dựa trên giao diện web như Google hay [Yahoo](https://vi.wikipedia.org/wiki/Yahoo!) hiện nay chỉ phân tích nội dung rất đơn giản dựa trên tần suất của từ khoá và thứ hạng của trang và một số tiêu chí đánh giá khác để đưa ra kết luận, kết quả là rất nhiều tìm kiếm không nhận được câu trả lời phù hợp, thậm chí bị dẫn tới một liên kết không liên quan gì do thủ thuật đánh lừa của các trang web nhằm giới thiệu sản phẩm (có tên tiếng Anh là [SEO](https://vi.wikipedia.org/wiki/T%E1%BB%91i_%C6%B0u_h%C3%B3a_c%C3%B4ng_c%E1%BB%A5_t%C3%ACm_ki%E1%BA%BFm) viết tắt của từ *search engine optimization*). Thực tế cho đến bây giờ chưa có máy tìm kiếm nào hiểu được ngôn ngữ tự nhiên của con người trừ trang www.ask.com được đánh giá là "hiểu" được những câu hỏi có cấu trúc ở dạng đơn giản nhất. Mới đây cộng đồng mạng đang xôn xao về trang [Wolfram Alpha](https://vi.wikipedia.org/wiki/Wolfram_Alpha), được hứa hẹn là có khả năng hiểu ngôn ngữ tự nhiên của con người và đưa ra câu trả lời chính xác. Lĩnh vực này hứa hẹn tạo ra bước nhảy trong cách thức tiếp nhận tri thức của cả cộng đồng.
* [Tóm tắt văn bản](https://vi.wikipedia.org/w/index.php?title=T%C3%B3m_t%E1%BA%AFt_v%C4%83n_b%E1%BA%A3n&action=edit&redlink=1): Từ một văn bản dài tóm tắt thành một văn bản ngắn hơn theo mong muốn nhưng vẫn chứa những nội dung thiết yếu nhất.
* [Khai phá dữ liệu](https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u) (*data mining*) và [phát hiện tri thức](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A1t_hi%E1%BB%87n_tri_th%E1%BB%A9c&action=edit&redlink=1): Từ rất nhiều tài liệu khác nhau phát hiện ra tri thức mới. Thực tế để làm được điều này rất khó, nó gần như là mô phỏng quá trình [học tập](https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_t%E1%BA%ADp), khám phá khoa học của con người, đây là lĩnh vực đang trong giai đoạn đầu phát triển. Ở mức độ đơn giản khi kết hợp với [máy tìm kiếm](https://vi.wikipedia.org/wiki/M%C3%A1y_truy_t%C3%ACm_d%E1%BB%AF_li%E1%BB%87u) nó cho phép đặt câu hỏi để từ đó công cụ tự tìm ra câu trả lời dựa trên các thông tin trên web mặc cho việc trước đó có câu trả lời lưu trên web hay không (giống như trang Yahoo! hỏi và đáp, nơi chuyên đặt các câu hỏi để người khác trả lời), nói một cách nôm na là nó đã biết xử lý dữ liệu để trả lời câu hỏi của người sử dụng, thay vì máy móc đáp trả những gì chỉ có sẵn trong bộ nhớ.

### Phương pháp giải quyết

* Machine Learning và lý thuyết xác suất có một sự liên hệ rất khăng khít. Các phương pháp phân loại dựa trên lý thuyết xác suất về cơ bản có thể hiểu là việc tính xem xác suất một sự việc của chúng ta sẽ xảy ra theo hướng như thế nào. Xác suất của hướng nào càng cao thì khả năng sự việc xảy ra theo hướng đó càng nhiều. Điều này đặc biết có ý nghĩa trong bài toán dự đoán và phân lớp của lĩnh vực Machine Learning. Một câu hỏi đặt ra là: Vậy rốt cuộc thì chiếc máy tính vô hồn của chúng ta sẽ xác định xác suất đó như thế nào ???. Theo thống kê học hiện đại thì tương ứng với mỗi bài toán giải quyết theo phương pháp xác suất thường đi kèm theo một phân phối xác suất phù hợp với bài toán đó. Tương ứng với mỗi phân phối xác suất chúng ta có một cách tính riêng các đại lượng cần thiết cho quá trình chạy các thuật toán như kỳ vọng, độ lệch chuẩn ... mà chúng ta sẽ cùng nhau tìm hiểu và tiến hành tính toán trong suốt phạm vi bài viết này. Giờ thì chúng ta tiếp tục đến với phần tiếp theo thôi.
* Không còn quá xa lạ với chúng ta nữa rồi. Nó chính là sự liên hệ giữa các xác suất có điều kiện. Điều đó gợi ý cho chúng ta rằng chúng ta có thể tính toán một xác suất chưa biết dựa vào các xác suất có điều kiện khác. Thuật toán Naive Bayes cũng dựa trên việc tính toán các xác suất có điều kiện đó. Nghe tên thuật toán là đã thấy gì đó ngây ngô rồi. Tại sao lại là Naive nhỉ. Không phải ngẫu nhiên mà người ta đặt tên thuật toán này như thế. Tên gọi này dựa trên một giả thuyết rằng các chiều của dữ liệu *X*=(*x*1​,*x*2​,....,*xn*​) là độc lập về mặt xác suất với nhau.



Chúng ta có thể thấy rằng giả thuyết này có vẻ khá ngây thơ vì trên thực tế điều này có thể nói là không thể xảy ra tức là chúng ta rất ít khi tìm được một tập dữ liệu mà các thành phần của nó không liên quan gì đến nhau. Tuy nhiên, giả thiết ngây ngô này lại mang lại những kết quả tốt bất ngờ. Giả thiết về sự độc lập của các chiều dữ liệu này được gọi là Naive Bayes (xin phép không dịch). Cách xác định class của dữ liệu dựa trên giả thiết này có tên là Naive Bayes Classifier (NBC). Tuy nhiên dựa vào giả thuyết này mà bước training và testing trở nên vô cùng nhanh chóng và đơn giản. Chúng ta có thể sử dụng nó cho các bài toán large-scale. Trên thực tế, NBC hoạt động khá hiệu quả trong nhiều bài toán thực tế, đặc biệt là trong các bài toán phân loại văn bản, ví dụ như lọc tin nhắn rác hay lọc email spam. Trong bài viết này mình sẽ cùng với các bạn áp dụng lý thuyết về NBC để giải quyết một bài toán mới đó chính là bài toán chuẩn đoán bệnh tiểu đường

## Hiện trạng cơ sở vật chất và con người

### Cơ sở vật chất

Máy tính ngày càng giúp con người giải quyết những vấn đề mà con người không thể giải quyết được và muốn máy tính hiểu ngôn ngữ của con người thì chúng tôi đã tạo ra phần mềm giúp xử lý ngôn ngữ tự nhiên trên máy tính, giúp máy tính có thể hiểu ngôn ngữ tự nhiên.

### Con người

Thời đại hiện nay, công nghệ thông tin ngày càng phát triển và sản phẩm của sự phát triển đó là những chiếc máy tính thông minh được ra đời và ngày càng hoàn thiện hơn. Máy tính muốn được sử dụng rộng rãi thì ngôn ngữ máy tính phải thật gần gũi, giúp người dùng dễ dàng thao tác và sử dụng thì cần phải có các công cụ hỗ trợ để xử lý ngôn ngữ tự nhiên trên máy tính. Giúp máy tính hiểu được ngôn ngữ tự nhiên cảu người dùng và thực hiện yêu cầu người dùng.

## Yêu cầu sơ bộ về phần mềm

### Tin học

* Phần mềm thích hợp trên gần như tất cả các máy tính từ cấu hình trung bình đến cấu hình cao.
* Phần cứng không đòi hỏi quá cao.
* Cấu hình mẫu:
* Processor: Intel (R) Core (TM) i5-6200U CPU @ 2.30GHz 2.40GHz
* RAM: 4GB
* Operating System: Windows 10 Home,
* VGA: AMD Radeon R7 M360 (Total memory: 6088MB)
* Công cụ:
* Netbean IDE 8.2
* Ngôn ngữ lập trình:
* Java

### Con người

* Ngôn ngữ chính: Tiếng Anh
* Giao diện: Dễ nhìn, dễ sử dụng.

## Yêu cầu đồ án

Đồ án yêu cầu xây dựng phân tích một e-mail đầu vào có phải là thư rác hay không dựa vào bộ huấ luyện.

Bộ huấn luyện:

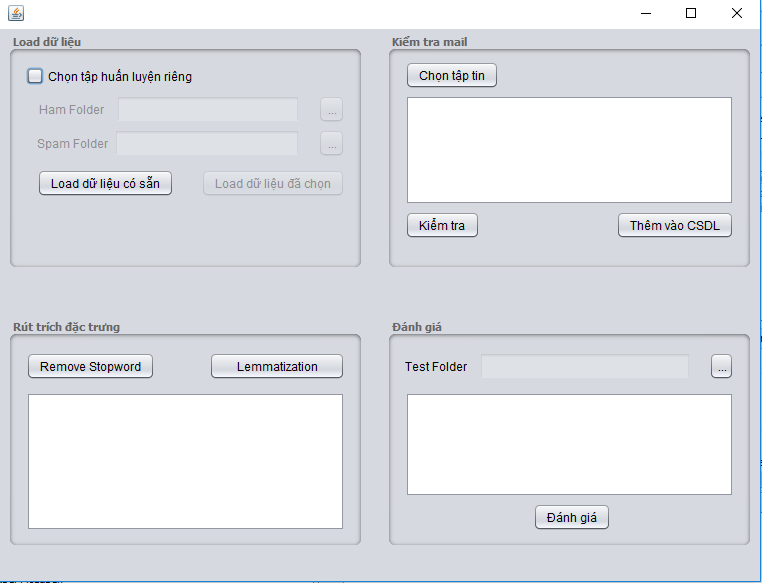
1. Thư mục Ham: chứa các e-mail quan trọng, không phải thư rác
2. Thư mục Spam: chứa các e-mail rác (quảng cáo, virus, lừa đảo .. )

* Chương trình nhận input đầu vào là một e-mail dạng text (file \*.txt)
* Chương trình phân tích và xác định e-mail có phải là thư rác không
* Chương trình sử dụng thuật toán phân lớp Naïve Bayes để tính toán xác suất và cho ra kết quả

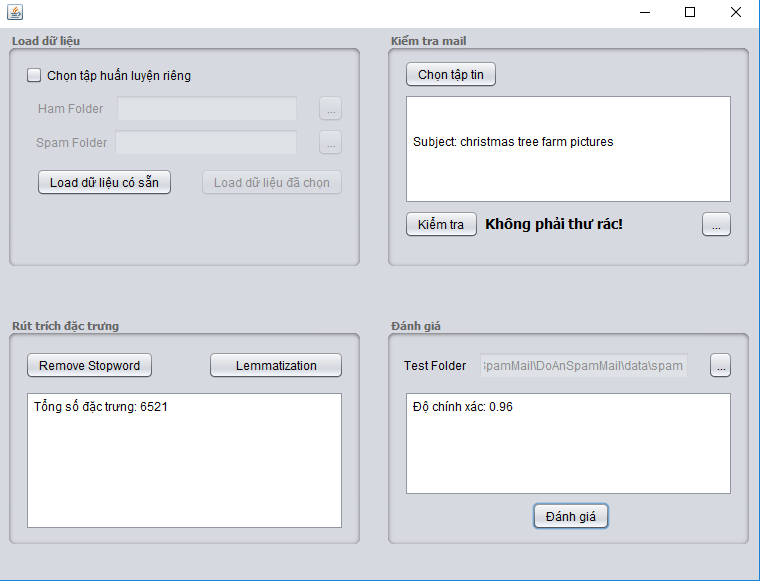
# THIẾT KẾ

## Thiết kế giao diện chính

### Giao diện ban đầu

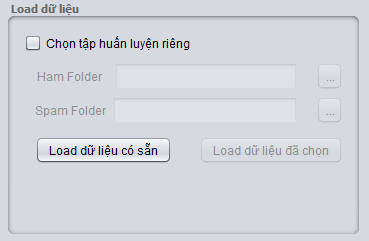


### Giao diện chạy chương trình



## Các nút lệnh và thao tác

### Phần Load dữ liệu



Có thể sử dụng dữ liệu có sẵn => nhấn nút Load dữ liệu có sẵn

Hoặc Chọn tập huấn luyện riêng

### Phần Kiểm tra mail

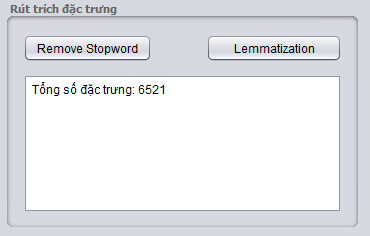


Button Chọn tập tin: chọn đường dẫn đến tập tin cần kiểm tra

Button Kiểm tra: chạy chương trình và xuất kết quả có phải thư rác hay không

Button góc dưới bên phải: thêm tập tin đã chọn vào tập huấn luyện

### Phần Rút trích đặc trưng

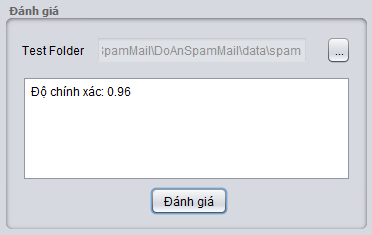


Lựa chọn đặc trưng từ bộ huấn luyện

Button Remove Stopword: Xóa các Stopword

Button lemmatization: rút trích đặc trưng bằng kĩ thuật Lemmatization

### Phần Đánh giá



Button góc trên bên phải: chọn đường dẫn đến thư mục chứa các tập tin cần kiểm tra để tính độ chính xác

Button Đánh giá: chạy chương trình tính toán và xuất ra độ chính xác của phần mềm.

# CÀI ĐẶT PHẦN MỀM

## Ngôn ngữ lập trình Java

### Sự phổ biến

Tính đến ngày 9 tháng 12 năm 2017, Java là ngôn ngữ lập trình phổ biến nhất thế giới với tỉ lệ 13,268%, nhưng đã giảm 4,59% so với cùng kì năm trước. Java và C vẫn luôn chiếm hai vị trí đầu tiên trong bảng các ngôn ngữ lập trình phổ biến nhất suốt 20 năm qua, nhưng Java là ngôn ngữ duy nhất giữ được tỉ lệ trên 10%, trong khi C đã tụt xuống 6,477% vào ngày 2/8/2017.

### Khả năng của ngôn ngữ Java

Là một ngôn ngữ bậc cao như C, C++, Perl, SmallTalk,.. cho nên có thể được dùng để tạo ra các ứng dụng để giải quyết các vấn đề về số, xử lý văn bản, tạo ra trò chơi, và nhiều thứ khác.

Có các môi trường lập trình đồ họa như Visual Java, Symantec Cafe, Jbuilder, Jcreator, ...

Có khả năng truy cập dữ liệu từ xa thông qua cầu nối JDBC (Java DataBase Connectivity)

Hỗ trợ các lớp hữu ích, tiện lợi trong lập trình các ứng dụng mạng (Socket) cũng như truy xuất Web.

Hỗ trợ lập trình phân tán (Remote Method Invocation ) cho phép một ứng dụng có thể được xử lý phân tán trên các máy tính khác nhau.

Và luôn được bổ sung các tính năng cao cấp khác trong các phiên bản sau.

### Những đặc điểm của ngôn ngữ Java

Đơn giản, hướng đối tượng và quen thuộc.

Mạnh mẽ và an toàn.

Kiến trúc trung lập và di động.

Thực thi với hiệu suất cao.

Dịch ra bytecode, phân luồng và năng động.

Dễ sử dụng cho người dùng Java

### Máy ảo Java (JMV - Java Virtual Machine)

Để đảm bảo tính đa nền, Java sử dụng cơ chế Máy ảo của Java. ByteCode đó là ngôn ngữ máy của Máy ảo Java tương tự như các lệnh nhị phân của các máy tính thực. Một chương trình sau khi được viết bằng ngôn ngữ Java (có phần mở rộng là .java) phải được biên dịch thành tập tin thực thi được trên máy ảo Java (có phần mở rộng là .class). Tập tin thực thi này chứa các chỉ thị dưới dạng mã Bytecode mà máy ảo Java hiểu được phải làm gì.

Khi thực hiện một chương trình, máy ảo Java lần lượt thông dịch các chỉ thị dưới dạng Bytecode thành các chỉ thị dạng nhị phân của máy tính thực và thực thi thực sự chúng trên máy tính thực.

Máy ảo thực tế đó là một chương trình thông dịch. Vì thế các hệ điều hành khác nhau sẽ có các máy ảo khác nhau. Để thực thi một ứng dụng của Java trên một hệ điều hành cụ thể, cần phải cài đặt máy ảo tương ứng cho hệ điều hành đó.

### Hai kiểu ứng dụng dưới ngôn ngữ java

**Applet** chạy trên browsers và là chương trình nhỏ hơn so với application, hơn nữa **Applet** không có phương thức main như application.

**Applet** là công cụ mạnh mẽ, là cách tốt nhất để tạo ra một chương trình java. Có ít lập trình viên có kinh nghiệm lập trình ứng dụng! Điều này không có nghĩa là applications không có đất dụng võ, mà mỗi cái (Applet và application) đều có tầm quan trọng như nhau trong mảng của mỗi cái. Và có điều chung nữa là cả hai đều không phụ thuộc platform 🙂

**Applet** được dùng để xử lí các vấn đề phía người dùng (client), còn application giải quyết cả vấn đề của client và server.

**Application** và **applet** có nhiều điểm tương đồng như cùng có chung nhiều đặc điểm và cùng chia sẻ tài nguyên với nhau. **Applet** được tạo ra bằng cách kế thừa lớp java.applet.**Applet** còn **Application** được thực thi bằng cách gọi hàm main.

### Bộ phát triển ứng dụng Java (JDK- Java Development Kit)

**JDK** là một bộ công cụ cho phép người lập trình phát triển và triển khai các ứng dụng bằng ngôn ngữ java được cung cấp miễn phí bởi công ty JavaSoft (hoặc Sun). Có các bộ Jdk cho các hệ điều hành khác nhau. Các ấn bản của JDK không ngừng được phát hành, các bạn có thể tải về từ địa chỉ http://java.sun.com hoặc http://www.javasoft.com

Bộ công cụ này gồm các chương trình thực thi đáng chú ý sau:

**javac**: Chương trình biên dịch các chương trình nguồn viết bằng ngôn ngữ java ra các tập tin thực thi được trên máy ảo Java.

**java**: Đây là chương trình làm máy ảo của Java, thông dịch mã Bytecode của các chương trình kiểu application thành mã thực thi của máy thực.

**appletviewer**: Bộ thông dịch, thực thi các chương trình kiểu applet.

**javadoc**: Tạo tài liệu về chú thích chương trình nguồn một cách tự động.

**jdb**: Trình gở rối.

**rmic**: Tạo Stub cho ứng dụng kiểu RMI.

rmiregistry: Phục vụ danh bạ (Name Server) trong hệ thống RMI

## Cài đặt

### Phương pháp phân tích e-mail

Áp dụng phương pháp xác suất thống kê và học máy: Naïve Bayes

Naïve Bayes là phương pháp phân lớp dựa trên xác suất có điều kiện. Phương pháp này dựa trên các đặc trưng của phần tử cần phân lớp. Đối với phần tử là một văn bản, đặc trưng có thể là các ký hiệu phân tách nhau bằng khoảng trắng.

Phương pháp phân lớp Naive Bayes chỉ áp dụng với số phân lớp là 2.

**Ví dụ:**

Cho một phần tử d và 2 lớp c1, c2 . Giả sử phần tử d có các đặc trưng xk , k = 1, n

Khi đó, xác suất để phần tử d thuộc lớp ci là :

P (d | ci ) = P (x1 , x2 , x3 , .., xn | ci )

= P(x1 | ci ) \* P(x2 | ci ) \* ... \* P(xn | ci )

**Trong đó cần xác định:**

P(ci ) = (số phần tử của lớp ci) / (tổng số phần tử của cả 2 lớp)

P (xk | ci ) = (số lần xuất hiện xk trong ci ) / (số lần xuất hiện của tất cả đặc trưng của d trong ci )

Do có những đặc trưng x không xuất hiện trong các phần tử của lớp c1 nhưng lại có trong những phần tử của lớp c2. Khi đó P(x | c1 ) = 0 nên cần dùng ước lượng Laplace. Khi đó:

P (xk | ci ) = ((số lần xuất hiện xk trong ci ) + 1) / ((số lần xuất hiện của tất cả đặc trưng trong ci )+nV )

Với nV là số lượng các đặc trưng x trong tất cả các phần tử của tất cả các lớp.

Sử dụng công thức xác suất có điều kiện dựa trên các ước lượng đã để chọn lớp có xác suất cao nhất.

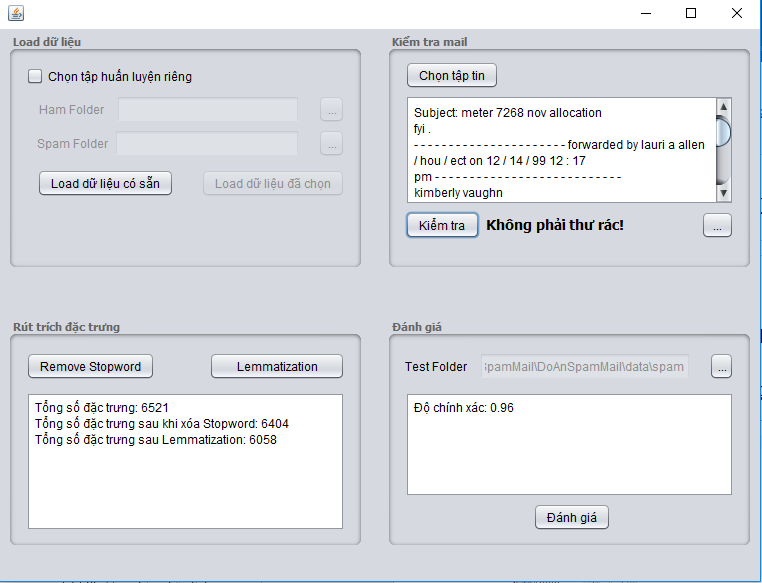
### Phương pháp lựa chọn đặc trưng

**Loại bỏ Stopword**: bỏ những từ không quan trọng, không có nghĩa

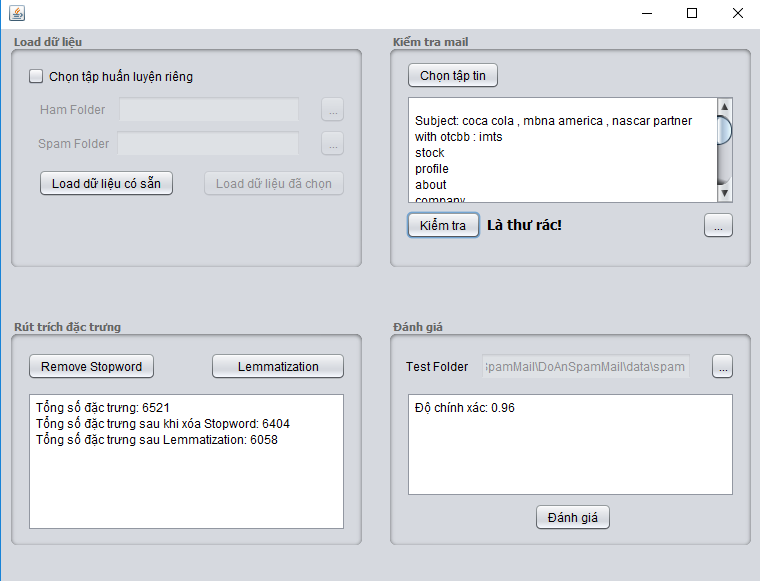
**Phân loại từ**: cố Gắng loại bỏ bớt các từ thuộc một nhóm nào đó có ý nghĩa tương tự nhau : ví dụ: chữ số (number), Nhãn hiệu máy tính(brand”apple,Dell,Asus,Hp,Acer,..”),vùng miền(Area”Châu á, lục địa âu,..”), quốc gia(thay bằng Nation”Việt nam, trung quốc , …”)

# KẾT QUẢ lọc spam mail bằng phương pháp phân lớp naïve bayes

## Ví dụ 1: Phân tích Mail thường



## Ví dụ 2: Phân tích Mail rác



# tỔNG KẾT

## Kết quả đạt được

* Giao diện dễ nhìn, người dùng dễ sử dụng.
* Chương trình xử lí bộ huấn luyện là tập các mail Ham (mail quan trọng cần đọc) và các mail Spam (mail rác).
* Chương trình nhận Input đầu vào là một e-mail cần xác định là thư rác hay không.
* Chương trình sử dụng phương pháp phân lớp Naïve Bayes để tính toán xác suất, so sánh và xuất kết quả.
* Chương trình tính toán các độ đo để kiểm tra độ chính xác (Precision, Recall, ..).

## Hạn chế

* Chỉ xử lí dữ liệu là các file text Tiếng Anh.

## Hướng phát triển

* Xây dựng chương trình phân tích cho cả Tiếng Anh và Tiếng Việt
* Kết hợp nhiều phương pháp phân loại để nâng cao tính chính xác của chương trình

# TÀI LIỆU THAM KHẢO

[1]" Ngôn Ngữ Lập Trình Java", *Viẹtack*. [Online].

Available: <http://vietjack.com/java/>.

[2]"Chương Trình Lọc Thư Rác Bằng Java", V*iblo*, 2007. [Online].

Available: <https://viblo.asia/p/chuong-trinh-loc-thu-rac-bang-java-code-vi-du-RnB5p7yblPG>

# LINK GITHUB

<https://github.com/tradaik/Java>