TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

——————— * ———————

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

NGÀNH CÔNG NGHỆ THÔNG TIN

NGHIÊN CỨU PHƯƠNG PHÁP TÓM TẮT ĐƠN VĂN BẢN TỰ ĐỘNG SỬ DỤNG MÔ HÌNH SVM ƯỚC LƯỢNG XÁC SUẤT

Sinh viên thực hiện : **TRẦN THỊ DIỆU LINH**
Lớp: **VIỆT NHẬT B – K58**
Giảng viên hướng dẫn : **PGS.TS LÊ THANH HƯƠNG**

HÀ NỘI 5-2018

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên:

   Họ và tên sinh viên: Trần Thị Diệu Linh

   Điện thoại liên lạc: 01675408520 Email: linhttd.bk@gmail.com

   Lớp: IS1 - k58 Hệ đào tạo: Kỹ sư

   Đồ án tốt nghiệp được thực hiện tại: Trường Đại học Bách Khoa Hà Nội

   Thời gian làm ĐATN: Từ 01/2018 đến 05/2018

2. Mục đích nội dung của ĐATN:
   - Tìm hiểu về các phương pháp tóm tắt văn bản.
   - Nghiên cứu phương pháp tóm tắt văn bản tự động sử dụng mô hình SVM ước lượng xác suất.
   - Đánh gía nhận xét kết quả.
   - Đề xuất phương pháp cải tiến chất lượng tóm tắt văn bản tự động.
3. Các nhiệm vụ cụ thể của ĐATN
   - Tìm hiểu các công nghệ:
     - Nền tảng Python.
     - Tìm hiểu các thư viện SVM.
     - Tìm hiểu các công cụ xử lý ngôn ngữ tự nhiên.
   - Tìm hiểu lý thuyết:
     - Lý thuyết các thuật toán machine learning cơ bản.
     - Các phương pháp xử lý ngôn ngữ tự nhiên.
     - Thuật toán tính toán các đặc trưng của câu trong văn bản ngôn ngữ tự nhiên.
     - Tìm hiểu, phân tích cấu trúc một số bộ dữ liệu huấn luyện phù hợp cho bài toàn tóm tắt văn bản tự động.
     - Thuật toán SVM.
   - Cài đặt và xây dựng hệ thống tóm tắt văn bản theo hướng trích rút câu quan trọng
     - Xây dựng các chương trình tiền xử lý ngôn ngữ tự nhiên.
     - Xây dựng chương trình tính toán các đặc trưng của câu.
     - Chương trình mô hình hóa dữ liệu.
     - Cài đặt thuật toán SVM(support vector machine) huấn luyện dữ liệu và kiểm thử.
     - Sử dụng thư viện đánh giá ROUGE.
4. Lời cam đoan của sinh viên:

   Em – Trần Thị Diệu Linh - cam kết ĐATN là công trình nghiên cứu của bản thân em dưới sự hướng dẫn của PGS.*TS Lê Thanh Hương*.

   Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

Hà Nội, ngày 26 tháng 05 năm 2018
Tác giả ĐATN

Trần Thị Diệu Linh

5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

Hà Nội, ngày tháng năm
Giáo viên hướng dẫn

PGS.TS Lê Thanh Hương

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Trong suốt lịch sử, số lượng thông tin ngày một tăng và có quá ít thời gian đọc thông tin luôn luôn là hai trở ngại lớn trong việc tìm kiếm kiến thức. Như vậy, việc xác định thông tin quan trọng trong một văn bản là một vấn đề vô cùng cần thiết. Để giải quyết vấn đề quá tải thông tin và dư thừa thông tin, giúp chúng ta có thể xác định nhanh chóng và hiệu quả các thông tin mà mình cần, có khá nhiều cách tiếp cận đã được thực hiện, trong đó tóm tắt văn bản tự động giúp giải quyết khá tốt vấn đề trên.

Đã có nhiều phương pháp tóm tắt văn bản được đề xuất, tuy nhiên, mỗi phương pháp đều có điểm mạnh và điểm yếu riêng.Trong phạm vi bài toàn tóm tắt đơn văn bản theo hướng trích rút, phương pháp học máy có giám sát SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại với số chiều có thể cực lớn. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn. Việc ứng dụng phương pháp SVM rất phù hợp cho bài toán tóm tắt văn bản theo hướng trích rút.

Đồ án trình bày về quá trình nghiên cứu phương pháp tóm tắt văn bản tự động sử dụng mô hình SVM ước lượng xác suất và xây dựng chương trình thử nghiệm cho phương pháp này.

Trong thời gian thực hiện đồ án, dưới sự hướng dẫn của **PGS.TS Lê Thanh Hương**, em đã được tìm hiểu về các thuật toán như SVM, phương pháp, thuật toán xử lý ngôn ngữ tự nhiên, nền tảng Phython và các thư viện mã nguồn mở để xây dựng hệ thống thử nghiệm cho phương pháp tóm tắt văn bản theo hướng trích rút sử dụng SVM của mình.

Nội dung đồ án được trình bày theo các phần chính sau:

Chương I: Mở đầu

Đặt vấn đề và giới thiệu tổng quan về bài toán.

Chương II: Tổng quang về bài toán tóm tắt văn bản

Chi tiết bài toán tóm tắt văn bản tự động theo hướng trích rút, giới thiệu một số phương pháp và hướng tiếp cận bài toán.

Đề xuất mô hình xây dựng chương trình thử nghiệm.

Chương III: Xây dựng mô hình đề xuất

Phân tích bộ dữ liệu đã nghiên cứu, các bước xây dựng mô hình thử nghiệm.

Chương IV: Kết quả và đánh giá sản phẩm

Kết quả đánh giá của chương trình phân tích bộ dữ liệu học máy và phương pháp SVM áp dụng trong bài toàn đặt ra.

Chương V: Kết luận và hướng phát triển

Trình bày những mục tiêu đã đạt được, những đóng góp của đồ án, những hạn chế chưa được khắc phục và hướng phát triển trong tương lai.

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc nhất tới PGS.TS Lê Thanh Hương, cô luôn tận tình hướng dẫn, định hướng nghiên cứu cho em trong suốt 3 kỳ thực hiện khóa luận tốt nghiệp.

Em xin chân thành cảm ơn các thầy cô giáo trong viện CNTT & TT, trong 5 năm học tập tại trường đại học Bách Khoa Hà Nội đã truyền đạt kiến thức về cả chuyên môn và đời sống, tạo cho em những điều kiện học tập tốt nhất.

Cuối cùng, em muốn gửi lời cảm ơn tới gia đình và bạn bè, những người thân yêu luôn bên cạnh ủng hộ và động viên em trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp

Mặc dù đã cố gắng hoàn thành đồ án trong phạm vi và khả năng cho phép nhưng em chắc chắn sẽ không tránh khỏi nhiều thiếu sót. Vì vậy em kính mong nhận được sự thông cảm của quý Thầy Cô và các bạn.

Em xin chân thành cảm ơn !

MỤC LỤC

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH ẢNH

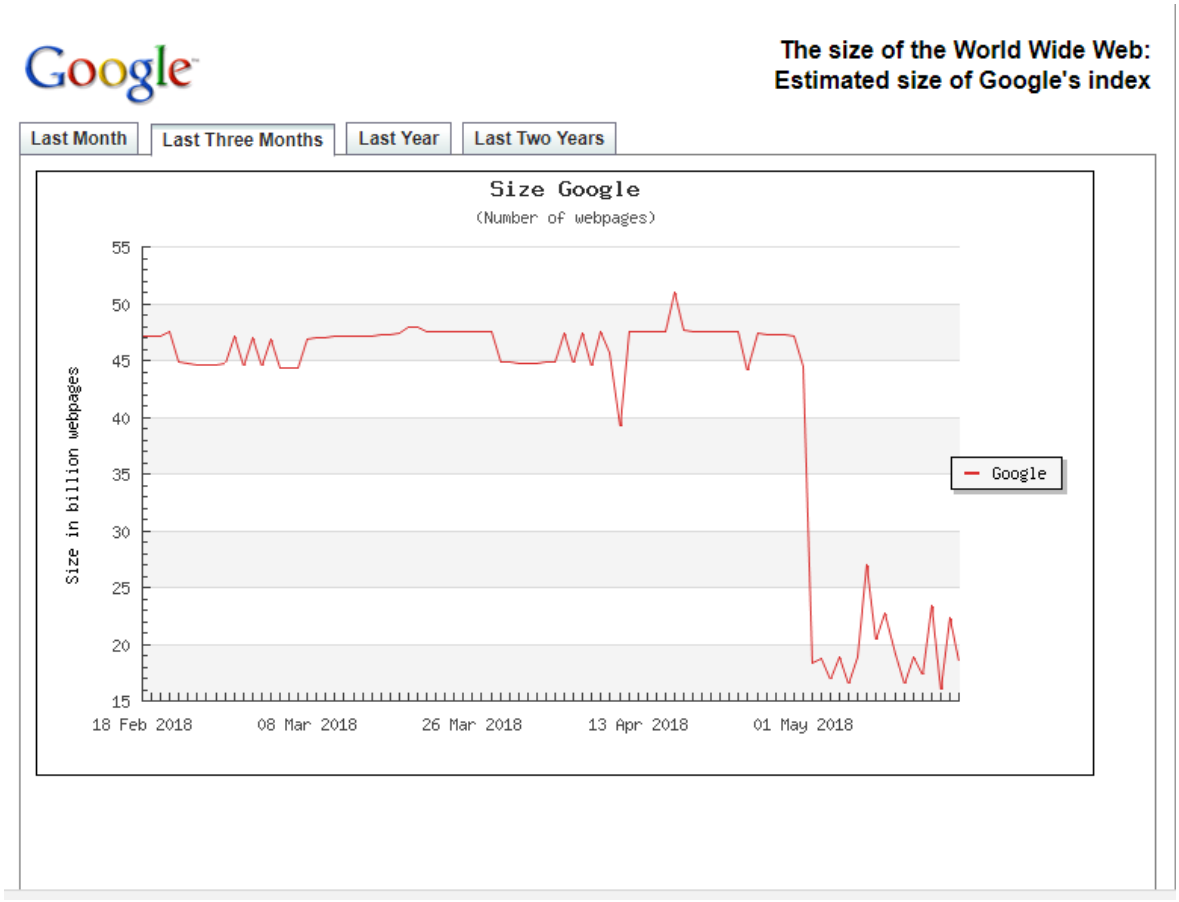
DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ đầy đủ	Ý nghĩa
SVM	Support vector machine	Máy vector hỗ trợ
SCU	Summary Content Unit	Đơn vị nội dung tóm tắt
DUC	Document Understanding Conference	Hội thảo về tài liệu
NIST	National Institute of Standards and Technology	Viện Tiêu chuẩn và Công nghệ Quốc gia
WWW	World Wide Web	Mạng lưới toàn cầu: không gian thông tin toàn cầu
TTVB	Tóm tắt văn bản	
ROUGE	Recall-Oriented Understudy for Gisting Evaluation	
RNN	Recurrent Neural Network	Mạng nơ ron đệ quy
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên

CHƯƠNG I: MỞ ĐẦU

1. Đặt vấn đề

Trong những năm gần đây, chúng ta được chứng kiến sự phát triển như vũ bão của World-Wide-Web. Theo thống kê của WorldWideWebSize.com vào cuối tháng 4 năm 2018 có khoảng 50 tỷ trang web đã được indexed bởi Google, khoảng 2500TB dữ liệu trên Web.



Hình 1: Số lượng trang Web được indexed bởi Google đầu năm 2018

Trước sự phát triển đó thì vấn đề đặt ra là *làm thế nào để con người có thể sử dụng một cách hiệu quả lượng thông tin khổng lồ trên Internet?*

Việc tóm tắt thông tin giúp ta có thể quyết định xem tiếp tục tập trung vào phần nào, nhất là trong các văn bản phức tạp như bài báo khoa học hay toàn bộ nội dung một cuốn sách. Ngoài ra nó còn có thể ứng dụng trong rất nhiều các lĩnh vực khác mà con người cần phải tóm tắt một lượng rất lớn các dữ liệu như tài chính, dữ liệu thuốc của bệnh nhân trong y học.

Bài toán tóm tắt văn bản là một trong những bài toán kinh điển trong lĩnh vực xử lý dữ liệu văn bản. Xử lý dữ liệu văn bản bao gồm:

- Kiểm tra lỗi chính tả (spelling-checker)
- Kiểm tra lỗi văn phạm (grammar-checker)
- Từ điển đồng nghĩa (thesaurus)
- Phân tích văn bản (text analyzer)
- Phân loại văn bản (text classification)
- **Tóm tắt văn bản (text summarization)**
- Tổng hợp tiếng nói (speech synthesis)
- Nhận dạng giọng nói (speech recognization)
- Dịch tự động (automatic translation)
- …

Tóm tắt văn bản là công việc phân tích nội dung của văn bản và sau đó sinh ra một văn bản tóm tắt có kích thước nhỏ hơn văn bản ban đầu, loại bỏ đi những thông tin không quan trọng nhưng vẫn đảm bảo giữ được những nội dung cốt lõi của văn bản. Do đó để công việc tóm tắt văn bản chính xác cần phải đáp ứng được các yêu cầu sau:

- Các văn bản khi phân tích thì phải "hiểu" được nội dung để xác định được các tiêu chuẩn trong văn bản.
- Các văn bản tóm tắt cần được kiểm tra bằng một thang đo tiêu chuẩn.

Rõ ràng việc tóm tắt văn bản chính là công việc khai phá dữ liệu văn bản (text data mining).

2. Lịch sử phát triển của tóm tắt văn bản

Tóm tắt văn bản bắt đầu từ những năm cuối thập kỉ 1950 với nghiên cứu của Luhn(1958) dựa trên tần số từ. Ý tưởng cơ bản của phương pháp tần số từ dựa trên kiến thức cho rằng tần số của từng từ trong văn bản là một độ đo hữu dụng để đánh giá tầm quan trọng của chúng.

Tiếp theo đó là phương pháp tóm tắt dựa trên vị trí của các câu trong văn bản của Baxendale (1958) và những nghiên cứu của Edmundson(1969) về vị trí của các câu trong văn bản và các từ/cụm từ mang ý nghĩa tổng quát. Theo đó, những câu bắt đầu và kết thúc của đoạn văn bài viết hay những câu chưa những từ như 'important' (đặc biệt), 'result are' (kết qủa là) … là những câu có ý nghĩa quan trọng.

Đầu những năm 1970, tiếp tục có những nghiên cứu với hướng tiếp cận ngoài (sử dụng các cụm từ dấu hiệu) và được ứng dụng trong các phần mềm thương mại

Những năm 1980, phát triển nhiều nghiên cứu với nhiều hướng khác nhau, đặc biệt là hướng tiếp cận mức thực thể dựa trên trí tuệ nhân tạo như sử dụng script (Lehnert 1981), các luật sản xuất mạng và logic (Fum 1985), mạng ngữ nghĩa (Reimer và Hahn 1988) cũng như các hướng tiếp cận kết hợp (Rau 1989) hay (Aretoulaki 1994).

Willam B. Cavnar (1994) : biểu diễn văn bản dựa trên n-gram thay cho cách biểu diễn truyền thống bằng từ khoá.

Chinatsu Anoe (1997) đã phát triển hệ DimSum để tóm tắt văn bản sử dụng xử lý ngôn ngữ tự nhiên và kĩ thuật thống kê dựa trên hệ thống tf-idf, sử dụng WordNet để xem xét ngữ nghĩa của từ và đề xuất một số kĩ thuật lượng giá.

Jaine Carbonell (1998) đã tóm tắt văn bản bằng cách xếp hạng các câu trội (câu chưa các ý chính của văn bản) và rút ra các câu trội.

Jade Goldstein (1999) : phân loại tóm tắt dựa trên độ đo liên quan, phương phpas sử dụng kết hợp giữa ngữ học, thống kê. Một câu được đặc trưng bằng các đặc tính ngữ học và độ đo thống kê.

J.Larocca Neto (2000) đã tạo tóm tắt văn bản dựa trên các dãy từ trong câu được chọn theo hệ số tf, sau đó dùng kỹ thuật gom cụm (clustering) để tạo tóm tắt.

Yoshio (2001) đã tạo tóm tắt văn bản tiếng Nhật. Có 2 phương pháp là rút câu dựa trên từ khoá và rút câu dựa trên kiến trúc ngữ nghĩa trong đó có xây dựng độ đo mối liên kiết giữa hai từ.

Hiện nay, một số nghiên cứu về xử lý ngôn ngữ tự nhiên cũng bước đầu được áp dụng trong tóm tắt văn bản. Mặt khác, các nghiên cứu về tóm tắt đa văn bản, đa ngôn ngữ và tóm tắt đa phương tiện cũng bắt đầu phát triển.

3. Phân loại các hệ thống tóm tắt văn bản

3.1. Phân loại theo kết quả (output)

Tóm tắt theo hướng trích rút (Extract): là một bản tóm tắt bao gồm các nội dung được rút trích từ văn bản gốc. Nói cách khác, văn bản tóm tắt được tạo ra bằng cách bỏ đi các từ, cụm từ hoặc câu không quan trọng và giữ lại các từ, cụm từ hoặc câu quan trọng trong văn bản gốc.

Tóm tắt theo hướng tóm lược (Abstract): là một bản tóm tắt có chứa cả các nội dung, từ ngữ không được thể hiện trong văn bản gốc. Hoặc có thể hiểu văn bản tóm tắt đã được biên tập lại bằng các từ ngữ, nội dung khác đi (có thể không nằm trong văn bản gốc) mà vẫn thể hiện được ý nghĩa quan trọng mà văn bản gốc thể hiện.

3.2. Phân loại theo số lượng tài liệu :

Tóm tắt đơn văn bản :

Bài toán tóm tắt văn bản đơn cũng giống như các bài toán tóm tắt khác, là một quá trình tóm tắt tự động với đầu vào là một văn bản, đầu ra là một đoạn mô tả ngắn gọn nội dung chính của văn bản đầu vào đó. Văn bản đơn có thể là một trang Web, một bài báo, hoặc một tài liệu với định dạng xác định (ví dụ : .doc, .txt)… Tóm tắt văn bản đơn là bước đệm cho việc xử lý tóm tắt đa văn bản và các bài toán tóm tắt

phức tạp hơn. Chính vì thế những phương pháp tóm tắt văn bản ra đời đầu tiên đều là các phương pháp tóm tắt cho văn bản đơn.

Các phương pháp nhằm giải quyết bài toán tóm tắt văn bản đơn cũng tập trung vào hai loại tóm tắt là: tóm tắt theo trích xuất và tóm tắt theo tóm lược.

Tóm tắt đa văn bản :

Tóm tắt đa văn bản có thể được coi như là một mở rộng của tóm tắt đơn văn bản.

Mục đích của tóm tắt đa văn bản: Là quá trình trích xuất nội dung từ một tập các văn bản có liên quan đến nhau, trong quá trình đó các thông tin dư thừa sẽ được loại bỏ và những thông tin quan trọng sẽ được biểu diễn dưới hình thức cô đọng, súc tích và giàu cảm súc đến người sử dụng hoặc chương trình cần dùng [MM99].

Tóm tắt đa văn bản được xác định là một bài toán có độ phức tạp cao, ngoài những thách thức đã được biết đến đối với tóm tắt đơn văn bản như sự cô đọng của thông tin và mạch lạc về nội dung, tóm tắt đa văn bản còn có những thách thức như cần phải xác định những thông tin trùng lặp giữa các văn bản, xác định thông tin quan trọng trong nhiều văn bản hay việc sắp xếp các thông tin trong văn bản tóm tắt.

3.3. Phân loại theo mục đích, chức năng tóm tắt (Function)

Tóm tắt chỉ thị (Indicative): tóm tắt nhằm cung cấp một chức năng tham khảo để chọn tài liệu dựa vào nội dung quan trọng. Ví dụ: Trong tóm tắt tin tức, tóm tắt đưa ra chi tiết chính của từng sự kiện.

Tóm tắt thông tin (Information): tóm tắt bao gồm tất cả các thông tin nổi bật có trong văn bản nguồn tại nhiều mức độ chi tiết khác nhau, tùy theo tỷ lệ nén được chỉ thị.

Tóm tắt đánh giá (Evaluation): tóm tắt nhằm mục đích đánh giá vấn đề chính của văn bản nguồn. Tóm tắt dạng này tập chung lấy ra các quan điểm, ý kiến chủ quan của tác giả nói đến trong văn bản.

3.4. Phân loại theo nội dung

Tóm tắt chung (Generalized): tóm tắt nhằm mục đích đưa ra các nội dung quan trọng bao quát nhất từ văn bản gốc.

Tóm tắt hướng truy vấn (Query-based): tóm tắt nhằm mục đích đưa ra kết quả dựa vào câu truy vấn của người dùng. Tóm tắt này thường được sử dụng trong quá trình tìm kiếm thông tin (information retreival). Đầu vào của tóm tắt dạng này không chỉ là văn bản gốc mà còn thêm vào truy vấn thể hiện thông tin mà người dùng quan tâm.

3.5. Phân loại theo miền dữ liệu

Tóm tắt trên một miền dữ liệu (Domain): tóm tắt nhắm vào một miền nội dung nào đó, như tin tức khủng bố, tin tức tài chính, tin khoa học công nghệ…

Tóm tắt trên một thể loại (Genre): tóm tắt nhắm vào một thể loại văn bản nào đó, như báo chí, email, web, bài báo…

Tóm tắt độc lập (Independent): tóm tắt thực hiện trên nhiều thể loại văn bản và nhiều miền dữ liệu khác nhau.

3.6. Phân loại theo mức độ chi tiết

Tóm tắt tổng quan (overview): tóm tắt miêu tả tổng quan tất cả các nội dung nổi bật trong văn bản nguồn.

Tóm tắt tập trung sự kiện (event): tóm tắt miêu tả một sự kiện cụ thể nào đó trong văn bản nguồn. Sự kiện được quan tâm đến được coi như một đầu vào trong quá trình xử lý tóm tắt tự động. Mục tiêu là chỉ đưa ra những nội dung có liên quan đến sự kiện đang được quan tâm mà thôi.

4. Ứng dụng của bài toán tóm tắt văn bản

Bài toán tóm tắt văn bản có thể ứng dụng vào rất nhiều hệ thống xử lý ngôn ngữ tự động khác nhau. Có thể kể tới một vài ứng dụng tiêu biểu sau đây:

- ✓ Tóm tắt tin tức.
- ✓ Tóm tắt kết quả tìm kiếm trong các máy tìm kiếm (search engine).
- ✓ Thu thập dữ liệu thông minh.
- ✓ Tóm tắt các văn bản, bài báo khoa học.
- ✓ Tóm tắt nội dung hội nghị, cuộc họp.
- ✓ Ứng dụng trong hệ thống trả lời tự động.

5. Lý do chọn đề tài

Với xu thế phát triển bùng nổ của internet hiện nay kéo theo một lượng thông tin khổng lồ về tất cả các lĩnh vực trong xã hội sinh ra trong mỗi giờ mỗi phút. Điều đó, một mặt tạo điều kiện cho con người có thể tiếp cận một cách nhanh chóng hơn với thông tin nhưng mặt khác lại làm cho con người chìm ngập trong biển thông tin khổng lồ khiến họ không thể xác định được thông tin nào là cần thiết, thông tin nào là vô ích. Nhu cầu cần có một phương án giải quyết vấn đề đó được đặt ra cấp thiết đối với mỗi con người. Bởi vậy, tóm tắt văn bản đã ra đời nhằm phục vụ nhu cầu đó của con người, giúp cho con người có thể tiếp cận thông tin một cách nhanh chóng và chính xác nhất.

Tóm tắt văn bản (TTVB) đã xuất hiện từ rất lâu, nhưng thường được thực hiện một cách truyền thống do con người. Những nghiên cứu về TTVB bắt đầu từ những năm 60 tại các phòng thí nghiệm nghiên cứu của Mỹ. Từ đó có nhiều phương pháp đã được đề xuất, nhiều hệ thống đã được xây dựng. Các phương pháp này thường dựa trên những kỹ thuật cơ bản được đề xuất bởi Luhn, Sdmundson và Salton là trích rút các câu quan trọng từ trong văn bản gốc và kết hợp lại thành văn bản tóm tắt. Với sự phát triển của internet, chủ đề về TTVB đã thu hút sự quan tâm của nhiều nhà nghiên cứu trong lĩnh vực xử lý ngôn ngữ tự nhiên và tra cứu thông tin (WAS 2000, 2001, 2002), nhiều các chủ đề đặc biệt trong các phiên của các hội thảo ACL, COLING, SINGIR đã được tổ chức.

Hiện nay, tóm tắt văn bản là một lĩnh vực quan trọng trong xử lý văn bản thu hút nhiều nhà nghiên cứu quan tâm. Ứng dụng của TTVB trong nhiều lĩnh vực khác nhau như sinh tiêu đề tự động (headline generation), rút gọn thông tin sử dụng trong các thiết bị cầm tay như PDA, điện thoại di động,…Trong TTVB có lĩnh vực nhỏ hơn được coi là mở đầu của TTVB, nó là tiền đề cho các hình thức tóm tắt phực tạp hiện nay, đó chính là tóm tắt đơn văn bản. Hiện nay đã có nhiều phương pháp tóm tắt văn bản được đề xuất, tuy nhiên, mỗi phương pháp đều có điểm mạnh và điểm yếu riêng. Trong phạm vi bài toàn tóm tắt đơn văn bản theo hướng trích rút các câu quan trọng, hay nói cách khác là phân loại câu quan trọng và không quan trọng trong văn bản để đưa các câu quan trọng vào văn bản tóm tắt, phương pháp học máy có giám sát SVM[2] là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại với số chiều có thể cực lớn. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn. Việc ứng dụng phương pháp SVM[2] rất phù hợp cho bài toán tóm tắt văn bản theo hướng trích rút.

Chính vì vậy, em chọn để tài ***"nghiên cứu phương pháp tóm tắt đơn văn bản tự động sử dụng mô hình svm ước lượng xác suất"*** làm đồ án tốt nghiệp của mình, nhằm mục đích muốn tìm hiểu về một phương pháp mới cho bài toán tóm tắt đơn văn bản, đồng thời cũng muốn so sánh hiệu quả của phương pháp xây dựng với các phương pháp đã có. Để có thể mở ra một hướng đi mới hiệu quả hơn cho bài toán tóm tắt đơn văn bản nói riêng và bài toán tóm tắt văn bản nói chung.

CHƯƠNG II: TỔNG QUAN VỀ BÀI TOÁN TÓM TẮT VĂN BẢN

1. Bài toán tóm tắt văn bản

TTVB là quá trình thực hiện giảm đi độ dài, sự phức tạp của một văn bản trong khi vẫn giữ lại được các nội dung có giá trị của nó. TTVB nhằm đưa ra thể hiện về nội dung một cách ngắn gọn của văn bản.

Có thể phát biểu bài toán TTVB trong phạm vi nghiên cứu của đồ án như sau:

Đầu vào :	Một văn bản có độ dài trên 5000 từ
Đầu ra :	Nội dung ngắn gọn khoảng 250 từ

Bảng 1: Bài toán TTVB

2. Các phương pháp giải quyết bài toán tóm tắt đơn văn bản

Hiện nay, có rất nhiều phương pháp nhằm giải quyết bài toán tóm tắt đơn văn bản nhưng đa phần đều tập trung vào hai loại tóm tắt là: tóm tắt trích rút và tóm tắt tóm lược.

2.1. Tóm tắt theo trích rút

Đa số các phương pháp tóm tắt theo loại này đều tập trung vào việc trích rút ra các câu hay các ngữ nổi bật từ các đoạn văn bản và kết hợp chúng lại thành một văn bản tóm tắt.

Một số nghiên cứu giai đoạn đầu thường sử dụng các đặc trưng như vị trí của câu trong văn bản, tần suất xuất hiện của từ, ngữ hay sử dụng các cụm từ khóa để tính toán trọng số của mỗi câu, qua đó chọn ra các câu có trọng số cao nhất cho văn bản tóm tắt.

Các kỹ thuật tóm tắt gần đây sử dụng các phương pháp học máy và xử lý ngôn ngữ tự nhiên nhằm phân tích để tìm ra các thành phần quan trọng của văn bản. Sử dụng các phương pháp học máy có thể kể đến các phương pháp của Kupiec, Penderson and Chan năm 1995 sử dụng phân lớp Bayes để kết hợp các đặc trưng lại với nhau hay nghiên cứu của Lin và Hovy năm 1997 áp dụng phương pháp học máy nhằm xác định vị trí của các câu quan trọng trong văn bản.

Bên cạnh đó việc áp dụng các phương pháp phân tích ngôn ngữ tự nhiên như sử dụng mạng từ Wordnet của Bazilay và Elhadad vào năm 1997

2.2. Tóm tắt theo tóm lược

Các phương pháp tóm tắt không sử dụng trích rút để tạo ra tóm tắt có thể xem như là một phương pháp tiếp cận tóm tắt theo tóm lược. Các hướng tiếp cận có thể

kể đến như dựa vào trích rút thông tin (information extraction), ontology, hợp nhất và nén thông tin,… Một trong những phương pháp tóm tắt theo tóm lược cho kết quả tốt là các phương pháp dựa vào trích rút thông tin, phương pháp dạng này sử dụng các mẫu đã được định nghĩa trước về một sự kiện hay là cốt truyện và hệ thống sẽ tự động điền các thông tin trong mẫu có sẵn rồi sinh ra kết quả tóm tắt.

Tuy nhiên hiện nay các kỹ thuật sinh ra văn bản từ văn bản gốc còn hạn chế, mới chỉ dừng ở mức nghiên cứu, kết quả chưa được cao.

3. Các hướng tiếp cận đối với tóm tắt đơn văn bản

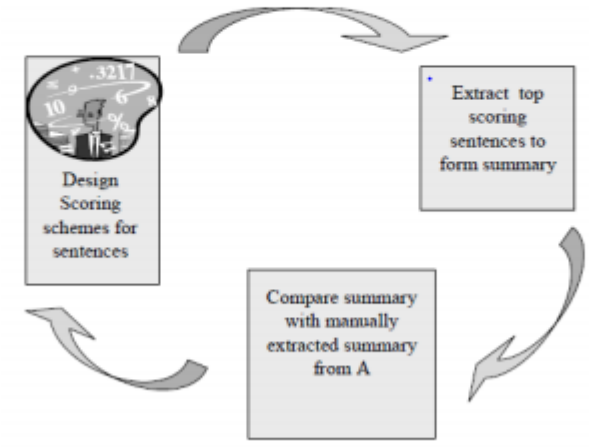
Mặc dù có 2 loại tóm tắt là tóm tắt trích rút và tóm tắt tóm lược, tuy nhiên để thực hiện tóm tắt tóm lược cần có một lượng tri thức đầy đủ về lĩnh vực cần tóm tắt. Điều này hiện nay còn hạn chế nhiều, do đó các hướng tiếp cận đa số tập trung vào dạng tóm tắt trích rút câu.

Sau đây là một số hướng tiếp cận cho bài toán tóm tắt đơn văn bản:

3.1. Phương pháp thống kê

Hầu hết các nghiên cứu đầu tiên cho tóm tắt đơn văn bản đều tập trung trên những văn bản kỹ thuật (các bài báo khoa học). Các phương pháp cổ điển thường tập trung vào các đặc trưng hình thái để tính điểm cho các câu và trích rút các câu quan trọng để đưa vào tóm tắt.

Bước 1



Bước 2

Hình 2: Hướng tiếp cận chung cho phương pháp cổ điển

Ý tưởng của hướng tiếp cận này:

- Thu thập dữ liệu
- Tạo các bản tóm tắt thủ công
- Thiết kế các công thức toán hay logic để tính điểm cho các câu.
- Lặp cho đến khi tóm tắt tự động đạt được tính tương đương với tóm tắt thủ công:
    - Tính điểm cho từng câu để tạo ra bản tóm tắt cho từng văn bản trong ngữ liệu dựa vào các đặc trưng về hình thái.
    - So sánh tóm tắt được tạo tự động với tóm tắt được tạo thủ công.
    - Cải thiện lại phương thức tính điểm cho câu.

Các nghiên cứu đại diện cho phương pháp này:

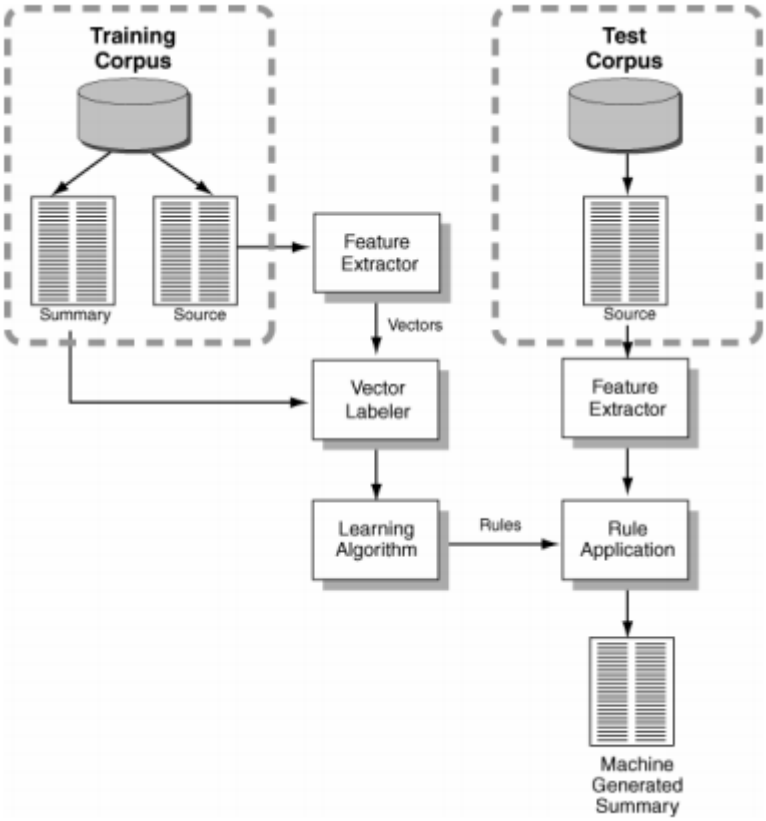
- **Luhn(1958)**
    - Sử dụng các đặc trưng như: word frequency, stop words, word distance.
    - Dùng phương pháp so khớp từng kí tự để giải quyết stemming.
- **Baxendale(1958)**
    - Sử dụng các đặc trưng như: sentence position.
    - Thử nghiệm 200 đoạn câu, 85% các câu đầu là câu chính và 7% các câu cuối và câu chính.
    - Phương pháp khá chính xác nhưng quá chủ quan và ngây ngô. Phương pháp này được sử dụng khá nhiều vào các hệ thống học máy sau này.
- **Edmundson(1969)**
    - Điển hình nhất trong phương pháp cổ điển.
    - Sử dụng các đặc trưng như: word frequency, stop words, position, cue words, title.
    - Sử dụng phương pháp kết nối tuyến tính để kết hợp các điểm đặc trưng lại với nhau:

      $Si = w1*Ci + w2*Ki + w3*Ti + w4*Li$
    - Thử nghiệm với 400 văn bản kỹ thuật và kết quả đạt 44%.

3.2. Phương pháp máy học (machine learning)

Năm 1990, với sự phát triển của nhiều kỹ thuật học máy trong xử lý ngôn ngữ, một số nhà nghiên cứu đã ứng dụng các kỹ thuật này vào trong TTVB tự động. Một số nghiên cứu điển hình của phương pháp này là: Navie – Bayes, Decision Tree, Hidden Makov Model, Log – Linear, Neural Network, SVM.

Framework chung cho hệ thống tóm tắt văn bản bằng phương pháp máy học:



Hình 3: Mô hình chung cho hệ thống TTVB bằng phương pháp học máy

3.2.1. Phương pháp Naïve-Bayes

Các hướng tiếp cận theo phương pháp này giả định rằng các đặc trưng của văn bản độc lập nhau. Sử dụng bộ phân lớp Navie – Bayes để xác định câu nào thuộc về tóm tắt và ngược lại:

Cho s là các câu cần xác định. F1…..Fk là các đặc trưng đã được chọn và giả định các thuộc tính độc lập với nhau. Xác suất của câu s thuộc về tóm tắt được tính như sau:

$$P(s \in \mathcal{S} \mid F_1, F_2, ..F_k) = \frac{\prod_{i=1}^{k} P(F_i \mid s \in \mathcal{S}) \cdot P(s \in \mathcal{S})}{\prod_{i=1}^{k} P(F_i)}$$

Sau khi tính xác suất các câu, n câu có xác suất cao nhất sẽ được trích rút.

Các nghiên cứu đại diện cho phương pháp này:

- **Kupiec(1995)**
  - Các đặc trưng sử dụng: word frequency, location, cue word, title & leading, sentence length, uppercase words.
  - Ngữ liệu: 188 cặp văn bản khoa học và tóm tắt. Tổng số câu: 568 câu.Số câu khớp trực tiếp với tóm tắt 451 (79%).
- **Aone(1999)**

- Kết hợp thêm nhiều đặc trưng phong phú hơn: tf.idf(single word, two-noun word, named-entities), discourse (cohension) (sử dụng Wordnet và kỹ thuật xử lý ngôn ngữ tự nhiên để phân tích sự tham chiếu đối với các thực thể).
- Ngữ liệu: sử dụng ngữ liệu của TREC.
- Hệ thống: DimSum.

Phương pháp OPP (Optimal Position Policy)

Lin và Hovy (1997) đã nghiên cứu tính quan trọng của đặc trưng vị trí câu(sentence position) và cho rằng các câu trong văn bản tuân theo một cấu trúc diễn ngôn (diễn giải) có thể dự đoán được. Và do cấu trúc trong các loại văn bản khác nhau, nên đặc trưng về vị trí câu không thể định nghĩa đơn giản như trong phương pháp Navie – Bayes.

Lin và Hovy đã đề ra phương pháp Optimal Position Policy cho một thể loại văn bản (văn bản tin tức của Zif-Davis về máy tính và phần cứng). Phương pháp thực hiện:

- Với mỗi văn bản, tính năng suất của mỗi vị trí câu với các từ khóa chủ đề.
- Xếp hạng các vị trí câu với năng suất trung bình bằng thủ tục OPP.
- Lấy ra n vị trí câu trong bảng xếp hạng làm tóm tắt.

3.2.2. Phương pháp SVM

Phương pháp SVM được coi là công cụ mạnh cho những bài toán phân lớp phi tuyến tính được các tác giả Vapnik và Chervonenkis phát triển mạnh mẽ năm 1995.

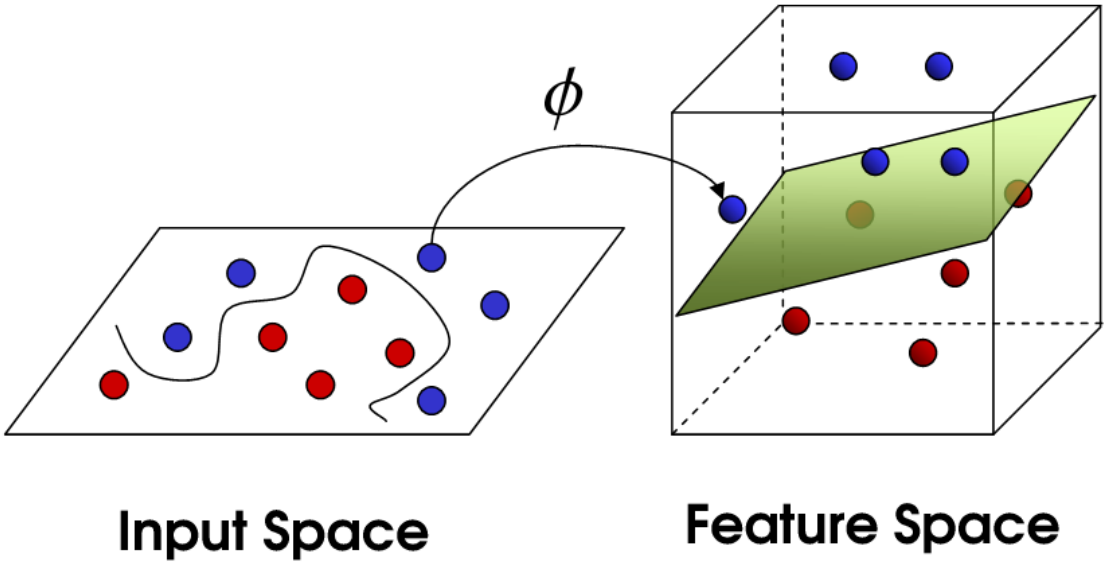
SVM là phương pháp học có giám sát được sử dụng rộng rãi trong lĩnh vực phân lớp mẫu và nhận dạng mẫu. SVM là một họ các phương pháp dựa trên cơ sở các hàm nhân (kernel methods) để tối thiểu hoá rủi ro ước lượng. Phương pháp này được Boser, Guyon, và Vapnik giới thiệu lần đầu tiên vào năm 1995 để giải quyết vấn đề phân lớp mẫu hai lớp sử dụng nguyên tắc **cực tiểu hóa rủi ro cấu trúc (Structural Risk Minimization)**. Phương pháp tiếp cận này dựa trên lý thuyết toán học thống kê nên có một nền tảng toán học chặt chẽ để đảm bảo rằng kết quả đạt được là tối ưu. SVM là một trong những phương pháp máy học trong đó các khái niệm dựa trên dữ liệu đã thu thập được trước đó. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.

Là thuật toán học giám sát (supervied learning) được sử dụng cho phân lớp dữ liệu.

Là 1 phương pháp thử nghiệm, đưa ra 1 trong những phương pháp mạnh và chính xác nhất trong số các thuật toán nổi tiếng về phân lớp dữ liệu

SVM là một phương pháp có tính tổng quát cao nên có thể được áp dụng cho nhiều loại bài toán nhận dạng và phân loại

Cơ sở lý thuyết:



Hình 4: Mô phỏng phân loại SVM

Support vector machine (SVM) xây dựng (learn) một siêu phẳng (hyperplane) để phân lớp (classify) tập dữ liệu thành 2 lớp hay nhiều lớp riêng biệt.

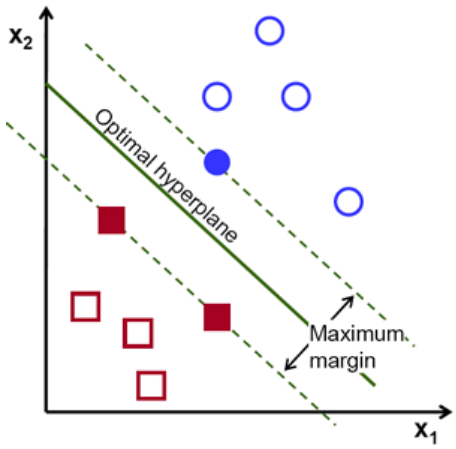
Một siêu phẳng là một hàm tương tự như phương trình đường thẳng, y = ax + b. Trong thực tế, nếu ta cần phân lớp tập dữ liệu chỉ gồm 2 feature, siêu phẳng lúc này chính là một đường thẳng.

Về ý tưởng thì SVM sử dụng thủ thuật để ánh xạ tập dữ liệu ban đầu vào không gian nhiều chiều hơn. Khi đã ánh xạ sang không gian nhiều chiều, SVM sẽ xem xét và chọn ra siêu phẳng phù hợp nhất để phân lớp tập dữ liệu đó.

Bằng cách sử dụng một kernel, SVM ánh xạ tập dữ liệu ban đầu vào không gian nhiều chiều

a. Thuật ngữ margin trong SVM:

Margin là khoảng cách giữa siêu phẳng đến 2 điểm dữ liệu gần nhất tương ứng với các phân lớp.



Hình 5: Margin trong SVM

SVM cố gắng maximize margin này, từ đó thu được một siêu phẳng tạo khoảng cách xa nhất so với phần tử của 2 lớp. Nhờ vậy, SVM có thể giảm thiểu việc phân lớp sai (misclassification) đối với điểm dữ liệu mới đưa vào.

b. <u>Bài toán tối ưu của SVM</u>

- Dữ liệu huấn luyện của SVM là tập các điểm dữ liệu
- D = {(x_1,y_1), (x_2,y_2), …, (x_1,y_1)}, trong đó, x_i là vector dữ liệu biểu diễn đối tượng cần phân lớp d_i ($x_i \in R^n$), $y_i \in$ {+1,-1}, cặp (x_i, y_i) được hiểu là vector x_i được gán nhãn là y_i.
- Một siêu phẳng phân chia dữ liệu được gọi là "tốt nhất", nếu khoảng cách từ điểm dữ liệu gần nhất đến siêu phẳng là lớn nhất. Phương trình tổng quát của một siêu phẳng phân chia như vậy được biểu diễn có dạng như sau:

$$\mathbf{w^T x + b = 0} \qquad \mathbf{(1)}$$

Trong đó:

W^T: Vector trọng số, $W^T = \{w_1, w_2,\ldots,w_n\}$.

n: Số thuộc tính (hay còn gọi là số chiều của dữ liệu).

b: Bộ trọng số.

Với T = 2 (dữ liệu hai chiều) → siêu phẳng phân chia là đường thẳng.

Với T = 3 (dữ liệu ba chiều) → siêu phẳng phân chia là mặt phẳng.

- Tổng quát cho dữ liệu n chiều thì sẽ được phân cách bởi một siêu phẳng.
- Siêu phẳng phân chia có vai trò quan trọng trong việc phân lớp, nó quyết định xem một bộ dữ liệu sẽ thuộc về lớp nào. Ta xét trên ví dụ sau:
- Với bộ dữ liệu huấn luyện hai chiều, ta có 2 thuộc tính A_1 và A_2: X={x_1, x_2}, với x_1, x_2 là giá trị của thuộc tính A_1, A_2 và W = {w_1, w_2}. Phương trình siêu phẳng có thể viết lại như sau:

$$H: w_0 + w_1x_1 + w_2x_2 = 0$$

Trong đó: w_0 tương đương với hằng số *b* trong PT tổng quát của siêu phẳng.

Vì vậy mỗi điểm nằm trên siêu phẳng phân cách thỏa mãn:

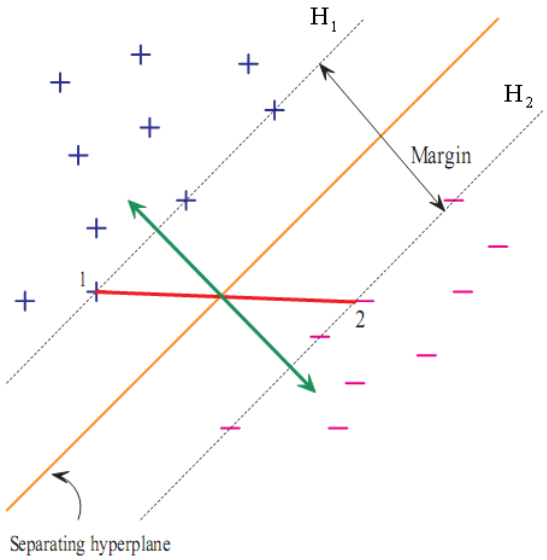
$$H_1: w_0 + w_1x_1 + w_2x_2 > 0$$

Tương tự, những điểm nằm dưới siêu phẳng phân cách phải thỏa mãn:

$$H_2: w_0 + w_1x_1 + w_2x_2 < 0$$

Bằng cách điều chỉnh trọng số w_0 ta có:

$$H_1: w_0 + w_1x_1 + w_2x_2 \geq 1 \; với y_i = +1$$
$$H_2: w_0 + w_1x_1 + w_2x_2 \leq -1 \; với y_i = -1$$



Hình 6: Đường biểu diễn H1 và H2

Trong đó:

Đường màu đỏ là khoảng cách Euclidean của hai điểm 1 và 2.

Đường màu xanh là khoảng cách Euclidean nhỏ nhất.

- Điều này có nghĩa là nếu bất kì bộ nào nằm tại hoặc trên H_1 đều thuộc về lớp +1, và bất kì bộ nào nằm tại hoặc dưới H_2 đều thuộc về lớp -1. Kết hợp 2 bất đẳng thức trên ta có:

$$y_i(w_0 + w_1x_1 + w_2x_2) \geq 1, \forall i \qquad (5)$$

- Mỗi bộ huấn luyện nằm tại các mặt biên H_1 hay H_2thỏa mãn phương trình trên được gọi là support vectors. Support vectors là những bộ gần với siêu phẳng phân chia tuyến tính nhất.

- Tuy nhiên trong thực tế có thể tìm được vô số những siêu phẳng phân chia trên cùng một tập dữ liệu. Do đó mục tiêu của phương pháp phân lớp SVM là tìm một siêu phẳng phân cách giữa hai lớp sao cho khoảng cách lề giữa hai lớp đạt cực đại, nghĩa là có sai sót phân loại bé nhất trên bộ dữ liệu.

- Siêu phẳng có biên độ lớn nhất sẽ được chọn như là siêu phẳng phân chia tập dữ liệu một cách tốt nhất. Tức là, nếu có 2 siêu phẳng có thể phân chia được tất cả những bộ dữ liệu cho trước với biên độ của nó. Siêu phẳng với biên độ lớn hơn sẽ chính xác hơn trong việc phân loại các bộ dữ liệu trong tương lai so với siêu phẳng có biên độ nhỏ hơn. Điều này là lý do tại sao (trong suốt giai đoạn học hay huấn luyện), SVM tìm những siêu phẳng có biên độ lớn nhất, gọi là MMH (maximum marginal hyperlane).Siêu phẳng có biên độ lớn nhất là siêu phẳng có khoảng cách từ nó tới hai mặt bên của nó thì bằng nhau (mặt bên song song với siêu phẳng). Khoảng cách đó là khoảng cách ngắn nhất từ MMH tới bộ dữ liệu huấn luyện gần nhất của mỗi lớp. Siêu phẳng có biên độ lớn nhất này cho chúng ta một sự phân loại tốt nhất giữa các lớp.

- Việc huấn luyện SVM với mục đích trên có thể được sử dụng để phân lớp dữ liệu mà dữ liệu đó có thể phân chia tuyến tính. Chúng ta xem SVM được huấn luyện là SVM tuyến tính.

- Ngoài ra hướng tiếp cận của SVM tuyến tính có thể được mở rộng để tạo ra SVM không tuyến tính cho việc phân lớp các dữ liệu không thể phân chia tuyến tính (hay gọi tắt là dữ liệu không tuyến tính). Những SVM như vậy có khả năng tìm những ranh giới quyết định không tuyến tính (những mặt không tuyến tính) trong không gian đầu vào. Những SVM như vậy được gọi là SVM phi tuyến.

- Để tìm ra các support vectors và MMH, đồng nghĩa với việc tìm được bộ phân lớp trên bộ dữ liệu đã cho. Có ba trường hợp có thể xảy ra đối với từng bộ dữ liệu, mỗi trường hợp sẽ đưa ra một bài toán tối ưu. Việc cần làm là giải quyết bài toán tối ưu đó.

3.2.3. Phương pháp Decision Tree

Lin và Hovy (1999) đại diện của phương pháp này giả định rằng, các đặc trưng không độc lập nhau. Tác giả đã kiểm tra nhiều đặc trưng và ảnh hưởng của chúng lên quá trình trích rút. Hệ thống tóm tắt của Lin là loại tóm tắt hướng về truy vấn (Query - based).

Các đặc trưng: position (OPP), numeric data, proper name, pronoun & adjective, weekday hoặc month. Cùng với 2 đặc trưng mới: query signature (số từ truy vấn có trong câu) và IR signature (những từ nổi bật, quan trọng ~ tf*idf).

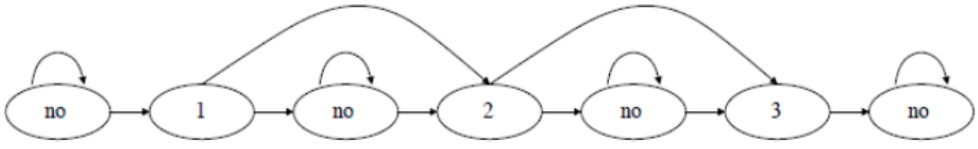
Hệ thống Summarist của Lin và Hovy sử dụng thuật toán C4.5 để huấn luyện cây quyết định. Hệ thống sử dụng tập ngữ liệu của TIPSTER-SUMMAC

3.2.4. Phương pháp Hidden Makov Model

Những hướng tiếp cận trước đều không dựa trên những đặc trưng và không tuần tự. Conroy và O'leary (2001) đã đưa ra hướng tiếp cận dựa trên mô hình HMM với ý tưởng cơ bản là sử dụng một chuỗi tuần tự các câu. Tác giả đưa ra khái niệm về sự phụ thuộc cục bộ (local dependencies) giữa các câu và sử dụng mô hình HMM để xác định sự phụ thuộc này.

Các đặc trưng sử dụng: position, number of term, likelihood of sentence.

Mô hình HMM bao gồm 2s + 1 trạng thái, trong đó s là số trạng thái tóm tắt (câu thuộc tóm tắt) và s + 1 là câu không thuộc tóm tắt.



Hình 7: Ví dụ về mô hình Hidden Makov Model

Mô hình HMM xây dựng ma trận chuyển vị trí M, coi các đặc trưng là đa biến và tính xác suất của các câu qua từng trạng thái.

Sử dụng tập ngữ liệu của TREC và được đánh giá với 2 hệ thống khác là DimSum và QR, kết quả đều cho độ đo Precision cao hơn.

3.2.5. Phương pháp Log-Linear

Osborne (2002) đại diện cho mô hình này cũng coi các đặc trưng là không độc lập với nhau và sử dụng mô hình Log-Linear khắc phục giả định này.

Các đặc trưng sử dụng: word pair, sentence length, sentence position và discourse features (nằm trong introduction hay conclusion).

Mô hình huấn luyện của Log-Linear được thực hiện như sau:

$$P(c \mid s) = \frac{1}{Z(s)} \exp\left(\sum_i \lambda_i f_i(c, s)\right)$$

Trong đó, c là nhãn muốn gán cho câu s, f_i là đặc trưng thứ i và λ_i là trọng số kết nối các đặc trưng. Nhãn c có 2 khả năng: thuộc tóm tắt hoặc không thuộc tóm tắt.

Giai đoạn phân lớp câu mới được thực hiện như sau:

$$\text{label}(s) = \arg\max_{c \in C} P(c) \cdot P(s, c) = \arg\max_{c \in C} \left(\log P(c) + \sum_i \lambda_i f_i(c, s)\right)$$

Kết quả được đo bằng độ đo f2 = 2pr/(p+r). Tác giả đã đánh giá với hướng tiếp cận Bayes và kết quả luôn cho độ đo f2 cao hơn.

3.3. Phương pháp phân tích ngôn ngữ tự nhiên

Phương pháp tiếp theo sử dụng các kỹ thuật phân tích ngôn ngữ tự nhiên phức tạp. Không phải tất cả các phương pháp phân tích ngôn ngữ tự nhiên đều sử dụng học máy, đôi khi phương pháp chỉ sử dụng một số các heuristic để tạo trích rút.

Hầu hết các phương pháp này đều dựa trên cấu trúc diễn ngôn (discourse structure) hay cấu trúc diễn đạt (thể hiện) của văn bản, như: cấu trúc các section của văn bản, liên kết ngữ pháp (trùng lặp, tỉnh lược, liên hợp), liên kết từ vựng (đồng nghĩa, bao hàm, lặp lại), cấu trúc chính phụ.

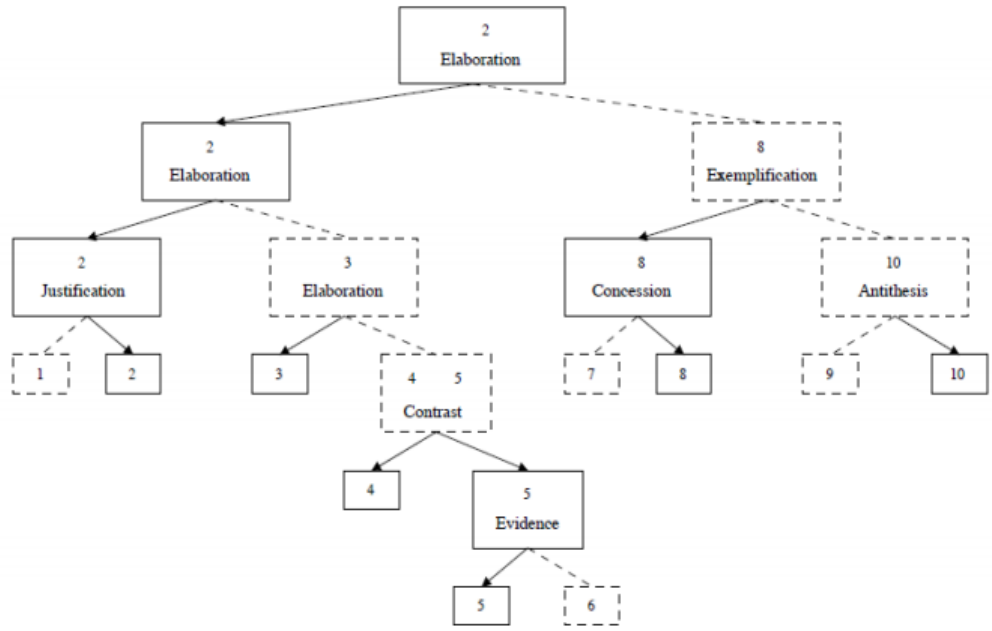
Các nghiên cứu đại diện cho phương pháp này:

- **Rhetorical Structure(Cấu trúc văn bản, ngữ pháp): Ono (1994)**
  - Xây dựng một thủ tục để trích rút các cấu trúc chính phụ (rhetorical structure) từ các văn bản tiếng Nhật và xây dựng một cây nhị phân để thể hiện.

- Các bước để trích rút cấu trúc: phân tích câu, trích rút một quan hệ chính phụ, phân đoạn, tạo ứng viên và đánh giá độ ưu tiên.
- Sau khi xây dựng cây sẽ thực hiện tỉa nhánh để giảm bớt câu và tạo tóm tắt.
- Kết quả đạt được 51% các câu chính được xác định và 74% các câu quan trọng nhất được xác định.

- **Lexical Chain(Cấu trúc, ngữ nghĩa chuỗi từ vựng): Barzilay và Elhadad(1997)**
  - Hai tác giả cũng đã sử dụng một lượng đáng kể những phân tích ngôn ngữ trong TTVB dựa trên chuỗi từ vựng (lexical chain). Chuỗi từ vựng là chuỗi các từ liên quan trong văn bản.
  - Các bước thực hiện: phân tích đoạn văn bản, xác định các chuỗi từ vựng và sử dụng các từ vựng tốt nhất để xác định câu được chèn vào tóm tắt.
  - Để tìm các chuỗi từ vựng tác giả sử dụng Wordnet. Các từ có liên quan với nhau sẽ được đưa vào chuỗi. Sự liên quan được tính bằng khoảng cách trong Wordnet. Chuỗi sẽ được tính điểm dựa vào chiều dài và sự đồng nhất của nó.
  - Kết quả đạt được tốt hơn hệ thống tóm tắt của Microssoft với độ Precision là 61 và recall 67 (Microsoft là 33 và 27).
  - Hạn chế: Không thể kiểm được chiều dài và mức độ chi tiết của tóm tắt do số chuỗi còn ít. Tóm tắt thiếu sự kết dính và chưa chi tiết so chọn cả câu.
- **Rhetorical Structure(Cấu trúc ngữ nghĩa, đoạn văn): Marcu (1998)**

Sử dụng các heuristic dựa trên cấu trúc diễn đạt với các đặc trưng truyền thống. Lý thuyết về cấu trúc diễn đạt được tác giả thể hiện thông qua lý thuyết cấu trúc chính phụ(Rhetorical Structure Theory). Lý thuyết cho rằng hai khoảng văn bản không trùng lặp có mối quan hệ trung tâm (nucleus) và vệ tinh (satellite). Trong đó, trung tâm quan trọng hơn vệ tinh và độc lập hoàn toàn trong cấu trúc chính phụ. Cấu trúc trọng tâm và vệ tinh được biểu diễn thành cây nhị phân.



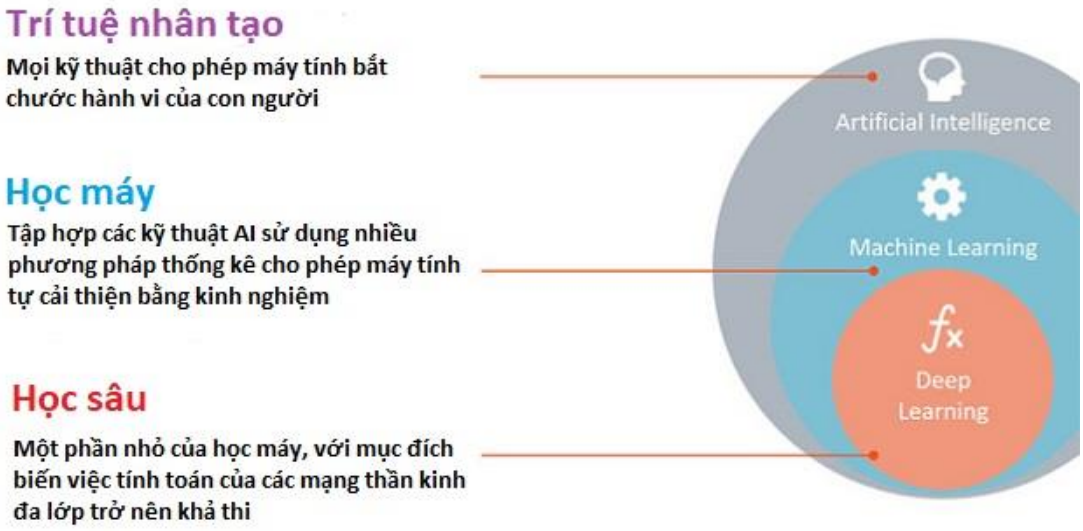
Hình 8: Văn bản biểu diễn dưới dạng cây nhị phân

Để tính điểm cho các cấu trúc, tác giả sử dụng nhiều độ đo khác nhau như: clustering-based metric, marker-based metric, rhetorical clustering-based technique, shape-based metric, title-based metric, position-based metric, connectedness-based metric và sử dụng phương pháp kết hợp tuyến tính. Lấy ra n câu chứa cấu trúc có điểm cao nhất.

Hệ thống đạt được hiệu quả độ đo F 75.42% cao hơn 3.5% so với baseline bằng phương pháp lấy n câu đầu. Ngữ liệu được sử dụng là từ TREC

3.4. Phương pháp học sâu (deep learning)

Trong những năm qua, thuật ngữ "deep learning" (học sâu) đã dần len lỏi mỗi khi có cuộc hội thoại nào bàn về trí tuệ nhân tạo (AI), dữ liệu lớn (Big Data) và phân tích (Analytics). Và với lý do chính đáng – đây là một cách tiếp cận đầy hứa hẹn tới AI khi phát triển các hệ thống tự trị, tự học, những thứ đang cách mạng hóa nhiều ngành công nghiệp.



Hình 9: Tổng quan về mô hình học sâu

Học sâu là cho một hệ thống máy tính "ăn" rất nhiều dữ liệu, để chúng có thể sử dụng và đưa ra các quyết định về những dữ liệu khác. Dữ liệu này được nạp thông qua các mạng thần kinh, tương tự như học máy. Những mạng lưới này – các cấu trúc logic yêu cầu một loạt các câu hỏi đúng/sai, hoặc trích xuất một giá trị số, của mỗi bit dữ liệu đi qua chúng và phân loại theo các câu trả lời nhận được.

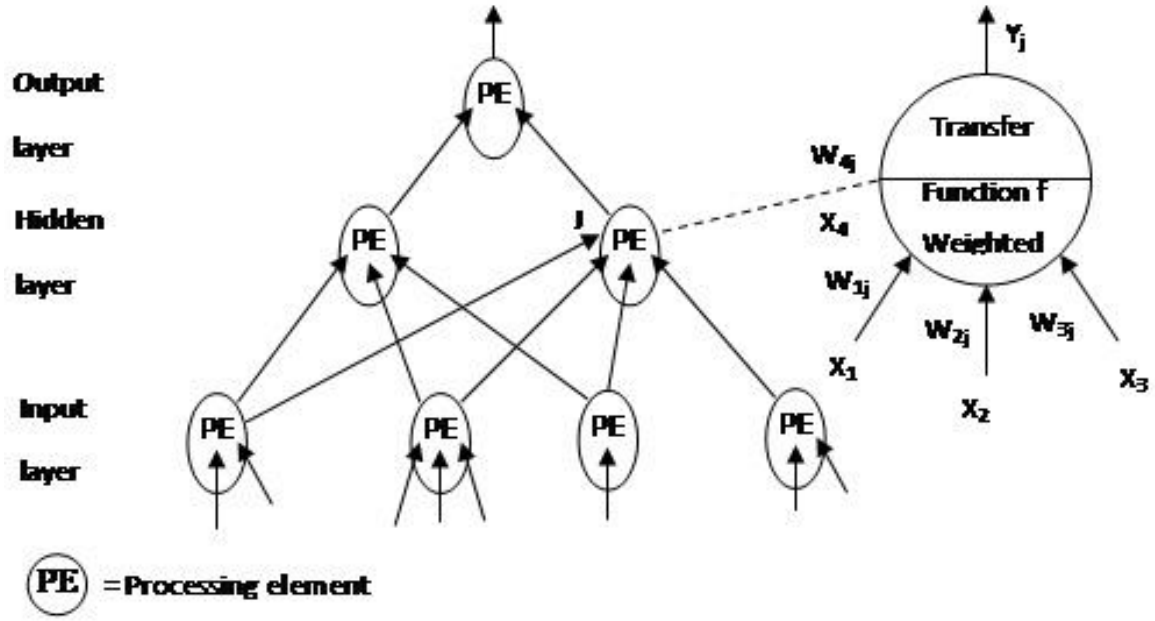
Vì công việc của học sâu là tập trung phát triển những mạng lưới này, chúng đã trở thành "mạng thần kinh sâu" (Deep Neural Network) – những mạng logic phức tạp cần thiết để xử lý các bộ dữ liệu lớn, như thư viện hình ảnh của Google hay Instagram.

Mạng neural nhân tạo (Artificial neural network)

Mạng Nơron nhân tạo (Artificial Neural Network- ANN) là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các Nơron được gắn kết để xử lý thông tin. ANN giống như bộ não con người, được học bởi kinh nghiệm (thông qua huấn luyện), có khả năng lưu giữ những kinh nghiệm hiểu biết (tri thức) và sử dụng những tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data).

Kiến trúc chung của một mạng nơron nhân tạo (ANN) gồm 3 thành phần đó là: Input Layer, Hidden Layer và Output Layer (Xem Hình 3.2).

Trong đó, lớp ẩn (Hidden Layer) gồm các Nơron nhận dữ liệu input từ các Nơron ở lớp (Layer) trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Trong một ANN có thể có nhiều lớp ẩn.



Hình 10: Tổng quan mạng neural

Trong đó các Processing Elements (PE) của ANN gọi là Nơron, mỗi Nơron nhận các dữ liệu vào (Inputs) xử lý chúng và cho ra một kết quả (Output) duy nhất. Kết quả xử lý của một Nơron có thể làm Input cho các Nơron khác.

Bên trên là tổng quan về kiến trúc mạng ANN cơ bản. Để phục vụ những bài toán phức tạp ta cũng cần những kiến trúc mạng phức tạp hơn. Một số kiến trúc mạng phổ biến hiện nay như:

- Deep Neural Network (DNN)
- Deep Belief Network (DBN)
- Deep Boltzmann Machine (DBM)
- **Recurrent Neural Network (RNN)**
- Convolution Neural Network (CNN)
- Multi-modal/multi-tasking
- Deep Stacking Network (DSN)

Trong các kiến trúc trên mô hình mạng neural hồi quy RNN là mô hình được áp dụng rất rộng rãi trong các bài toán xử lý ngôn ngữ tự nhiên (NLP). Do mô hình RNN mô hình hoá được bản chất dữ liệu trong NLP. Dữ liệu trong NLP có đặc tính chuỗi và có sự phụ thuộc lẫn nhau giữa các thành phần (trạng thái) trong dữ liệu.

Hạn chế của hướng tiếp cận học sâu với bài toán đặt ra trong đề tài:

Với hướng tiếp cận này, hiện nay chưa có kho dữ liệu phù hợp cho bài toán tóm tắt văn bản của đề tài. Trên thế giới, các kết quả được nghiên cứu trên kho dữ liệu với các văn bản có độ dài khoảng 100 từ và đưa ra một headline cho văn bản đó.

4. Đề xuất hướng giải quyết

Với bài toán tóm tắt văn bản theo hướng trích rút các câu quan trọng để đưa vào văn bản tóm tắt. Ý tưởng đưa ra : cần phân tích từng câu trong văn bản, và phân lớp chúng thành hai lớp câu quan trọng và câu không quan trọng.

Nhận thấy phương pháp học máy SVM là 1 trong những phương pháp mạnh và chính xác nhất trong số các thuật toán nổi tiếng về phân lớp dữ liệu hiện nay.,được coi là công cụ mạnh cho những bài toán phân lớp phi tuyến tính, SVM là phương pháp học có giám sát được sử dụng rộng rãi trong lĩnh vực phân lớp mẫu và nhận dạng mẫu. Phương pháp tiếp cận này dựa trên lý thuyết toán học thống kê nên có một nền tảng toán học chặt chẽ để đảm bảo rằng kết quả đạt được là tối ưu. Phương pháp này cho phép tận dụng được nguồn dữ liệu rất nhiều và sẵn có.

Bên cạnh đó SVM[2] là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại với số chiều có thể cực lớn. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn. Việc ứng dụng phương pháp SVM[2] rất phù hợp cho bài toán tóm tắt văn bản theo hướng trích rút.

Do đó em giải quyết bài toán phân lớp câu quan trọng và không quan trọng bằng hướng tiếp cận sử dụng SVM.

Các câu trong các văn bản của bộ dữ liệu huấn luyện sẽ được xử lý, phân tích và mô hình hóa cho phù hợp với đầu vào của mô hình huấn luyện. Từ đó cho SVM học các dữ liệu đã được phân lớp với các tùy chọn phù hợp để sinh ra model huấn luyện, phục vụ cho đánh dấu câu quan trọng trong các văn bản kiểm thử.

5. Phương pháp đánh giá

Recall-Oriented Understudy for Gisting Evaluation (ROUGE)[5] là một phương pháp do Lin và Hovy đưa ra vào năm 2003 cũng dựa trên các khái niệm tương tự. Phương pháp này đã cho ra kết quả khả quan và được sự đánh giá cao của cộng đồng nghiên cứu tóm tắt văn bản.
Có 5 đánh giá ROUGE được đưa ra

5.1. ROUGE- N (N-gram Co-Occurrence Statistics)

ROUGE-N là một thu hồi n-gram giữa một bản tóm tắt tự động và một tập hợp các tài liệu tóm tắt tham khảo summaries. ROUGE-N được tính như sau [21]:

$$ROUGE - N = \frac{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} Count_{\text{match}}(gram_n)}{\sum_{S \in \{\text{ReferenceSummaries}\}} \sum_{gram_n \in S} Count\ (gram_n)}$$

Trong đó: n là chiều dài của n-gram, $Count_{match}(gram_n)$ là số lượng tối đa n-gam có thể sảy ra đồng thời trong bản tóm tắt tự động và bản tóm tắt tham khảo.

Rõ ràng là ROUGE-N là một biện pháp liên quan đến thu hồi vì mẫu số của phương trình là tổng của số n-gram xảy ra ở phía tài liệu tóm tắt tham khảo.

5.2. ROUGE –L (Longest Common Subsequence).

ROUGE-L tính tỷ lệ giữa chiều dài của chung dài nhất 'tóm tắt dãy (LCS) và chiều dài của bản tóm tắt tài liệu tham khảo như mô tả bởi phương trình:

$$R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n}$$

$$F_{lcs} = \frac{(1+\beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

Trong đó: m là độ dài của bản tóm tắt tài liệu tham khảo câu X và n là chiều dài của ứng cử viên câu Y. LCS(X,Y) là độ dài LCS của x và Y. R là sự thu hồi của X và Y, P là độ chính xác giữa X và Y. Tham số được chọn trong DUC là 8.

5.3. ROUGE-W (Weighted Longest Common Subsequence)

ROUGE-W: Trọng số của chuỗi chiều dài lớn nhất, là mở rộng của ROUGE-L:

$$R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right)$$

$$P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right)$$

$$F_{wlcs} = \frac{(1+\beta^2) R_{wlcs} P_{wlcs}}{R_{wlcs} + \beta^2 P_{wlcs}}$$

Trong đó: m là độ dài của bản tóm tắt tài liệu tham khảo câu X và n là chiều dài của ứng cử viên câu Y. LCS(X,Y) là độ dài LCS của x và Y. R là sự thu hồi của X và Y, P là độ chính xác giữa X và Y. Tham số được chọn trong DUC là 8.

5.4. ROUGE –S (Skip-Bigram Co-Occurrence Statistics).

Sử dụng sự chồng chéo của skip-bigram giữa bản tóm tắt ứng cử viên và bản tóm tắt tham khảo:

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)}$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)}$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2}+\beta^2 P_{skip2}}$$

Trong đó: SKIP2(X,Y) là số lượng bigram giữa X và Y. R là thu hồi giữa X và Y, còn P là độ chính xác giữa X và Y.

5.5. ROUGE –SU (Extension of ROUGE-S).

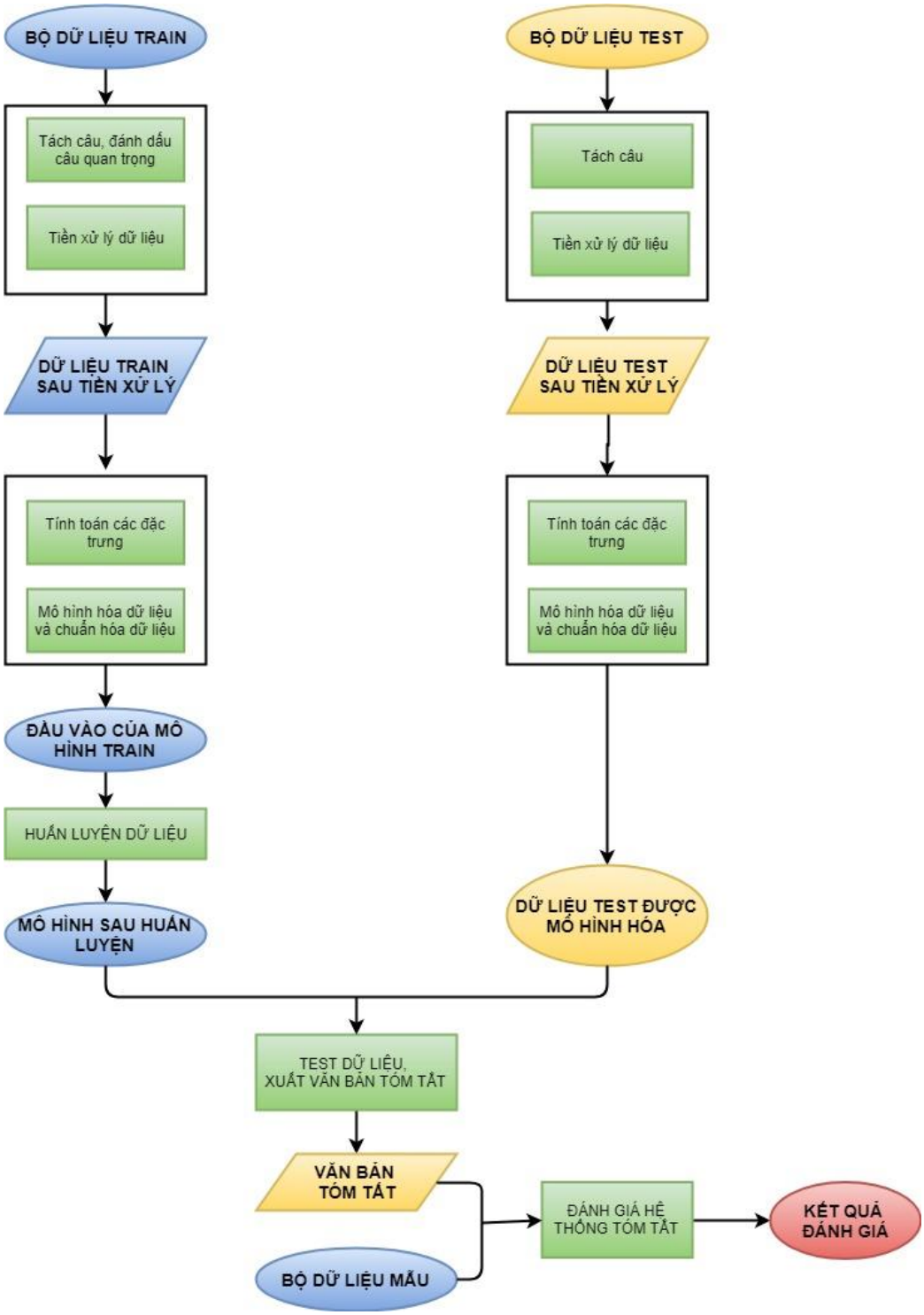
Một vấn đề tiềm năng cho ROUGE-S là nó không cung cấp cho bất kỳ giá trị cho một câu ứng cử viên nếu câu không có bất kỳ cặp từ xảy ra đồng thời với câu tham chiếu. Để đạt được điều này, mở rộng ROUGE-S với việc bổ sung unigram là đơn vị đếm. Các phiên bản mở rộng được gọi là ROUGE-SU. Cũng có thể có được ROUGE-SU từ ROUGE-S bằng cách thêm một dấu hiệu bắt đầu của câu vào lúc bắt đầu của ứng cử viên và câu tham khảo.

⇨ Trong 5 đánh giá ROUGE được đưa ra đối với bài toán tóm tắt văn bản thường sử dụng đánh giá ROUGE-N : tương đương với độ đo Recall. Trong đồ án, em đánh giá kết quả văn bản tóm tắt bằng đánh giá ROUGE-1 unigram và ROUGE-2 bigram

CHƯƠNG III: MÔ HÌNH ĐỀ XUẤT

1. Tổng quan về mô hình đề xuất

Với bài toán tóm tắt văn bản tự động tiếng Anh theo hướng trích rút, sử dụng mô hình SVM , em đề xuất mô hình sau để thử nghiệm cho phương pháp này :



Hình 11: Mô hình đề xuất cho bài toán kiểm thử

Các pha xử lý trong mô hình đề xuất :

- Tách câu và tiền xử lý
- Tính toán các đặc trưng và mô hình hóa dữ liệu
- Huấn luyện và kiểm thử
- Đánh giá kết quả

2. Bộ dữ liệu huấn luyện

2.1. Tổng quan

Trong quá trình nghiên cứu đồ án, em đã tìm hiểu một số tập dữ liệu huấn luyện, trong đó có 2 bộ dữ liệu phù hợp cho bài toán tóm tắt trích rút của mình:

- Bộ dữ liệu Bác Mới:
  Bộ dữ liệu huán luyện với kích thước lớn 3.6GB, bao gồm 1.002.394 bài báo. Độ dài mỗi bài báo từ 300 đến 1500 từ.
- Bộ dữ liệu DUC 2007:
  Một tập các bài báo về 45 chủ đề khác nhau, trong đó có 23 chủ đề đã được tổng hợp và đánh dấu các câu trọng tâm, kèm theo đó là các bản tóm tắt thủ công của chuyên gia.

Bộ dữ liệu Báo Mới là tập dữ liệu phong phú về chủ đề, với những bài báo và thông tin chân thực, thực tế, có tính linh hoạt và ứng dụng lớn. Bên cạnh đó các chương trình tóm tắt văn bản Tiếng Việt còn khá ít, đạt kết cả chưa cao. Việc nghiên cứu thử nghiệm trên bộ dữ liệu Tiếng Việt mang lại nhiều ý nghĩa và ứng dụng trong thực tiễn so với bộ dữ liệu tiếng Anh.

Tuy nhiên các bài báo trong bộ dữ liệu tiếng Việt có độ chênh lệch về độ dài khá lớn, nhiều bài báo có độ dài nhỏ hơn 400 từ, nhiều bài báo có nội dung rời rạc, do đó cần thiết phải lọc và loại bỏ các bài báo không phù hợp với bài toán. Công đoạn này mất nhiều thời gian. Bên cạnh đó, mỗi bài báo chỉ bao gồm 1 câu tiêu đề, có hoặc không có từ 1 đến 2 câu tóm tắt nội dung của bài báo (câu Description), do đó đặt ra vấn đề phải đánh dấu các câu quan trọng trong văn bản dựa vào các câu Description đó, dẫn đến giảm độ chính xác của mô hình huấn luyện so với bộ DUC2007 có đánh dấu. Bộ dữ liệu Báo Mới không có bản tóm tắt thủ công để đánh giá kết quả.

Do đó, trong phạm vi bài toán đưa ra của đồ án, em chọn bộ dữ liệu DUC2007 để huấn luyện và kiểm thử mô hình.

2.2. Cấu trúc bộ dữ liệu DUC2007

DUC2007 bao gồm 2 tập dữ liệu:

Main task:

Tập dữ liệu trong main task được chia thành các thư mục theo chủ đề. Mỗi chủ đề bao gồm 25 văn bản liên quan. Mỗi chủ đề và cụm tài liệu được gửi cho 4 đơn vị đánh giá NIST khác nhau, bao gồm cả nhà phát triển chủ đề đó. Các chuyên gia sẽ tạo ra văn bản tóm tắt khoảng 250 từ của cụm tài liệu, đáp ứng được nhu cầu thông tin được thể hiện trong chủ đề. Và những văn bản tóm tắt của chuyên gia được dùng trong việc đánh giá nội dung, kết quả của các bài tóm tắt của các hệ thống tự động.

==> bộ dữ liệu phù hợp trong phạm vi nghiên cứu của đồ án.

Main task bao gồm 3 tệp dữ liệu:

- Kết quả đánh giá từ chuyên gia
- Tập hợp các kết quả của các hệ thống tóm tắt tự động đã tham gia
- Tập các văn bản được đánh dấu kết quả của chuyện gia và các hệ thống tóm tắt.

Em sử dụng tập (2007 SCU-marked corpus) để huấn luyện trong đồ án của mình.

Update task (pilot):

Mục đích của bộ dữ liệu là tạo ra những văn bản tóm tắt ngắn khoảng 100 từ theo giả định rằng người đọc đã đọc một số tài liệu trước đó.

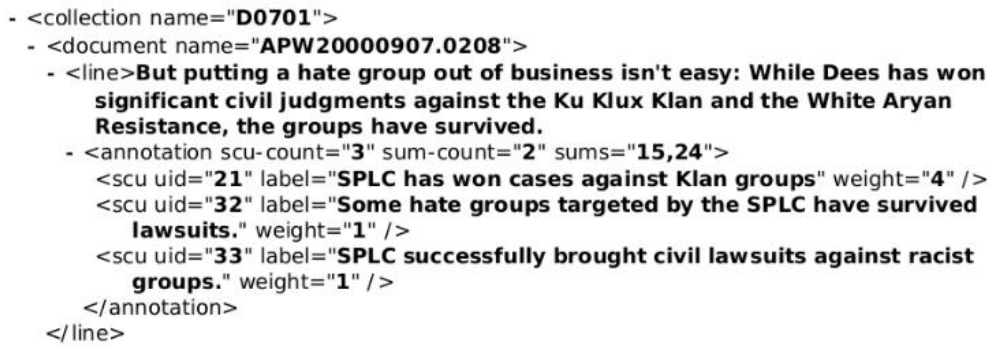
Đối với mỗi chủ đề, các tài liệu sẽ được sắp xếp theo trình tự thời gian và sau đó được chia thành 3 bộ, A,B,C.

Trong đó các dấu thời gian trên tất cả các tài liệu trong mỗi bộ được sắp xếp theo thời gian (A) <thời gian (B) <thời gian (C). Sẽ có khoảng 10 tài liệu trong Bộ A, 8 trong Bộ B và 7 trong Bộ C.

==> Trong phạm vi nghiên cứu của đồ án, bộ dữ liệu Update task (pilot) không phù hợp.

a. 2007 SCU-marked corpus:

- Kho dữ liệu bao gồm một file XML (.scu) cho mỗi chủ đề, trong đó các câu được xác định bởi trình ranh giới câu cục bộ.
- Các câu trong file XML trong mỗi chủ đề được lưu đưới dạng phần tử trong thẻ <line>.
- Các câu quan trọng được đánh dấu bằng thẻ <annotation> gồm 3 giá trị thuộc tính và một hoặc nhiều thẻ <scu> , các trường xuất hiệu trong cấu trúc được chú thích như sau:



Hình 12: Thẻ <annotation> của câu có nhiều SCU

Trong đó:

scu-count: số lượng SCU được nhận bởi câu đó.

Định dạng: số nguyên

Có giá trị bằng với số phần tử SCU

sum-count: số lượng các văn bản tóm tắt đã sử dụng câu đó.

Định dạng: số nguyên

sums: số thứ tự nhận dạng người tham gia ẩn danh, phân cách dấu phẩy.

Số lượng số nhận dạng người tham gia ẩn danh này bằng giá trị sum-count

uid: mã định danh SCU

Định dạng: số nguyên

label: nội dung của SCU

Định dạng: chuỗi

weight: số lượng bản tóm tắt bằng tay, trong đó có thể hiện các SCU

Định dạng: số nguyên.

Hình 12 cung cấp một ví dụ lấy từ chủ đề D0701.

2.3. Ưu, nhược điểm của bộ dữ liệu

Ưu điểm của bộ dữ liệu:

- Bộ dữ liệu có cấu trúc rõ ràng, mạch lạc, dễ phân tích.
- Bộ dữ liệu bao gồm các bảng tóm tắt chuẩn của chuyên gia, phù hợp cho đánh giá kết quả thử nghiệm.
- SCU-marked corpus là bộ dữ liệu tổng hợp, được gán nhán câu quan trọng và câu không quan trọng, phù hợp cho mô hình huấn luyện phân lớp của SVM.

Nhược điểm:

- Kích thước bộ dữ liệu còn khá nhỏ:
- Số lượng phần tử sau khi đã phân tích: 12.832 phần tử (tương đương với 12.832 câu trong các văn bản huấn luyện) trong đó 2/3 số phần tử được dùng cho huấn luyện. 1/3 số phần tử được dùng cho kiểm tra kết quả.

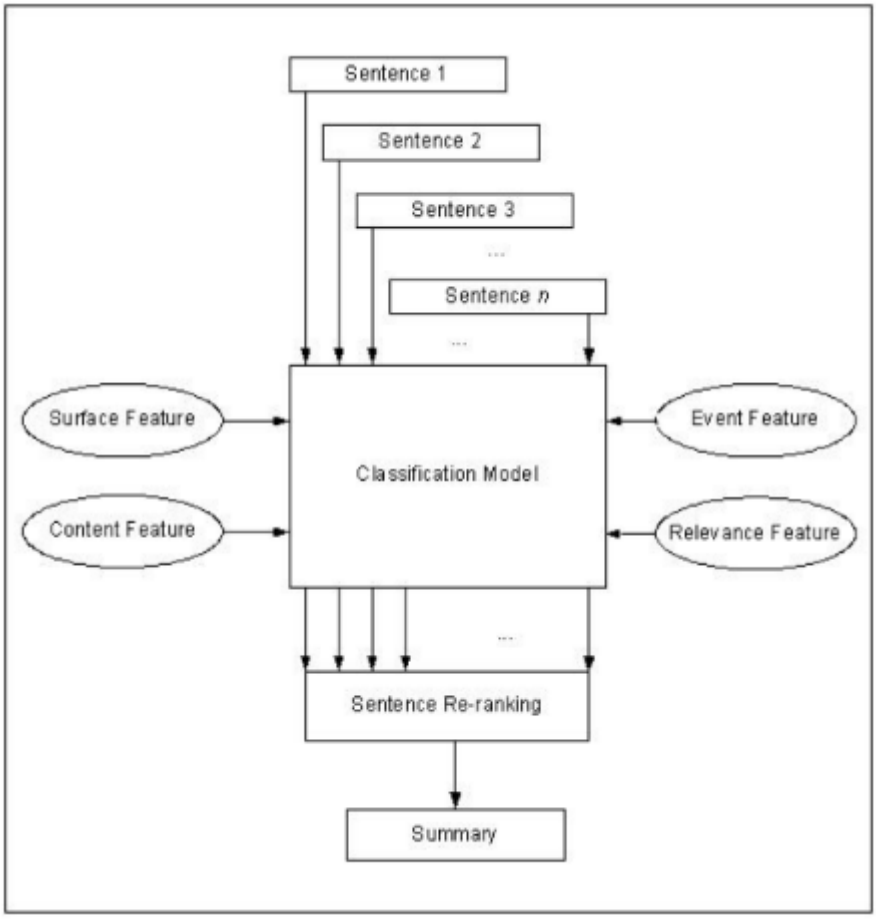
3. Các pha xử lý trong mô hình đề xuất

3.1. Tách câu và tiền xử lý

- Tách câu :
  Tách văn bản thành các đoạn, các câu.
- Đánh dấu câu quan trọng :
  Dựa vào thông số thống kê của tập dữ liệu huấn luyện đánh dấu câu quan trọng.
- Lower case :
  Các từ trong văn bản của bộ dữ liệu được chuyển đổi về chữ thường.
- Loại bỏ dấu câu :
  Các dấu câu như dấu chấm, dấu phẩy, dấu chấm than, dấu chấm hỏi, dấu ba chấm được thay bằng khoảng trắng.
- Loại bỏ stopword :
  Sử dụng từ điển từ dừng.
  Loại bỏ các từ dừng khỏi văn bản.

3.2. Tính toán các đặc trưng và mô hình hóa dữ liệu

Tham khảo bài báo của Kam-Fai Wong, Mingli Wu[1] về tóm tắt văn bản, bài báo đưa ra mô hình chung của tóm tắt văn bản trích rút dựa trên học máy như sau :



Hình 13: Mô hình chung của tóm tắt văn bản trích rút dựa trên học máy.

Trong đó có chỉ rõ ra 4 đặc trưng của câu :

- Surface Feature : đặc trưng bề mặt
- Content Featute : Đặc trưng về nội dung
- Event Feature : Đặc trưng sự kiện
- Relevance Feature : Đặc trưng về mức độ liên quan.

Một số nhà nghiên cứu đã thử nghiệm nhiều phương pháp tóm tắt văn bản trích rút dựa trên học máy, và đưa ra đánh giá. Kết quả đánh giá cho thấy đặc trưng sự kiện (Event Feature) không liên quan, không có tác dụng trong tóm tắt văn bản trích rút.

Các đặc trưng bài báo đưa ra :

STT		Tên các đặc trưng
1	Surface Feature	Position
2		Doc_First
3		Para_First
4		Length
5		Quote
6	Content Feature	Centroid_Uni
7		Centroid_Bi
8		SigTerm_Uni
9		SigTerm_Bi
10		FreqWord_Uni
11		FreqWord_Bi
12	Relevance Feature	FirstRel_Doc
13		FirstRel_Para

Bảng 2: Các đặc trưng của câu

Sau quá trình tìm kiếm và phân tích các bộ dữ liệu phù hợp cho mô hình, em sử dụng bộ dữ liệu DUC2007 để huấn luyện và thử nghiệm.

Bộ dữ liệu huấn luyện của DUC2007, các văn bản không được phân chia thành các đoạn văn, do đó em không xét 2 đặc trưng Para_First : câu đang xét có phải câu đầu đoạn hay không ? Và FirstRel_Para : độ liên quan của câu đang xét với câu đầu đoạn chứa câu đó.

Sau đây em xin trình bày chi tiết về 11 đặc trưng em sẽ sử dụng cho mô hình của mình bao gồm : Position, Doc_First, Length, Quote, Centroid_Uni, Centroid_Bi, SigTerm_Uni, SigTerm_Bi, FreqWord_Uni, FreqWord_Bi, FirstRel_Doc.

Surface features :

Các đặc trưng bề mặt dựa trên cấu trúc của tài liệu hoặc câu, bao gồm vị trí câu trong tài liệu, số từ trong câu, và số từ được trích dẫn trong câu.

Tên	Nội dung
Position	Đặc trưng về vị trí, được tính bằng thương số : 1/vị trí của câu đó trong văn bản
Doc_First	Cho biết câu đó có phải câu đầu tiên của văn bản hay không
Length	Số lượng từ trong câu
Quote	Số lượng từ được trích dẫn trong câu

Bảng 3: Các đặc trưng bề mặt

Content Features :

Đặc trưng về nội dung. Các đặc trưng được tính dựa trên các từ mang nội dung chủ chốt : từ trung tâm (centroid words), các thuật ngữ chữ ký (signature words) và các từ có tần số cao trong văn bản (frequent words) với cả hai đại diện Unigram và Bigram.

Bảng 3 nêu tóm tắt 6 đặc trưng về nội dung được sử dụng trong đồ án :

Tên	Nội dung
Centroid_Uni	Tổng khối lượng của các từ centroid unigram trong câu.
Centroid_Bi	Tổng khối lượng của các từ centroid bigram trong câu.
SigTerm_Uni	Số từ thuật ngữ unigram trong câu.
SigTerm_Bi	Số từ thuật ngữ bigram trong câu.
FreqWord_Uni	Tổng khối lượng các từ unigram có tần số cao trong câu.
FreqWord_Bi	Tổng khối lượng các từ bigram có tần số cao trong câu.

Bảng 4: Các đặc trưng nội dung.

a. Xác định các từ centroid unigram và centroid bigram.

Các từ trung tâm được xác định là 30% số từ xuất hiện trong văn bản có chỉ số TF-IDF lớn nhất.

Cách tính TF-IDF :

- **TF (Term Frequency):**

Là tần suất xuất hiện của một từ trong một đoạn văn bản. Với những đoạn văn bản có độ dài khác nhau, sẽ có những từ xuất hiện nhiều ở những đoạn văn bản dài thay vì những đoạn văn bản ngắn. Vì thế, tần suất này thường được chia cho độ dài của đoạn văn bản như một phương thức chuẩn hóa (normalization).

TF được tính bởi công thức:

$$tf(t) = \frac{f(t,d)}{T}$$

Với t là một từ trong văn bản.

f(t,d) là tần số xuất hiện của d trong đoạn văn bản d.

T là tổng số từ trong văn bản đó.

- **IDF (Inverse Document Frequency):**

Tính toán độ quan trọng của một từ. Khi tính toán TF, mỗi từ đều quan trọng như nhau, nhưng có một số từ trong tiếng Anh như *"is", "of", "that",...* xuất hiện khá nhiều nhưng lại rất ít quan trọng.

Vì vậy, chúng ta cần một phương thức bù trừ những từ xuất hiện nhiều lần và tăng độ quan trọng của những từ ít xuất hiện những có ý nghĩa đặc biệt cho một số đoạn văn bản hơn bằng cách tính IDF:

$$idf(t, D) = \log \frac{|D|}{1 + |\{d \in D: t \in d\}|}$$

Trong đó :

$|D|$ tổng số văn bản trong tập **D**

$|\{d \in D: t \in d\}|$ số văn bản chưa từ nhất định, với điều kiền t xuất hiện trong văn bản d (tức là tf(t,d) ≠ 0). Nếu từ đó không xuất hiện ở bất kỳ một văn bản nào trong tập thì mẫu số bằng 0 => phép chia không hợp lệ, vì thế người ta thường thay bằng mẫu thức $1 + |\{d \in D: t \in d\}|$.

Cơ số logarit trong công thức này không thay đổi giá trị của 1 từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi. (nói cách khác, thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF). Tuy nhiên việc thay đổi khoảng giá trị sẽ giúp tỷ lệ giữa IDF và TF tương đồng để dùng cho công thức TF-IDF như bên dưới.

- **TF-IDF**

  $tfidf(t, d, D) = tf(t, d) \times idf(t, D)$

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

b. Xác định các frequent words unigram và bigram trong văn bản

Các từ có tần số cao được xác định là 30% số từ xuất hiện trong văn bản có chỉ số TF lớn nhất.

Relevance Features

Các tính năng liên quan được kết hợp để khai thác các mối quan hệ giữa các câu. Nó được cho rằng:

(1) các câu liên quan đến các câu quan trọng là quan trọng;

(2) các câu liên quan đến nhiều câu khác là câu quan trọng.

Câu đầu tiên trong một tài liệu hoặc một đoạn là quan trọng, và các câu khác trong một tài liệu được so sánh với các câu hàng đầu.

Trong đồ án của mình, em lựa chọn tính đặc trưng FirstRel_Doc : Độ liên quan của câu hiện tại với câu đầu văn bản, để đại diện cho đặc trưng mức độ liên quan.

⇨ *Tính độ đương đồng giữa hai câu :*
Sử dụng bộ thư viện mã nguồn mở Nltk-examples/src/semantic : Sự tương tự về câu dựa trên ngữ nghĩa và các thống kê Corpus[1]

3.3. Huấn luyện và kiểm thử

SVM[2] xác định một hàm phân tách tuyến tính:

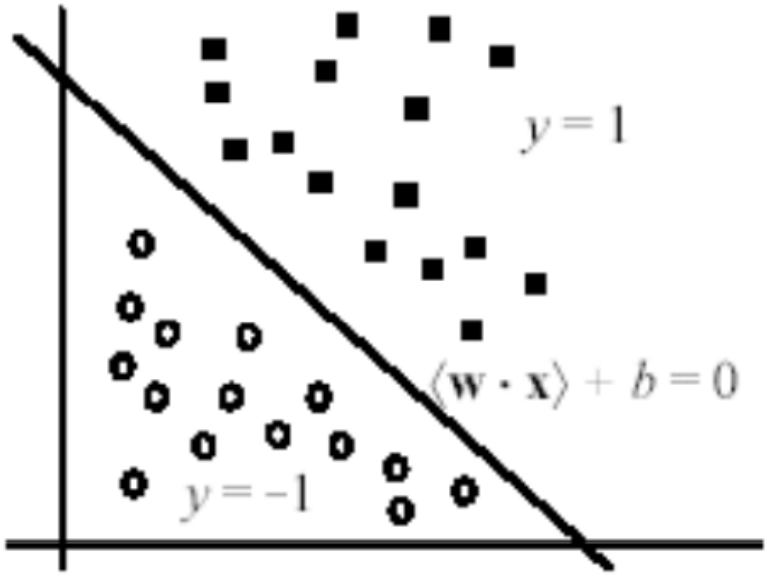
$$f(x) = \langle w.x \rangle + b$$

Trong đó, w là vector trọng số, b là tham số điều chỉnh bias, x là vector đặc trưng.

Mặt siêu phẳng xác định được được dùng để phân tách các ví dụ đầu vào. Nên với ví dụ đầu vào có vector đặc trưng x_i sẽ được gán vào lớp dương nếu $f(x_i) \geq 0$ tức nhãn lớp (ti) là 1 hoặc được gán vào lớp âm, nhãn lớp là 1 nếu ngược lại .

$$t_i = \begin{cases} -1, & \langle w.x_i \rangle + b < 0 \\ 1, & \langle w.x_i \rangle + b \geq 0 \end{cases}$$

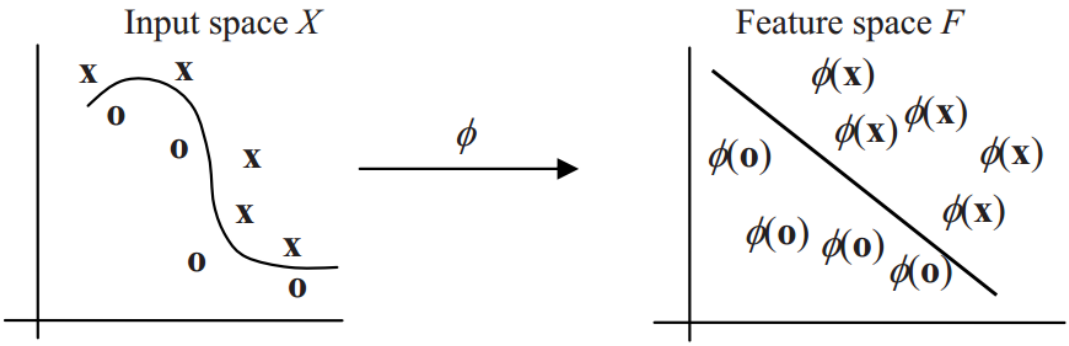
$\langle w.x \rangle + b = 0$ là mặt siêu phẳng phân tách các ví dụ huấn luyện lớp dương và các ví dụ huấn luyện lớp âm. Ví dụ hình 13



Hình 14: Các ví dụ huấn luyện được phân tách bởi mặt siêu phẳng

SVM[2] phân lớp tuyến tính đòi hỏi các ví dụ âm và dương có thể phân tách một cách tuyến tính, ranh giới quyết định là mặt siêu phẳng. Tuy nhiên trong nhiều bài toán thực tế, thì các tập dữ liệu có thể là phân lớp phi tuyến. Để xử lý với dữ liệu phân tách phi tuyến, phương pháp tương tự đối trường hợp phân tách tuyến tính. Ta chuyển các ví dụ từ không gian ban đầu sang không gian đặc trưng mà ranh giới quyết định

tuyến tính có thể phân tách các ví dụ âm và dương trong không gian sau chuyển đổi. Ví dụ hình 14

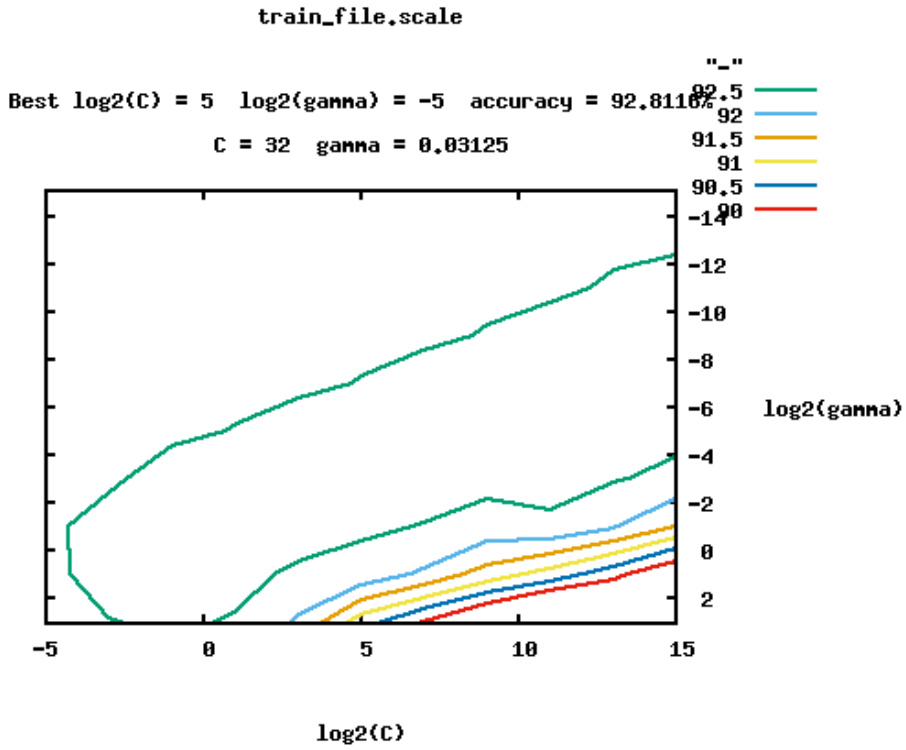


Hình 15: Chuyển đổi các ví dụ từ không gian ban đầu sang không gian đặc trưng

. Tuy nhiên việc chuyển đổi này không được thực hiện rõ ràng mà thay vào đó hàm nhân (kernel function) được sử dụng để tính mà không cần biết hàm chuyển đổi. Trong đồ án này em sử dụng hàm nhân Gauussian RBF (Gausian radial basis function) với γ (gamma) = 0.03125

$$K(x, x') = \exp(-\gamma||x - x'||^2\gamma$$

Em sử dụng hàm nhân này bởi đánh giá thực nghiệm của em cho thấy hàm nhân này có kết quả tốt nhất. Tập nhãn lớp 0 và 1 lần lượt tương ứng với câu không quan trọng và câu quan trọng.



Hình 16: Kết quả tốt nhất của model huấn luyện

Để có thể áp dụng SVM[2] em xác định tập đặc trưng phù hợp với bài toán TTVB và phân tích tập huẩn luyện.Phân tích bộ dữ liệu em trình bài chi tiết trong phần 1 chương III. Do SVM[2] chỉ làm việc với đầu vào là số thực vậy nên những giá trị trong vector đặc trưng em cần phải chuyển các đặc trưng không phải dạng số thực về dạng số thực. Cách để chuyển đổi em sẽ trình bày chi tiết trong phần 4 chương III

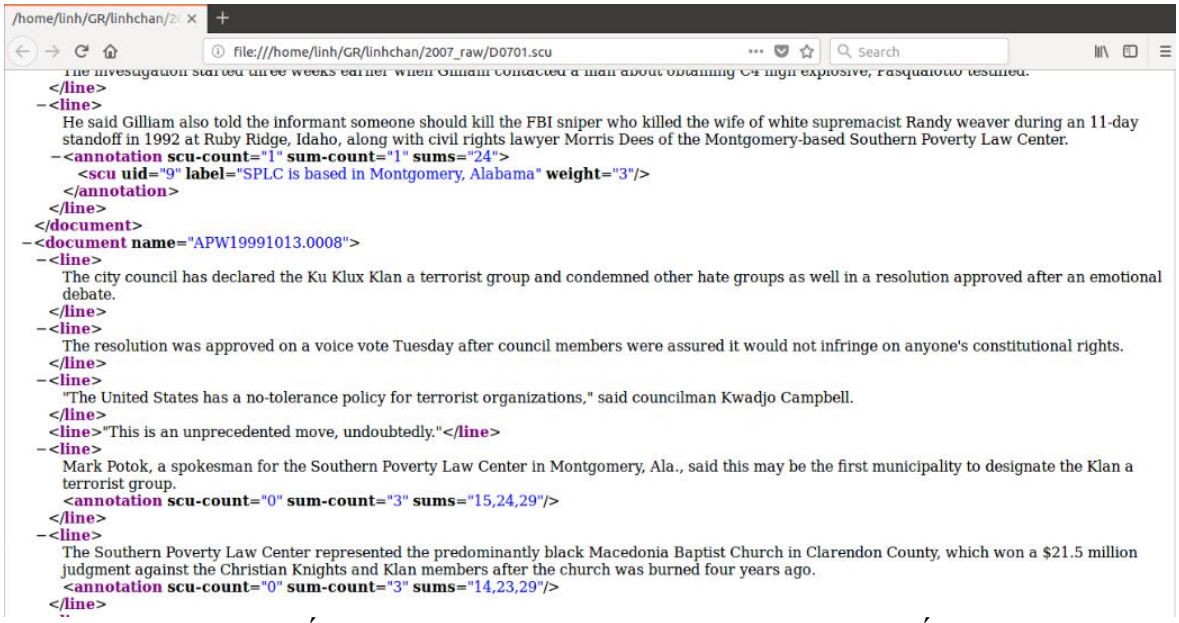
CHƯƠNG IV:CÀI ĐẶT VÀ ĐÁNH GIÁ KẾT QUẢ

1. Cài đặt

1.1. Khối tiền xử lý văn bản

1.1.1 Phân tích các văn bản trong tập huấn luyện

Các văn bản được lưu trữ trong folder 2007_raw của chương trình có cấu trúc như hình dưới đây:



Hình 17: Cấu trúc của một văn bản trong bộ dữ liệu huấn luyện.

Trong đó

- Thẻ <documet>: thẻ mở đầu văn bản, thuộc tính name trong thẻ này cho biết tên (ký hiệu của văn bản)
- Thẻ </ducument>: thẻ kết thúc văn bản.
- Thẻ <line>:thẻ bắt đầu dòng
- Thẻ </line>: thẻ kết thúc dòng
- Thẻ <annotation> nhận biết những câu được đánh dấu trong văn bản tóm tắt của chuyên gia, văn bản tóm tắt của các hệ thống thử nghiệm.
  Các thuộc tính trong thẻ này đã được trình bày chi tiết tại phần 1 của chương.

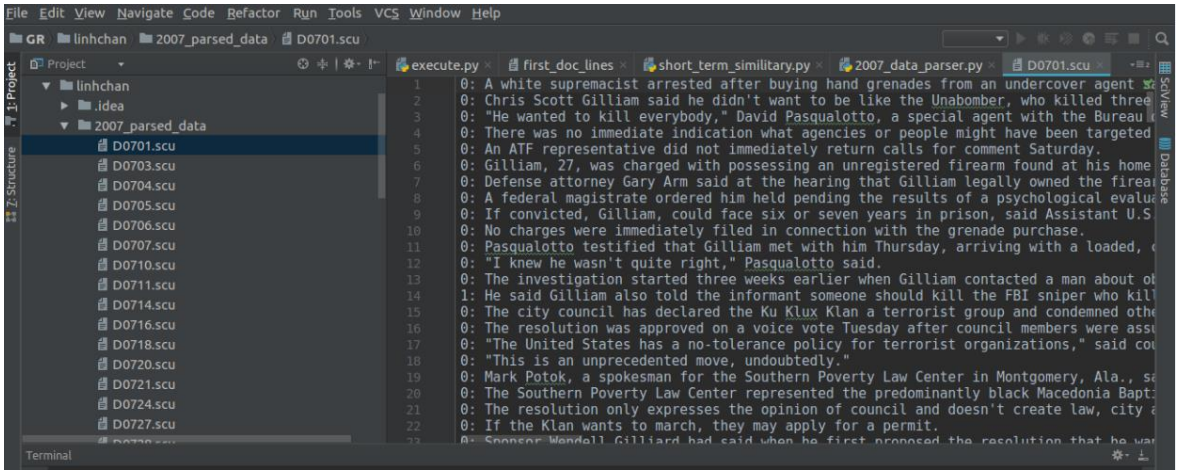
1.1.2. Tiền xử lý

Tách câu và đánh dấu câu quan trọng:

- Chương trình thử nghiệm nhận biết các thẻ <line> và trích rút câu trong thẻ.
- Nhận biết các câu quan trọng thông qua thẻ <annotation>, trong các thuộc tính mà thẻ <annotation> chỉ ra, thuộc tính scu-count cho biết câu đó có được sử dụng trong bản tóm tắt của chuyên gia hay không. Do đó, những câu

có thẻ <annotation> và thuộc tính scu-count > 0 được đánh dấu là câu quan trọng phục vụ huấn luyện.

- Dữ liệu sau khi tách câu và đánh dấu câu quan trọng được lưu trong folder 2007_parsed_data với định dạng như hình dưới đây:

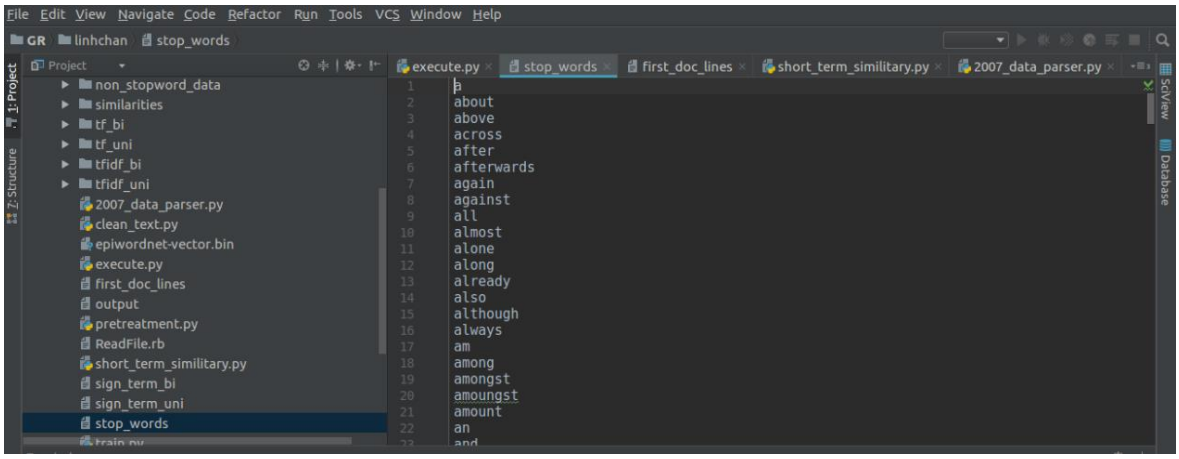


Hình 18: Dữ liệu sau khi tách câu và đánh dấu câu quan trọng.

Loại bỏ từ dừng (stop word) và dấu câu:

Danh sách các từ dừng được lưu trong file stop_words của chương trình.

Bao gồm 319 từ



Hình 19: File lưu danh sách các từ dừng trong tiếng Anh.

Chương trình đọc danh sách các từ trong file này và loại bỏ chúng khỏi các văn bản trong folder 2007_parsed_data, đồng thời thay dấu câu như dầu chấm, dấu chấm than, dấu chấm hỏi, dấu phẩy thành khoảng trắng.

Dữ liệu sau khi loại bỏ từ dừng và dấu câu được lưu trong folder non_stopword_data.

1.2. Mô hình hóa dữ liệu huấn luyện

1.2.1. Tính đặc trưng

11 đặc trưng được lựa chọn cho mô hình huấn luyện

No	Tên đặc trưng	No	Tên đặc trưng
1	Position	7	SigTerm_Uni
2	Doc_First	8	SigTerm_Bi
3	Length	9	FreqWord_Uni
4	Quote	10	FreqWord_Bi
5	Centroid_Uni	11	FirstRel_Doc
6	Centroid_Bi		

Bảng 5: Các đặc trưng của câu trong mô hình đề xuất.

Đặc trưng 1: Position
Cách tính đặc trưng Position: Giá trị nghịch đảo vị trí của câu đó trong văn bản chưa nó.
Gọi đến hàm Feature_Position(sentence, filename)
Đầu vào: câu và tên file chứa câu đó.
Đặc trưng này trả về giá trị: 1/vị trí của câu đó trong văn bản

Đặc trưng 2: Doc_First
Giá trị của đặc trưng này là 0 hoặc 1. Trả về 1 khi câu đó là câu đầu của văn bản.
Hàm def Feature_docfirst(sentence, docFirsts):
Đầu vào: một câu trong văn bản, mảng các câu đầu trong văn bản.
Hàm trả về giá trị 0 hoặc 1, là 1 khi câu đó là câu đầu của văn bản.
Các hàm hỗ trợ cho tính năng này:

- def get_first_doc()
  Đọc file xml từ 2007_raw ra file text first_docc_line tất cả các dòng đầu tiên của văn bản, đầu văn bản được đánh dấu bằng thẻ <document>
- def get_docfirst_list():
  Trả về 1 mảng các câu đầu đoạn, đọc từ file first_doc_line

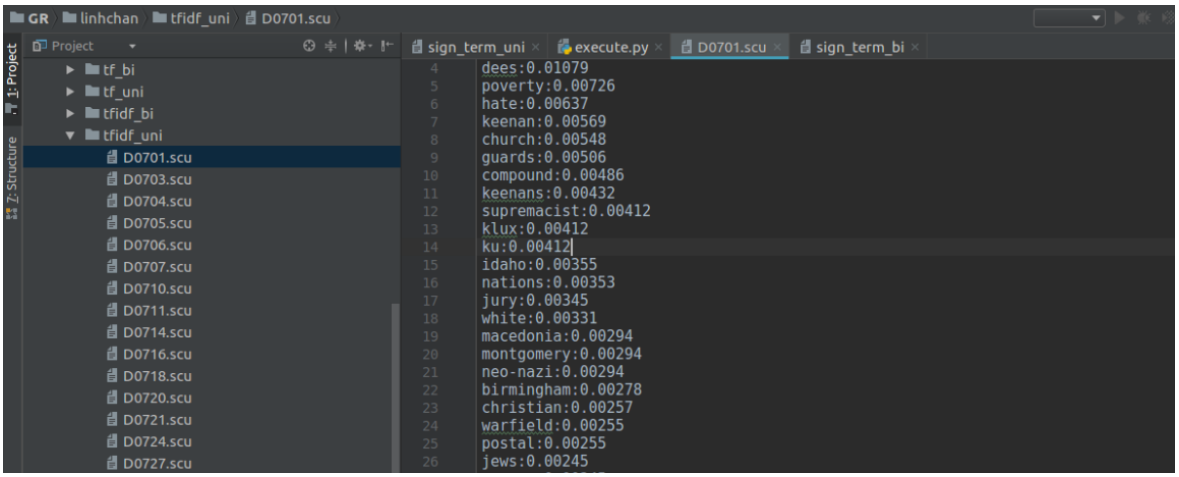
Đăch trưng 3: Length
Giá trị của đặc trưng: Độ dài của câu.
Hàm Feature_Length(sentence):
Đầu vào là 1 câu trong văn bản
Hàm trả về độ dài của câu

Đặc trưng 4: Quote
Giá trị của đặc trưng: Số lượng từ trích dẫn nằm trong câu.
Phương pháp tính: Tìm và tính số lượng của tất cả các từ nằm trong các cặp dấu đóng mở ngoặc đơn (), nháy kép “”, đóng mở ngoặc vuông, dấu *.
Hàm Feature_quote(sentence):
Đầu vào là 1 câu trong văn bản
Trả về số từ trích dẫn trong câu đó

Đặc trưng 5: Centroid_Uni
Tổng khối lượng của các từ trọng tâm Uni (từ centroid uni) có trong câu:
Các từ centroid được xác định bằng độ đo TF-IDF

Các hàm hỗ trợ tính độ đo TF-IDF unigram:

- def get_bloblist(folder):
  Hàm trả về 1 mảng các chuỗi, trong đó mỗi chuỗi là 1 văn bản, đầu vào là non-stopword-data hoặc bigram-data
- def tf(word, blob):
  Đầu vào: Một từ trong văn bản và văn bản đó.
  Trả về giá trị tf của từ đó trong văn bản blob.
- def n_containing(word, bloblist)trả về số lượng văn bản chứa word. bloblist là 1 mảng các văn bản.
  Đầu vào: một từ và mảng các văn bản trong tập huấn luyện hoặc tập test.
  Trả về số lượng văn bản chứa từ đó.
- def idf(word, bloblist):
  Đầu vào: Một từ và mảng các văn bản trong tập huấn luyện hoặc tập test.
  Trả về giá trị idf của từ đó trong văn bản blob.
- def tfidf(word, blob, bloblist):
  Đầu vào: Một từ, văn bản chứa từ đó và mảng các văn bản trong tập huấn luyện hoặc tập test.
  Trả về giá trị TF-IDF của từ đó.
- def tfidf_data(ngram):
  Đầu vào là lựa chọn unigram(1) hoặc bigram(2)
  Hàm ghi lại kết quả chỉ số IF-IDF của 30% tổng số từ có có chỉ số cao nhất (các từ centroid words) vào từng văn bản vào mỗi file riêng biệt trong folder TFIDF_UNI hoặc TFIDF_BI



Hình 20: Kết quả TI-IDF được lưu trong folder tfidf_uni

Hàm def Centroid(sentence, word_list):
Sẽ đọc trong file lưu chỉ số TF-IDF của từ và tính toán tổng trọng lượng các từ centroid trong câu.

Đặc trưng 6: Centroid_Bi

Tổng khối lượng của các từ trọng tâm Bi (từ centroid Bi) có trong câu:
Các hàm hỗ trợ xuất file bigram:

- def to_bigram(sentence):
  Đầ vào: một câu trong văn bản.

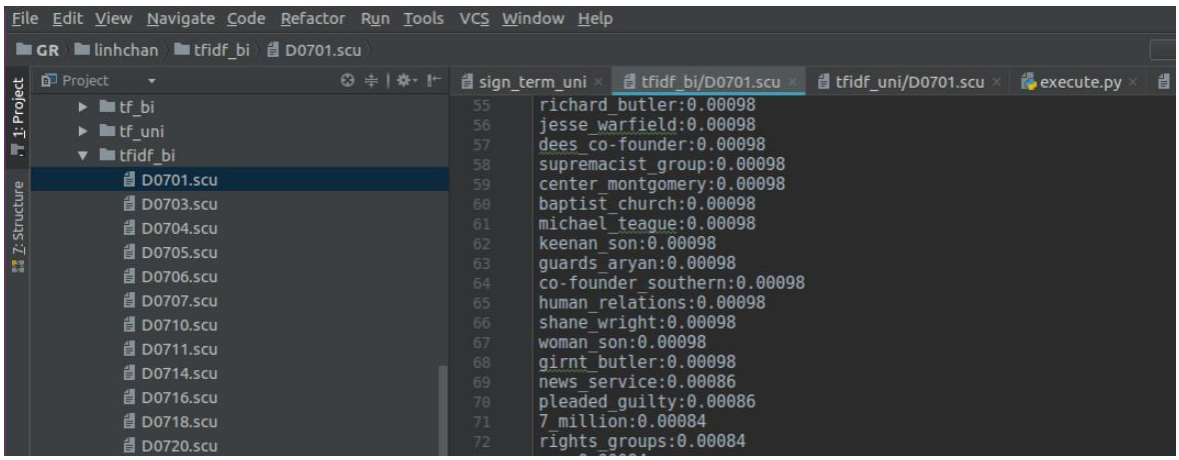
Trả về 1 chuỗi thay thế sentence gồm các bigram.

- def parse_to_bigram_data():
  Ghi lại các câu sau khi biến đổi sang bigram vào foder bigram-data

Các từ centroid-bi được xác định bằng độ đo tf-idf

Các hàm hỗ trợ tính TF-IDF bigram

- def get_bloblist(folder):
  Hàm trả về 1 mảng các chuỗi, trong đó mỗi chuỗi là 1 văn bản, đầu vào là non-stopword-data hoặc bigram-data
- def tf(word, blob):
  Đầu vào: Một từ trong văn bản và văn bản đó.
  Trả về giá trị tf của từ đó trong văn bản blob.
- def n_containing(word, bloblist)trả về số lượng văn bản chứa word. bloblist là 1 mảng các văn bản.
  Đầu vào: một từ và mảng các văn bản trong tập huấn luyện hoặc tập test.
  Trả về số lượng văn bản chứa từ đó.
- def idf(word, bloblist):
  Đầu vào: Một từ và mảng các văn bản trong tập huấn luyện hoặc tập test.
  Trả về giá trị idf của từ đó trong văn bản blob.
- def tfidf(word, blob, bloblist):
  Đầu vào: Một từ, văn bản chứa từ đó và mảng các văn bản trong tập huấn luyện hoặc tập test.
  Trả về giá trị TF-IDF của từ đó.
- def tfidf_data(ngram):
  Đầu vào là lựa chọn unigram(1) hoặc bigram(2)
  Hàm ghi lại kết quả chỉ số TF-IDF của 30% tổng số từ có có chỉ số cao nhất (các từ centroid words) vào từng văn bản vào mỗi file riêng biệt trong folder TFIDF_UNI hoặc TFIDF_BI



Hình 21: Kết quả tính TF-IDF được lưu trong folder tfidf_bi

- Hàm def Centroid(sentence, word_list):

Sẽ đọc trong file lưu chỉ số TF-IDF của từ và tính toán tổng trọng lượng các từ centroid trong câu.

Đặc trưng 7: SigTerm_Uni

Dánh sách các từ SigTerm _Uni tự tổng hợp

No	SigTerm _Uni	No	SigTerm _Uni
1	abstract	5	conclusion
2	brief	6	outline
3	highlight	7	summarization
4	conclude	8	summary

Bảng 6: Bảng các SigTerm_uni

Hàm hỗ trợ:

- def get_signterm_data(filename):
  Đầu vào: đọc file sigterm_data
  Trả về 1 mảng các từ sigterm
- def SigTerm(sentence, sigterms):
  Đầu vào: một câu trong văn bản và mảng các từ sigterm
  Trả về số lượng từ signterm trong câu

Đặc trưng 8: SigTerm_Bi

Dánh sách các từ SigTerm _Bi tự tổng hợp

No	SigTerm _Bi
1	main idea
2	the best
3	this article
4	this document
5	this paper

Bảng 7: Sigterm_bigram

Hàm hỗ trợ:

- def get_signterm_data(filename):
  Đầu vào: đọc file sigterm_data
  Trả về 1 mảng các từ sigterm
- def SigTerm(sentence, sigterms):
  Đầu vào: một câu trong văn bản và mảng các từ sigterm
  Trả về số lượng từ signterm trong câu

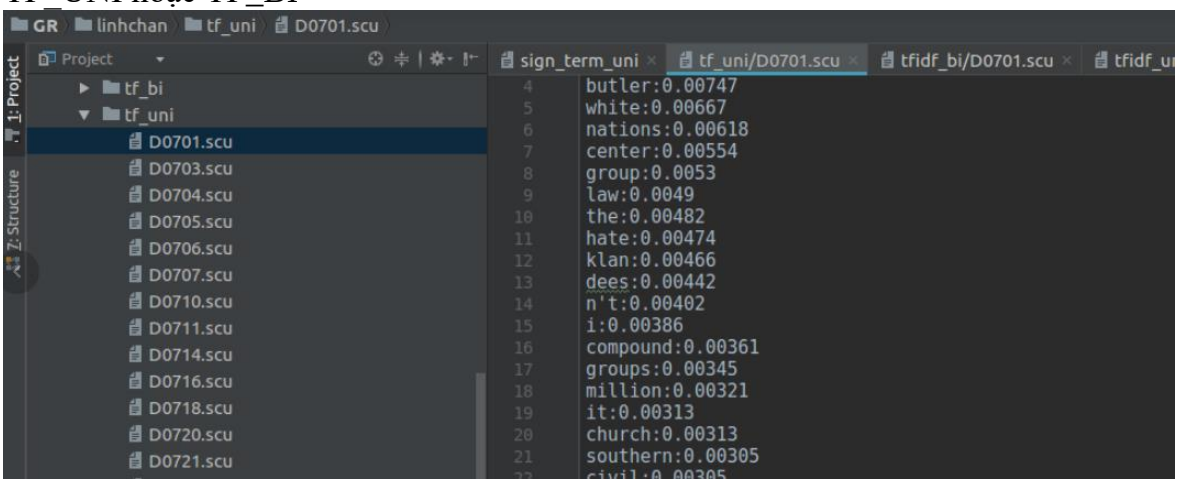
Đặc trưng 9: FreqWord_Uni

Các FreqWord_Uni trong mỗi văn bản được xác định là 30% các từ Unigram có chỉ số TF cao nhất trong mỗi văn bản

Các hàm hỗ trợ tính chỉ số TF unigram

- def tf(word, blob):
  Đầu vào: Một từ trong văn bản, và văn bản chứa từ đó.
  Trả về chỉ số TF của từ đó trong văn bản.
- def tf_data(ngram):
  Đầu vào là lựa chọn unigram(1) hoặc bigram(2)

Hàm ghi lại kết quả chỉ số TF của 30% tổng số từ có có chỉ số cao nhất (các từ frequent words) vào từng văn bản vào mỗi file riêng biệt trong folder TF_UNI hoặc TF_BI



Hình 22: Kết quả tính TF được lưu trong folder tf_uni

- Hàm def FreqWord(sentence, word_list):

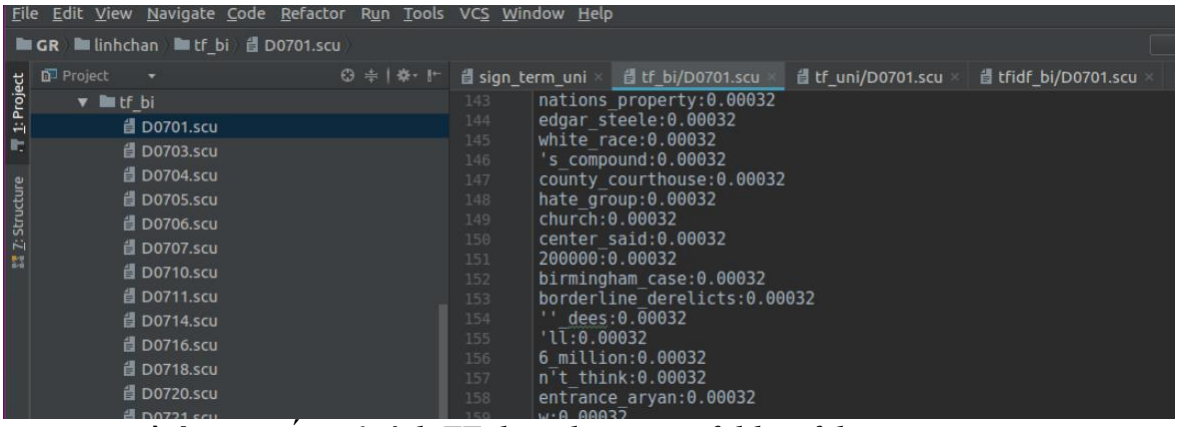
Sẽ đọc trong file lưu chỉ số TF của từ và tính toán tổng trọng lượng các từ freq_word trong câu.

Đặc trưng 10: FreqWord_Bi

Các FreqWord_Bi trong mỗi văn bản được xác định là 30% các từ Bigram có chỉ số TF cao nhất trong mỗi văn bản

Các hàm hỗ trợ tính chỉ số TF bigram

- def tf(word, blob):
  Đầu vào: Một từ trong văn bản, và văn bản chứa từ đó.
  Trả về chỉ số TF của từ đó trong văn bản.
- def tf_data(ngram):
  Đầu vào là lựa chọn unigram(1) hoặc bigram(2)
  Hàm ghi lại kết quả chỉ số TF của 30% tổng số từ có có chỉ số cao nhất (các từ frequent words) vào từng văn bản vào mỗi file riêng biệt trong folder TF_UNI hoặc TF_BI



Hình 23: Kết quả tính TF được lưu trong folder tf_bi

- Hàm def FreqWord(sentence, word_list):

Sẽ đọc trong file lưu chỉ số TF của từ và tính toán tổng trọng lượng các từ freq_word trong câu.

Đặc trưng 11: FirstRel_Doc

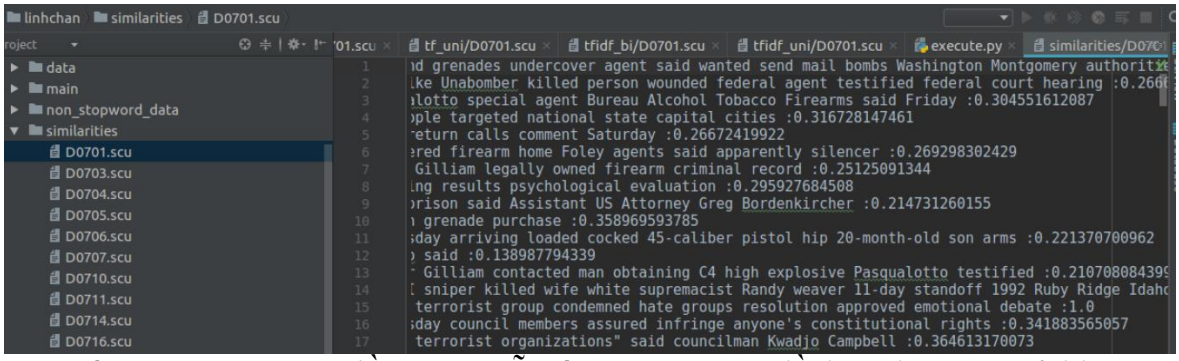
Thư viện hỗ trợ

Sử dụng bộ thư viện mã nguồn mở Nltk-examples/src/semantic : Sự tương tự về câu dựa trên ngữ nghĩa và các thống kê Corpus[1]

Các hàm hỗ trợ

- def get_docfirst_list():
  Trả về 1 mảng các câu đầu đoạn, đọc từ file first_doc_line
- def is_first_doc(sentence, doc_first_list):
  Đầu vào: một câu trong văn bản và mảng các câu đầu văn bản
  Kiểm tra 1 câu có phải câu đầu đoạn
- def cal_similarity():

Ghi vào folder similities mỗi câu và độ tương đồng của câu đó với câu tiêu đề kết quả được lưu trong folder similities



Hình 24: Độ tương đồng của mỗi câu với câu tiêu đề được lưu trong folder similities

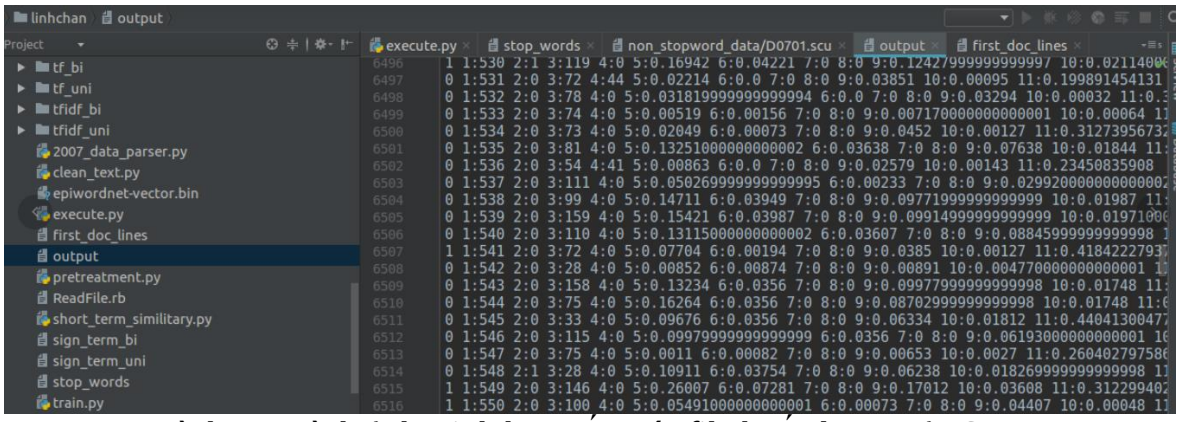
- Hàm def Feature_firstRelDoc(sentence, filename):
  Đầu vào: một câu trong văn bản và tên file văn bản tương ứng trong thư mục similarties
  Trả về giá trị độ tương đồng của câu đó với câu đầu đoạn.

1.2.2. Xuất file huấn luyện

Như đã trình bày, em sử dụng SVM[2] để phân loại từ quan trọng và từ không quan trọng. Từ trong quan trong sẽ bị loại bỏ đi, từ quan trọng được đưa vào văn bản tóm tắt. Em sử dụng thư viện LIBSVM của Chih-Chung Chang and Chih-Jen Lin [4][2]. Trong thư viện này định dạng của ví dụ huấn luyện và ví dụ kiểm thử là:

<label> <index1>:<value1> <index2>:<value2>…

Định dạng file phù hợp cho mô hình huấn luyện của SVM:



Hình 25: Hình ảnh minh họa cấu trúc file huấn luyện của SVM

- Mỗi dòng chứa một đối tương và được kết thúc bằng ký tự xuống dòng '\n'
- Trong đó:

<label> là một số nguyên cho biết nhãn lớp

Mỗi cặp <index>: <value> biểu thị cho một tính năng. Giá trị <index> là một số nguyên bắt đầu từ 1 và <value> là một số thực.

Giải pháp đề xuất đưa ra 11 đặc trưng của câu trong bộ dữ liệu. 11 đặc trưng này sẽ tương ứng với 11 cặp <index>: <value> của mỗi câu. <label> có giá trị bằng 0 hoặc 1, với 1 là câu được gán nhãn quan trọng.

1.3. Chuẩn hóa bộ dữ liệu

Svm-scale là công cụ thể chuẩn hóa dữ liệu đầu vào về giá trị trong khoảng [0;1]. Công thức chuẩn hóa:

$$\text{value}_c = \frac{l + (u - l) * (\text{value} - \text{f\_min}_{\text{index}})}{(\text{f\_max}_{\text{index}} - \text{f\_min}_{\text{index}})}$$

Trong đó, value_c là giá trị sau chuẩn hóa của đặc trưng thứ index, value là giá trị trước chuẩn hóa của đặc trưng thứ index, l, u lần lượt là giá trị nhỏ nhất và lớn nhất có thể sau chuẩn hóa, $\text{f\_max}_{\text{index}}, \text{f\_min}_{\text{index}}$ lần lượt là giá trị lớn nhất và giá trị của nhất của đặc trưng ở vị trí thứ index trên toàn bộ tập dữ liệu. Các giá trị $\text{f\_min}_{\text{index}}$ và $\text{f\_max}_{\text{index}}$ sẽ được lưu vào tệp "range" để phục vụ cho thao tác chuẩn hóa sau này.

Như em đã trình bày trong phần 3.2 của đồ án

Cặp **<index>:<value>** là một biểu diễn của một đặc trưng. Trong đó **<index>** là một giá trị số nguyên bắt đầu từ 1 và **<value>** là một số nguyên biểu diễn giá trị của đặc trưng đó. Ví dụ dãy các đặc trưng của một câu trong văn bản huấn luyện:

1:1 2:0 3:106 4:7 5:0.03529 6:0.00196 7:0 8:0 9:0.03713 10:0.00131 11:0.513208063011

Sau khi được chuẩn hóa trở thành:

1:-0.998261 2:-1 3:-0.448819 4:-0.942623 5:-0.823717 6:-0.962524 7:-1 9:-0.767195 10:-0.95958 11:0.0264161

1.4. Học từ bộ dữ liệu huấn luyện

Svm-train được sử dụng để học các quy tắc, luật từ bộ dữ liệu huấn luyện. Đầu vào của svm-train là tập dữ liệu huấn luyện xây dựng đã được chuẩn hóa và đầu ra là một mô hình. Mô hình này được lưu vào tệp train_file.model để gán nhãn lớp cho các ví dụ sau này.

Tuy nhiên, nếu chỉ phân lớp câu thành 2 lớp đơn thuần là câu quan trọng và không quan trọng sẽ nảy sinh một vấn đề có những văn bản không có câu nào được dự đoán là quan trọng.

Để giải quyết vấn đề này, SVM[2]hỗ trợ dự đoán xác suất

Cấu trúc dòng lệnh train:

```
svm-train [options] training_set_file [model_file]
```

Trong đó với tùy chọn:
-b probability_estimates: khi cài đặt tùy chọn này là 1 (mặc định là 0), model train hỗ trợ ước tính xác suất.

1.5. Gán nhãn dữ liệu và dự đoán xác suất

Svm-predict sử dụng mô hình đã học được để gán nhãn lớp cho các câu trong văn bản test. Em tính các giá trị đặc trưng của từng câu trong các văn bản. Sau đó, em chuẩn hóa chúng để làm đầu vào của svm-predict với tham số tùy chọn -b probability_estimates bằng 1, các câu trong văn bản thử nghiệm đã được gán nhãn lớp và ước tính xác suất chính là đầu ra.

Sau khi các câu trong tập văn bản thử nghiệm được gán nhãn và dự đoán xác suất xuất hiện trong các lớp, em sắp xếp chúng theo thứ tự giảm dần của xác suất nằm trong lớp câu quan trọng, từ đó xuất ra văn bản tóm tắt với độ dài khoảng 250 chữ.

2. Đánh giá kết quả

2.1. Bộ dữ liệu mẫu

Bộ dữ liệu mẫu được sử dụng cho quá trình kiểm thử trong đồ án tốt nghiệp này là các văn bản được tóm tắt thực hiện bởi chuyên gia từ các Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST).

Mỗi chuyên gia sẽ có một văn bản tóm tắt thủ công khác nhau. Do đó để đảm bảo tính khách quan, mỗi một văn bản trong bộ dữ liệu test cần có 2 đến 4 văn bản tóm tắt thủ công tương ứng.

Bộ dữ liệu mẫu gồm các bài tóm tắt thủ công của 45 chủ đề.

Mỗi chủ đề có 3 bản tóm tắt thủ công, mỗi bản tóm tắt có độ dài 250 từ.

2.2. Phương pháp đánh giá

Đánh giá kết quả tóm tắt văn bản là một việc làm khó khăn trong thời điểm hiện tại. Việc sử dụng ý kiến đánh giá của các chuyên giá ngôn ngữ được xem là cách đánh giá tốt nhất, tuy nhiên, cách làm này lại tốn rất nhiều chi phí. Bên cạnh các phương pháp đánh giá thủ công do cách chuyên gia thực hiện, vấn đề đánh giá tự động kết quả tóm tắt cũng nhận được nhiều sự chú ý hiện nay. Ví dụ, Saggion (2002)[6] đưa ra ba phương pháp đánh giá tóm tắt văn bản dựa vào nội dung đo độ tương tự giữa văn bản tóm tắt bằng tay và văn bản tóm tắt tự động. các phương pháp đó là: độ tương tự cosine, sự trùng lặp đơn vị (unit overlap) và chuỗi con chung dài nhất. Tuy nhiên, chúng vẫn chưa tương quan với đánh giá của con người. Sau thành công của ứng dụng đánh giá tự động trong đánh giá dịch máy như BLEU (Papineni – 2001)[7], Lin và Hovy đã đưa ra một phương pháp tương tự BLEU như thống kê trùng lặp n-gram có thể được áp dụng vào đánh giá tóm tắt tự động. Đó chính là độ đo ROUGE-N (Recal-Oriented Understudy for Gisting Evaluation)[5]. Trong đồ án này, em sử dụng độ đo ROUGE-N[5] để đánh giá kết quả của hệ thống tóm tắt văn bản tự động. ROUGE-N[5] là độ phủ n-gram giữa văn bản tóm tắt tự động và văn bản tóm tắt bằng tay tương ứng. ROUGE-N[5] được tính như sau:

$$ROUGE - N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

Trong đó:

- $gram_n$ là bộ n từ liên tiếp trong văn bản S.
- $Count_{match}(gram_n)$ là số trùng lặp $gram_n$ tối đa giữa văn bản tóm tắt tự động và văn bản tóm tắt bằng tay.

2.3. Các kết quả kiểm thử

Trong phần này, em thực hiện kiểm thử với 10 văn bản trong tập dữ liệu DUC2007

Để hiểu rõ hơn cho hệ thống của mình em xin đưa ra một ví dụ minh họa đầu vào, đầu ra của hệ thống tóm tắt đơn văn bản theo phương pháp SVM ước lượng xác suất:

Văn bản đầu vào:

Văn bản D0730

Văn bản có độ dài khoảng 10.000 từ.

Văn bản mẫu kiểm thử1:

In the words of President Clinton the line item veto "is very important in helping to preserve the integrity of federal spending".

The line item veto has been sought by presidents since Grant and was popularized by Reagan.

It was part of Republican "Contract with America" led by Speaker Newt Gingrich that enacted it.

The line item veto allows the president to veto particular items in spending bills and certain limited tax provisions passed by Congress.

Previously the president could only veto entire bills.

Bill Clinton is the only president to have had line item veto authority.

He has said that it should be used sparingly.

He used it 163 times, mostly to delete items from the military construction bill.

The line item veto was challenged by a group of most Democratic senators but was dismissed by the Supreme Court.

However, another challenge led by New York Mayor Giuliani and Idaho farmers resulted in a federal judge declaring the line item veto unconstitutional.

The Justice Department appealed that decision to the Supreme Court.

The Supreme Court rejected the line item veto as a departure from the basic constitutional requirement that presidents accept or reject bills in their entirety.

The Court found that the line item veto violates the "presentment clause" of Article I that establishes the process by which a bill becomes law.

The Court vote was 6-3 with Justice Stevens writing for the majority.

Văn bản mẫu kiểm thử2:

The line-item veto (LIV) has been sought by nearly every president this century as a tool to limit pork barrel spending which is traditionally reviled as the most cynically deployed and least utilitarian form of largess.
The 1998 budget included $300,000 for enhancing the flavor of peanuts, $150,000 for peanut competitiveness and $250,000 for pickle research.
President Clinton said the LIV is an important tool for striking unnecessary spending, for preserving the integrity of federal spending and enlivening the public debate over how to make the best use of public funds.
The Solicitor General contended that the LIV represents a presidential exercise of spending authority delegated by Congress.

110 years ago, Lord Bryce said the LIV was "desired by enlightened men and would save the nation millions of dollars a year".
The LIV is a prerogative given to 43 governors.
A Republican-controlled Congress passed the Line-Item Veto Act which went into effect January 1997.
President Clinton used the authority to veto 82 items in 11 bills, including money for New York hospitals, a tax break for Idaho potato growers, 38 projects worth $287M in military construction, $144M from a defense spending bill and $30M for intercepting asteroids.
The Line-Item Veto Act was overturned by U.S. District Judge Thomas Penfield Jackson in April 1997 and later ruled 6-3 as unconstitutional by the U.S. Supreme Court because it allows the president to alter a law after passage, thereby creating a new law not voted on by either house of Congress.

<u>Văn bản mẫu kiểm thử3:</u>

Congress passed a law authorizing the line item veto (LIV) in 1996 accepting arguments that the measure would help preserve the integrity of federal spending by allowing the president to strike unnecessary spending and tax items from legislation thus encouraging the government to live within its means.

It was considered in line with the historic relationship between Congress and the president and would provide a tool for eliminating wasteful pork barrel spending while enlivening debate over the best use of funds.

It was argued that the LIV would represent presidential exercise of spending authority delegated by Congress.

President Clinton exercised the LIV on 82 items in 1997 saving $1.9 billion in spending projected over five years.

The affected items were projects for specific localities, many in the area of military construction, which had been added to the president's budget by Congress.

The first court ruling on the LIV act was in U.S. District Court when in February 1998 it was ruled unconstitutional on the grounds that it violated the separation of powers.

The Department of Justice appealed that decision and in June 1998 the Supreme Court ruled the LIV act unconstitutional but on the grounds that it violated Article I, 7, Clause 2 (The "presentment clause") of the Constitution that establishes the process by which a bill becomes law.

President Clinton expressed his deep disappointment.

<u>Văn bản tóm tắt tự động bằng SVM ước lượng xác suất:</u>

The Supreme Court struck down as unconstitutional a law giving the president a line-item veto, which let him cancel specific items in tax and spending measures.

The Supreme Court ruled Thursday that Congress violated the principle of separation of powers when it gave the president the power to cut individual projects from tax and spending bills without vetoing the entire legislation. The Supreme Court ruled Thursday that Congress violated the principle of separation of powers when it gave the president the power to cut individual projects from tax and spending bills without vetoing the entire legislation. The Supreme Court, in a historic ruling affecting the constitutional balance of power at the heart of the American political system, struck down legislation Thursday that permits the president to veto specific provisions of spending and tax bills. U.S. President Bill Clinton said Friday he will appeal a federal judge's ruling that struck down a law giving the president the power to veto specific items in bills passed by Congress. U.S. President Bill Clinton said today he has not yet decided whether to strike out individual items in the comprehensive tax and budget bills awaiting his signature, noting that his new line-item veto authority ought to be used "somewhat sparingly." U.S. President Bill Clinton today used his line-item veto authority to have trimmed 144 million U.S. dollars from a 248 billion dollars defense spending bill. The U.S. Supreme Court Thursday struck down as unconstitutional the line-item veto law that lets the U.S. President strike out specific items in tax and spending measures.

Bảng 8: Một ví dụ tóm tắt của hệ thống

Kết quả kiểm thử trung bình trên 7 văn bản như sau:

SVM-ước lượng xác suất	Recall
ROUGE-1	0.39713
ROUGE-2	0.10085

Bảng 9: Kết quả đánh giá ROUGE của các văn bản tóm tắt

Hệ thống sẽ tóm tắt văn bản với độ dài khoảng 250 từ tương đương với độ dài văn bản được tóm tắt bởi con người. Giá trị ROUGE-1 được xác định là tương đương với đánh giá của con người thông qua những thống kê. Do đó, trong thi nghiệm này em sử dụng độ đo ROUGE-1 để đánh giá tóm tắt tự động. Cùng với ROUGE-1, tôi cũng sử dụng ROUGE-2 để tăng độ tin tưởng của đánh giá.

Mỗi chuyên gia sẽ có một văn bản tóm tắt thủ công khác nhau. Do đó để đảm bảo tính khách quan, mỗi một văn bản trong bộ dữ liệu test cần có 2 đến 4 văn bản tóm tắt thủ công tương ứng. Mỗi một văn bản tóm tắt tự động sẽ được đánh giá dựa trên 3 bản tóm tắt thủ công của con người.

So sánh với kết quả kiểm thử của những phương pháp tiếp cận khác trên cùng tập dữ liệu:

- Kết quả kiểm thử của các phương pháp tiếp cận NBC, COT theo bài báo Extractive Summarization Using Supervised and Semi-supervised Learning của Kam-Fai Wong, Mingli Wu.
- Kết quả kiểm thử của phương pháp tiếp cận Naïve Bayes và Naïve Bayes+Adaboost – đồ án nghiên cứu của sinh viên Vũ Thu Hiền – IS K58 cùng phòng nghiên cứu.

	ROUGE-1	ROUGE-2
NBC	0.353	0.061
COT	0.366	0.090
Naïve Bayes	0.4075	0.0971
Naïve Bayes+Adaboost	**0.4112**	0.0992
SVM	0.39713	**0.10085**

Bảng 10: Kết quả so sánh ROUGE giữa các mô hình

Nhận xét:

Độ chính xác của hệ thống theo độ đo Rouge-1 và Rouge-2 lần lượt là 0.39713 và 0.10085. So với các hướng tiếp cận khác trên cùng một bộ dữ liệu cho thấy hướng tiếp cận của em hứa hẹn trong việc giải quyết bài toán tóm tắt văn bản.

CHƯƠNG V: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

1.1. Mục tiêu đã hoàn thành

- Nắm vững nền tảng Python
- Hiểu sâu hơn về các phương pháp xử lý ngôn ngữ tự nhiên.
- Các đặc trưng trong xử lý ngôn ngữ tự nhiên và thuật toán tính các đặc trưng.
- Thuật toán, mô hình SVM
- Các thư viện và công cụ hỗ trợ xử lý ngôn ngữ tự nhiên.
- Các độ đo ROUGE

1.2. Đóng góp của đồ án

Đồ án đưa ra một cái nhìn tổng quan về kĩ thuật: Học máy có giám sát SVM

Áp dụng công nghệ này để giải quyết bài toán. Đồ án cũng đã chứng minh được hiệu quả của phương pháp này với bài toán tóm tắt văn bản bằng tiếng Anh.

1.3. Hạn chế còn tồn tại

Mặc dù về cơ bản các mục tiêu của đồ án đã được hoàn thành. Tuy nhiên vẫn còn một số hạn chế như sau:

- Bước tiền xử lý chưa đồng bộ các từ đồng nghĩa
- Văn bản tóm tắt chưa xử lý các câu có độ tương đương lớn.
- Các từ không quan trọng, từ dư thừa trong văn bản tóm tắt chưa được loại bỏ.
- Bộ dữ liệu huấn luyện còn chưa lớn.

2. Hướng phát triển trong tương lai

Vì thời gian đồ án còn hạn chế nên em xin đề xuất hướng phát triển vẫn chưa kịp thực hiện cho ứng dụng như sau:

- Tìm hiểu và sử dụng Word2Vec trong bước tiền xử lý văn bản: đồng nhất các từ đồng nghĩa.
- Trong văn bản tóm tắt, những câu được cho là tương đồng về ngữ nghĩa chỉ giữ lại một.
- Loại bỏ các từ thừa, từ không quan trọng trong câu.
- Phân tích xử lý bộ dữ liệu tiếng Việt của Báo Mới
- Tạo giao diện ứng dụng cho chương trình thử nghiệm: thân thiện với người dùng.
- Kết hợp với các hướng tiếp cận tóm tắt trích rút khác, tạo cơ chế vote câu quan trọng.

TÀI LIỆU THAM KHẢO

[1] – Extractive Summarization Using Supervised and Semi-supervised Learning của Kam-Fai Wong, Mingli Wu

[2] – B. Liu. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, 2006.

[3] – The Document Understanding Conference (DUC) : https://duc.nist.gov

[4] – LIBSVM Chih-Chung Chang and Chih-Jen Lin:
https://www.csie.ntu.edu.tw/~cjlin/libsvm

[5] – Slides of talk by Chin-Yew Lin:
http://users.dsic.upv.es/~dpinto/duc/RougeLin.pdf

[6] – Saggion H., D. Radev, S. Teufel, and W. Lam. Meta-Evaluation of Summaries in a Cross-Lingual Environment Using Content-Based Metrics. In *Proceedings of COLING-2002*, Taipei, Taiwan, 2002

[7] – Papineni, K., S. Roukos, T. Ward, and W.-J Zhu. BLEU : a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of* ACL, Philadelphia, USA, 2002

[8] – Đồ án tốt nghiệp đại học *kĩ thuật phân tích ma trận và deep learning trong tóm tắt văn bản-Dương Việt Hùng 12-2017*

THƯ VIỆN MÃ NGUỒN MỞ

[1] – Nltk-examples https://github.com/sujitpal/nltk-examples
[2] – LIBSVM Chih-Chung Chang and Chih-Jen Lin:
https://github.com/cjlin1/libsvm
[3] – NLTK https://github.com/nltk/nltk
[4] – TextBlob: https://github.com/sloria/TextBlob