|  |
| --- |
| TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT  **KHOA CÔNG NGHỆ THÔNG TIN**            **BÁO CÁO ĐỒ ÁN**  **KHOA HỌC MÁY TÍNH**    Tên đề tài:  **Học máy tóm tắt văn bản tiếng việt cho**  **trang web**      ***Sinh viên thực hiện*: Trần Văn An**  ***Lớp*: Khoa học máy tính ứng dụng K64A**  **Hà Nội – 2021** |

**Mục Lục**

[**CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI** 3](#_Toc122874737)

[**Lý do chọn đề tài** 3](#_Toc122874738)

[**Mục tiêu đề tài** 3](#_Toc122874739)

[***Mục tiêu đối với chương trình tóm tắt văn bản*** 3](#_Toc122874740)

[***Mục tiêu đối với sinh viên thực hiện*** 3](#_Toc122874741)

[**Nội dung nghiên cứu** 4](#_Toc122874742)

[**Phạm vi nghiên cứu** 4](#_Toc122874743)

[***Bố cục của đồ án*** 4](#_Toc122874744)

[**CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ** 4](#_Toc122874745)

[**Học máy là gì?** 4](#_Toc122874746)

[**Phân Nhánh học máy?** 5](#_Toc122874747)

[***Bài toán phân lớp trong học có giám sát.*** 6](#_Toc122874748)

[***Bài toán hồi quy trong học có giám sát.*** 6](#_Toc122874749)

[***Học không giám sát.*** 6](#_Toc122874750)

[***Học bán giám sát.*** 7](#_Toc122874751)

[***Học củng cố.*** 8](#_Toc122874752)

[**NLP là gì?** 8](#_Toc122874753)

[***NLP hoạt động như thế nào.*** 8](#_Toc122874754)

[***Có thể đạt được gì qua NLP:*** 9](#_Toc122874755)

[***Tại sao NLP lại cần thiết.*** 9](#_Toc122874756)

[***Các ứng dụng của NLP:*** 9](#_Toc122874757)

[**Tóm tắt văn bản là gì? Có mấy cách tóm tắt văn bản?** 9](#_Toc122874758)

[**2.** **Tiền xử lý dữ liệu dành cho tóm tắt văn bản** 10](#_Toc122874759)

[**Word embedding là gì?** 10](#_Toc122874760)

[**Các bộ thư viện sử dụng để tóm tắt văn bản?** 10](#_Toc122874761)

[***Giới thiệu về thư viện học máy:*** 10](#_Toc122874762)

[***Giới hiệu về một vài thư viện dùng trong tóm tắt văn bản:*** 11](#_Toc122874763)

[**Cách để xử lý văn bản đối với văn bản tiếng việt** 12](#_Toc122874764)

[**CHƯƠNG III: MÔ TẢ HỆ THỐNG** 12](#_Toc122874765)

[**Cách chương trình tóm tắt văn bản:** 12](#_Toc122874766)

[**Thuật toán, kỹ thuật được sử dụng trong chương trình:** 14](#_Toc122874767)

[***Emberding*** 14](#_Toc122874768)

[***Word2Vec*** 19](#_Toc122874769)

[***TFIDF*** 27](#_Toc122874770)

[**CHƯƠNG IV: THIẾT KẾ HỆ THỐNG** 29](#_Toc122874771)

[**Thu thập dữ liệu:** 29](#_Toc122874772)

[**Chuẩn bị dữ liệu:** 30](#_Toc122874773)

[***B1: load datasets đã đc chuyển đổi sang csv*** 30](#_Toc122874774)

[***B2: Xóa các thuộc tính ko liên quan đến chương trình:*** 30](#_Toc122874775)

[***B3: Kiểm tra missing cho các thuộc tính đã tách*** 31](#_Toc122874776)

[***B4: Tách cách dữ liệu missing*** 31](#_Toc122874777)

[**Tạo modul package huấn luyện cho chương trình:** 31](#_Toc122874778)

[***3.1 Tạo word embedding:*** 31](#_Toc122874779)

[***3.2 Tạo các modul và package:*** 32](#_Toc122874780)

[**Huấn luyện mô hình:** 38](#_Toc122874781)

[**CHƯƠNG V: KẾT QUẢ THỰC NGHIỆM** 39](#_Toc122874782)

[**Thực hiện tóm tắt thử 1 đoạn văn bản:** 39](#_Toc122874783)

[**Kết quả trả về:** 39](#_Toc122874784)

[**KẾT LUẬN** 39](#_Toc122874785)

[**TÀI LIỆU THAM KHẢO** 39](#_Toc122874786)

[**Tài liệu trong trường :** 39](#_Toc122874787)

[**Tài liệu trên internet:** 39](#_Toc122874788)

**Mục lục ảnh**

[Hình 2. 1 Sơ đồ phân nhánh của học máy. 9](file:///C:\Users\An%20Tran\Downloads\Đồ%20Án.docx#_Toc122875686)

[Hình 2. 2 Thuật toán phân cụm 11](file:///C:\Users\An%20Tran\Downloads\Đồ%20Án.docx#_Toc122875687)

[Chương 3 - h. 1 Biểu diễn cách chương trình tóm tắt văn bản 18](#_Toc122875641)

[Chương\_3 - h. 2 Biểu diễn các word dưới dạng one-hot vector và embedding 21](#_Toc122875642)

[Chương\_3 - h. 3 Biểu diễn các vector embedding trong không gian 21](#_Toc122875643)

[Chương\_3 - h. 4 Ma trận embedding(ma trận trọng số trong một neural nhân tạo) 22](#_Toc122875644)

[Chương\_3 - h. 5 Biểu diễn khoảng cách Euclidean 23](#_Toc122875645)

[Chương\_3 - h. 6 Biểu diễn khoảng cách cosin 24](#_Toc122875646)

[Chương\_3 - h. 7 Ví dụ one-hot vector 26](#_Toc122875647)

[Chương\_3 - h. 8 Ví dụ w2v từ đích - từ ngữ cảnh 27](#_Toc122875648)

[Chương\_3 - h. 9 Kiên trúc thần kinh của Skip gram với kích thước bối cảnh C=3 29](#_Toc122875649)

[Chương\_3 - h. 10 Minh họa skip-gram dới dạng neural 31](#_Toc122875650)

[Chương\_3 - h. 11 Kiến trúc thần kinh CBOW với C=3 34](#_Toc122875651)

[Chương\_3 - h. 12 Biểu diễn CBOW dưới dạng neural 35](#_Toc122875652)

[Chương 4 - h. 1 Cấu tạo tập dữ liệu 37](#_Toc122875753)

[Chương 4 - h. 2 Biểu diễn cách lấy dữ liệu ra khỏi các folder 38](#_Toc122875754)

[Chương 4 - h. 3 Biểu diễn cách load từng folder dữ liệu chuyển đổi và lưu lại thành df.csv 38](#_Toc122875755)

[Chương 4 - h. 4 Load dữ liệu đã đc chuyển đổi sang csv 39](#_Toc122875756)

[Chương 4 - h. 5 Lấy nội dung văn bản và tóm tắt 39](#_Toc122875757)

[Chương 4 - h. 6 dữ liệu missing 39](#_Toc122875758)

[Chương 4 - h. 7Chia tập dữ liệu 40](#_Toc122875759)

[Chương 4 - h. 8 hàm xử lý văn bản 40](#_Toc122875760)

[Chương 4 - h. 9 huấn luyện mô hình skip gram 41](#_Toc122875761)

[Chương 4 - h. 10 Lưu mô hình 41](#_Toc122875762)

[Chương 4 - h. 11 Biểu diễn pakage và modul tổng thể 41](#_Toc122875763)

[Chương 4 - h. 12 Biểu diễn cách các odul xử lý tóm tắt văn bản 42](#_Toc122875764)

[Chương 4 - h. 13 Cách tóm tắt văn bản bằng trọng số 42](#_Toc122875765)

[Chương 4 - h. 14 Cách tóm tắt văn bản tần suất 43](#_Toc122875766)

[Chương 4 - h. 15 Tóm tắt văn bản với countvec – cosin 43](#_Toc122875767)

[Chương 4 - h. 16 Tóm tắt với lexrank 44](#_Toc122875768)

[Chương 4 - h. 17 Tóm tắt với tfidf và cosin 44](#_Toc122875769)

[Chương 4 - h. 18 Mô hình kmeans+countvec+cosin 45](#_Toc122875770)

[Chương 4 - h. 19 mô hình kmeans + tfidf 45](#_Toc122875771)

[Chương 4 - h. 20 Mô hình kmeans + skipgram 46](#_Toc122875772)

[Chương 4 - h. 21Cách xử lý văn bản 47](#_Toc122875773)

[Chương 4 - h. 22 Chương trình chính \_\_init\_\_ package 48](#_Toc122875774)

[Chương 4 - h. 23 Cách gọi chương trình + huấn luyện + lấy kết quả 49](#_Toc122875775)

[Chương 4 - h. 24 vd tóm tắt 1 văn bản 50](#_Toc122875776)

[Chương 4 - h. 25 Độ chính xác của mô hình 50](#_Toc122875777)

# **CHƯƠNG I: TỔNG QUAN VỀ ĐỀ TÀI**

## **Lý do chọn đề tài**

Những năm gần đây là khoảng thời gian Internet có sự phát triển mạnh mẽ tại Việt Nam. Cách đây khoảng 7,8 năm thì Internet có lẽ vẫn còn khá xa lạ thì hiện nay hiện tượng người dùng truy cập và sử dụng các thông tin trên Internet đã trở nên rất phổ biến. Internet hầu như lưu trữ tất cả những thông tin mà người dùng tải lên nhưng người dùng lại không thể nào đọc hết tất cả nội dung rùi mới chọn ra nội dung mình muốn nên việc tóm tắt văn bản trở nên cần thiết hơn bao giờ hết.

Việc nghiên cứu các bài toán khai thác văn bản đã được nghiên cứu từ lâu nhưng hầu như các mô hình tóm tắt văn bản hiện tại vẫn chưa thể nào hoàn hảo với văn bản Tiếng Việt.

Tóm tắt văn bản là một bài toán rất được quan tâm trong lĩnh vực xử lý ngôn ngữ tự nhiên. Sản phẩm của quá trình tóm tắt văn bản thường ngắn hơn văn bản ban đầu, súc tích, nhưng vẫn giữ nguyên ý chính. Mặc cho sự quan trọng của việc tóm tắt văn bản như vậy nhưng đáng ngạc nhiên là cho đến hiện tại số lượng các nghiên cứu giải quyết bài toán tóm tắt đối với văn bản tiếng Việt lại rất ít. Bởi vậy em đã mạnh dạn chọn Tóm Tắt Văn Bản Tiếng Việt làm nội dung nghiên cứu cho đồ án. Qua việc nghiên cứu các phương pháp, kỹ thuật có thể ứng dụng để giải quyết bài toán, em hy vọng có thể tiếp cận với nhiều kỹ thuật tiên tiến và mở rộng vốn kiến thức của mình.

## **Mục tiêu đề tài**

### ***Mục tiêu đối với chương trình tóm tắt văn bản***

1. **Mục tiêu hướng đến chương trình**

* Chương trình có thể tóm tắt được ý chính của văn bản một cách chính xác
* Rút trích được từ trong cả đoạn văn bản dài và ngắn
* Sử dụng dành cho văn bản tiếng Việt

**b. Mục tiêu hướng đến người sử dụng chương trình**

* Người sử dụng có thể đọc nhanh qua các ý chính của văn bản để dưa ra lựa chọn(vd: chương trình được sử dụng trong trang web tin tức)

**c. Mục tiêu phụ của chương trình**

* Các thuật toán sử dụng với tốc độ khả quan có thể chấp nhận
* Sử dụng nhiều thuật toán để đưa ra kết quả cuối cùng tăng độ chính xác

### ***Mục tiêu đối với sinh viên thực hiện***

* Nắm được vững các kiến thức cơ bản để tạo ra 1 chương trình học máy như: thu thập dữ liệu, xử lý dữ liệu, chọn thuật toán, tạo ra chương trình,…
* Biết chọn các thuật toán dành cho các bài toán cụ thể như bài toán hiện tại tóm tắt văn bản NLP(Neuro-Linguistic Programming: Lập trình ngôn ngữ nơ ron)
* Biết cách tư duy phân tích bài toán và giải quyết vấn đề theo logic.

## **Nội dung nghiên cứu**

Để hoàn thành chương trình tóm tắt văn bản Tiếng Việt thì phải thực hiện tìm hiểu :

* Cách xử lý dữ liệu đầu vào
* Cách xử lý văn bản với tiếng Việt
* Các mô hình được sử dụng trong NLP- tóm tắt văn bản
* Cách hiện thực hóa chương trình tạo modul và package
* Cách huấn luyện mô hình và kiểm tra độ chính xác

## **Phạm vi nghiên cứu**

Đề tài được nghiên cứu trong phạm vi tóm tắt văn bản dành cho tiếng Việt được sử dụng trong các trang web cụ thể như trang web tin tức hoặc trang tìm kiếm. Ví dụ điển hình như việc tóm tắt qua hàng hóa trong trang thương mại điện tử hoặc các trang thông tin báo chí,…

### ***Bố cục của đồ án***

**Mở đầu**

Chương 1: Tổng quan về hệ thống

Chương 2: Cơ sở lý thuyết và công nghệ

Chương 3: Mô tả và phân thích hệ thống tiếp cận hướng đối tượng

Chương 4: Thiết kế hệ thống hướng đối tượng

Chương 5: Kết quả thực nghiệm .

**Kết luận**

Tài liệu tham khảo

# **CHƯƠNG II: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ**

## **Học máy là gì?**

Đầu tiên lý do để tìm hiểu học máy vì tóm tắt văn bản (NLP) là 1 phần của học máy. Tóm tắt văn bản được sử dụng từ các model của học máy và học sâu để có thể hiểu được văn bản và tóm tắt lại văn bản đó. Hiện tại chương trình sẽ sử dụng các mô hình học máy để thực hiện tóm tắt văn bản và chưa đề cập đến khái niệm, cách sử dụng học sâu(deep learning) ở đây.

Ngày nay, học máy (Machine learning) đang là một bước tiến mới, được ứng dụng rộng rãi vào đời sống của con người và mọi mặt của xã hội. Điển hình là ứng dụng học máy trong nhận diện khuôn mặt, ô tô tự lái, gợi ý mua hàng trên các nền tảng thương mại điện tử và các ứng dụng trí tuệ nhân tạo tiên tiến khác. Học máy tập trung vào việc tạo ra các hệ thống, máy móc có khả năng tự học mà không cần lập trình cụ thể, học dựa trên các kinh nghiệm mà các hệ thống thu nhập được trong suốt quá trinh hoạt động, điều này mô phỏng gần như giống với hoạt động trí tuệ của con người.

Học máy là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Nói đơn giản, học máy là một lĩnh vực nhỏ của khoa học máy tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

## **Phân Nhánh học máy?**

Diagram

Description automatically generatedĐầu tiên học máy là một phần trong AI, học máy là một thuật ngữ với khái niệm rất rộng và được phân chia thành các nhánh nhỏ, mỗi nhánh thuộc về các lĩnh vực khác nhau ví dụ như NLP cũng là một phần trong nó.

Hình 2. 1 Sơ đồ phân nhánh của học máy.

* 1. **Học có giám sát.**

Học có giám sát (supervised learning) là một kỹ thuật của ngành học máy nhằm mục đích xây dựng một hàm f từ dữ tập dữ liệu huấn luyện (Training data). Dữ liệu huấn luyện bao gồm các cặp đối tượng đầu vào và đầu ra mong muốn. Đầu ra của hàm f có thể là một giá trị liên tục hoặc có thể là dự đoán một nhãn phân lớp cho một đối tượng đầu vào.

Diagram

Description automatically generated

Bản chất của học có giám sát là máy sẽ học dựa theo những dữ liệu được gán nhãn để đưa ra dự đoán kết quả cho các tình huống chưa từng gặp phải theo một cách hợp lý.

Học có giám sát được chia thành 2 loại là :

* Phân lớp (Classification)
* Hồi Quy (Regression)

### ***Bài toán phân lớp trong học có giám sát.***

Một bài toán được gọi là phân lớp nếu các nhãn của dữ liệu đầu vào được chia thành một số hữu hạn lớp (miền giá trị là rời rạc). Chẳng hạn như tính năng xác định xem một email có phải là spam hay không của Gmail,…

Trong bài toán phân lớp được chia thành 2 loại: phân lớp nhị phân và phân lớp đa lớp;

* Phân lớp nhị phân: các giá trị của thuộc tính phụ thuộc sẽ khoong vượt quá 2 giá trị . vd như: bài toán dự đoán bệnh kết quả trả về có – không ,…
* Phân lớp đa lớp: các giá trị của thuộc tính phụ thuộc có từ 2 giá trị trở lên. Vd: phân loại các loại hoa,..

### ***Bài toán hồi quy trong học có giám sát.***

Bài toán hồi quy được chia làm 2 loại:

* Hồi quy tuyến tính

Hồi quy là một hình thức trong học máy với mục đích tạo ra một mô hình dự đoán đầu ra là các giá trị liên tục, chẳng hạn như giá cả, số lượng, khối lượng hoặc các giá trị vô hướng khác.

Hồi quy cho thấy được mối quan hệ giữa các biến trong dữ liệu cần quan sát và biến mà chúng ta cần dự đoán. Trong mô hình hồi quy tuyến tính, một tập dữ liệu được dán nhãn, tập dữ liệu sau đó được chia làm hai phần:

* Tập dữ liệu huấn luyện (Training dataset): Dữ liệu sử dụng trong mô hình nhằm xác định một hàm số biểu thị mối quan hệ giữa các biến đặc trưng và các nhãn.
* Tập dữ liệu kiểm tra (test dataset): Dữ liệu dùng để đánh giá hàm số đã tìm được trong mô hình sau khi sử dụng tập dữ liệu huấn luyện ở bước trước đó bằng cách so sánh giá trị dự đoán và giá trị thực tế.
* Hồi quy phi tuyến tính

Trong hồi quy phi tuyến tính, hàm số f(x) không thể xem là tuyến tính nếu như các tham số không phải là tuyến tính. Đồ thị của hàm số phi tuyến tính là một đường cong.

Hàm số phi tuyến tính sẽ như sau:

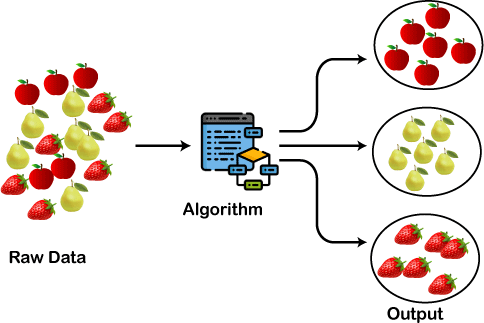
### ***Học không giám sát.***

Khác với học có giám sát, dữ liệu trong tập dữ liệu không có đầu ra, thay vì thuật toán gắn nhãn thì thuật toán sẽ phân cụm dữ liệu.

Mục tiêu của thuật toán này là tái cấu trúc dữ liệu đầu vào thành các đặc trưng mới và nhóm các đối tượng thành các đặc trưng tương tự.

Trong học không giám sát được chia thành 2 loại: phân cụm và Luật kết hợp.

1. **Phân cụm trong học không giám sát.**

Phân cụm là một hình thức trong học máy không giám sát, trong đó dữ liệu được phân nhóm thành các cụm nhỏ dựa vào sự liên quan giữa các dữ liệu trong mỗi nhóm. Gọi là không giám sát bởi bì nó không sử dụng các nhãn đã biết trước đó để huấn luyện mô hình, thay vào đó là các nhãn là cụm được gán cho các cụm, hoàn toàn dựa trên các đặc trưng của dữ liệu. Có rất nhiều thuật toán có thể phân cụm dữ liệu, thông dụng nhất có thể kể đến thuật toán K- Mrans. Một trong những lĩnh vực ứng dụng của phân cụm là nghiên cứu tiếp thị

Hình 2. 2 Thuật toán phân cụm

1. **Luật kết hợp trong học không giám sát.**

Là bài toán mà khi muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Ví dụ như khách hàng mua mặt hàng này sẽ mua thêm mặt hàng kia, khán giả xem phim này sẽ có xu hướng xem phim kia, dựa vào đó người ta có thể xây dựng được những hệ thống gợi ý khách hàng nhằm thúc đẩy nhu cầu mua sắm hoặc xem phim…

### ***Học bán giám sát.***

Học bán giám sát là một phương pháp được sử dụng để cho phép máy móc phân loại cả vật thể hữu hình và vật thể vô hình. Các đối tượng mà máy móc cần phân loại hoặc xác định có thể khác nhau như suy ra mô hình học tập của học sinh từ các video trong lớp học để rút ra các kết luận từ các nỗ lực đánh cắp dữ liệu trên máy chủ. Để tìm hiểu và suy luận về các đối tượng, máy móc được cung cấp thông tin nông cạn về các loại dữ liệu khác nhau dựa trên đó các máy cần học từ dữ liệu lớn, có cấu trúc và không cấu trúc mà chúng nhận được thường xuyên.

Đơn giản mà nói học bán giám sát là sự kết hợp giữa 2 phương pháp học có dám sát và học không dám sát. Dữ liệu được dán nhãn một phần và phần còn lại không được dán nhãn

### ***Học củng cố.***

Học củng cố hay học tăng cường là bài toán giúp cho hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất. Học tăng cường chủ yếu được áp dụng vào lý thuyết trò chơi. Một số thuật toán học tăng cường là Q-learning, Deep Q-Network (DQN), SARSA, Monte Carlo, …

Một ví dụ nổi tiếng về học củng cố chính là AlphaGo.Về cơ bản, AlphaGo bao gồm các thuật toán thuộc cả Học máy có giám sát và Học củng cố. Trong phần Học máy có giám sát, dữ liệu từ các ván cờ do con người chơi với nhau được đưa vào để huấn luyện. Tuy nhiên, mục đích cuối cùng của AlphaGo không phải là chơi như con người mà phải thậm chí thẳng cả con người. Chính vì thế, sau khi học xong các ván cờ của con người, AlphaGo tự chơi với chính nó để tìm ra các nước đi mới tối ưu hơn.

## **NLP là gì?**

Lập trình ngôn ngữ tư duy (Neuro Linguistic Programming – NLP) có thể được mô tả là “cách tiếp cận hướng tới sự kết nối giữa quá trình tư duy (Neuro), ngôn ngữ (Linguistic) và các mô thức hành vi đã có thông qua trải nghiệm (Programming)”. Lập trình ngôn ngữ tư duy là một phương pháp hiệu quả để nhanh chóng hiểu được những động cơ và động lực cho mỗi hành động của con người. Vì vậy, NLP ngày càng được áp dụng nhiều trong tất cả các ngành nghề và môi trường khác nhau.

Nguyên tắc đầu tiên trong thuyết NLP là đã là con người thì phải có khả năng yêu và tận hưởng cuộc sống. Khả năng này có thể bị ảnh hưởng bởi một sự kiện nào đó. Sự kiện này có thể là một sự kiện lớn, hay những sự kiện dường như rất nhỏ và không quan trọng – có thể có tác động rất lớn tới những hành động, suy nghĩ, biểu hiện và hành vi của chúng ta. Con người thường sử dụng lập trình ngôn ngữ tư duy để đạt được các mục tiêu cụ thể trong cuộc sống, chẳng hạn như phát triển cá nhân, kỹ năng giao tiếp và nâng tầm ảnh hưởng bản thân.

### ***NLP hoạt động như thế nào.***

Thông qua việc sử dụng NLP, bạn sẽ làm việc trên các giác quan, suy nghĩ và cảm xúc của mình để nhận thức được những yếu tố tác động này. Trọng tâm không phải là việc bạn nhớ được gì mà là cách trí nhớ của bạn hoạt động như thế nào, và ảnh hưởng của những kí ức, sự kiện lên thói quen hằng ngày của bạn. NLP sẽ dạy bạn nhìn nhận các trải nghiệm và sự kiện theo một cách hoàn toàn khác. Từ đó mà bạn có thể xử lý vấn đề theo một cách khác, và phát triển được một phương pháp tiếp cận có tính hệ thống cho mọi vấn đề. Áp dụng mô hình này, bạn sẽ biết cách lắng nghe và giao tiếp khác đi, và sẽ sớm nhận ra những thay đổi và kết quả tích cực mình có thể đạt được. NLP không tập trung vào vấn đề bạn gặp phải mà tập trung vào cách giải quyết vấn đề đó.

NLP là một quá trình tương tác giữa người thực hiện và khách hàng, bao gồm nhiều giai đoạn như lập báo cáo, thu thập thông tin về một vấn đề và đặt ra các mục tiêu mong muốn.

Cả quá trình này sẽ đi liền với các phản ứng phi ngôn ngữ từ phía khách hàng. Bước đầu tiên là thiết lập và duy trì sự thống nhất giữa người thực hiện và khách hàng, thông qua hoạt động kích thích để có được những phản hồi ngôn ngữ và phi ngôn ngữ từ khách hàng

### ***Có thể đạt được gì qua NLP:***

NLP là một tập hợp các kỹ thuật, điểm khởi đầu và các suy nghĩ thường được sử dụng để phát triển bản thân. NLP được sử dụng trong các buổi đào tạo để nâng cao năng lực của các nhà quản lý và nhân viên thông qua các buổi hội thảo, các khóa đào tạo nhóm hoặc qua các mối quan hệ cá nhân giữa khách hàng và người huấn luyện.

### ***Tại sao NLP lại cần thiết.***

Thông qua việc am hiểu NLP, bạn sẽ:

Học được cách điểu khiển trạng thái cảm xúc của bản thân

Phát triển được kỹ năng thuyết phục trong giao tiếp

Vượt qua được những trở ngại trong công việc và cuộc sống

Giúp được những người xung quanh mình thành công

Và làm được nhiều điều thú vị khác trong cuộc sống

### ***Các ứng dụng của NLP:***

NLP (Lập Trình Ngôn Ngữ Nơ ron) đã được ứng dụng rộng rãi với kết quả vô cùng khả quan trong rất nhiều lĩnh vực:

- Kinh doanh

- Huấn luyện

- Bán hàng

- Lãnh đạo

- Tâm lý học

- Thể thao

- Y tế

- Thương thuyết

- Giáo dục

- Và nhiều lĩnh vực khác …

## **Tóm tắt văn bản là gì? Có mấy cách tóm tắt văn bản?**

Tóm tắt văn bản là quá trình trích rút những thông tin quan trọng nhất từ một văn bản để tạo ra phiên bản ngắn gọn, xúc tích mang đầy đủ lượng thông tin của văn bản gốc kèm theo đó là tính đúng đắn về ngữ pháp và chính tả. Bản tóm tắt phải giữ được những thông tin quan trọng của toàn bộ văn bản chính. Bên cạnh đó, bản tóm tắt cần phải có bố cục chặt chẽ có tính đến các thông số như độ dài câu, phong cách viết và cú pháp văn bản. Phụ thuộc vào số lượng các văn bản, kỹ thuật tóm tắt có thể chia làm hai lớp: đơn văn bản và đa văn bản.

            Tóm tắt đơn văn bản chỉ đơn giản là rút gọn một văn bản thành một sự trình bày ngắn gọn. Trong khi đó tóm tắt đa văn bản phải rút gọn một tập các văn bản thành một sự tóm tắt. Tóm tắt đa văn bản có thể xem như một sự mở rộng của tóm tắt đơn văn bản và thường dùng với thông tin chứa trong các cụm văn bản, để người dùng có thể hiểu được cụm văn bản đó. Tóm tắt đa văn bản phức tạp hơn tóm tắt đơn văn bản vì phải làm việc trên số lượng văn bản nhiều hơn.

Tóm tắt văn bản theo mô hình học máy hiện tại có 2 hướng tiếp cận :

* Extraction : Tóm tắt theo kiểu trích chọn. Phương pháp tóm tắt trích chọn là công việc chọn ra một tập con những từ đã có, những lời nói hoặc những câu của văn bản gốc để đưa vào khuôn mẫu tóm tắt.
* Abstraction: Tóm tắt kiểu trừ tượng máy sẽ tự sinh ra từ ngũ cho phù hợp(Cách này rất khó thực hiện và rất ít mô hình cho ra kết quả khả quan thời điểm hiện tại)

1. **Tiền xử lý dữ liệu dành cho tóm tắt văn bản.**

Tiền xử lý dữ liệu là bước đầu tiên trong quá trình xây dựng tóm tắt văn bản. Tiền xử lý dữ liệu thực chất là làm sạch dữ liệu trước khi đưa vào mô hình huấn luyện, làm sạch dữ liệu giúp cho tóm tắt chính xác và đạt hiệu quả cao hơn.

* Đầu tiên là phát hiện ranh giới câu trong đoạn văn bản, điều này giúp ích cho việc phân tách câu, vì mô hình tóm tắt sẽ tính toán, đánh giá từng câu để xem xét xem câu có liên quan đến bản tóm tắt hay không. Bộ công cụ NLTK 1được sử dụng cho việc tách câu.
* Tiếp đến là tách câu thành các từ để sử dụng chúng vào quá trình đào tạo bộ từ vựng tiếng Việt. Công cụ Underthesea 2được sử dụng cho việc tách từ.
* Bước cuối cùng trong quá trình tiền xử lý dữ liệu là loại bỏ các ký tự đặc biệt trong văn bản, cũng như là loại bỏ stopword. Các ký tự đặc biệt thường gặp là: !\*&^%$()-#@,.’;=-?/+{}[]. Danh sách các stopword trong tiếng Việt được sử dụng gồm 1.942 từ được đề xuất bởi Lê Văn Duyệt

## **Word embedding là gì?**

Word embedding, còn gọi là “nhúng từ”, là một kỹ thuật biểu diễn các từ trong văn bản, được giới thiệu lần đầu bởi Bengio và các cộng sự vào năm 2003. Hai phương pháp nổi tiếng được sử dụng để đào tạo word embedding là Word2vec và Glove 4 . Word2vec gồm 2 mô hình là CBOW (Continuous Bag of Word) dùng để dự đoán một từ dựa vào các từ lân cận và Skipgram dùng để dự đoán ngữ cảnh xung quanh từ. Trong phạm vi nghiên cứu, phương pháp Skipgrama sẽ được sử dụng kết hợp với hướng tiếp cận Extraction mục đích nhằm biến các câu thành vector mà vẫn giữ được ngữ nghĩa.

## **Các bộ thư viện sử dụng để tóm tắt văn bản?**

* Hiện tại có rất nhiều bộ thư viện có sẵn để tóm tắt văn bản nhưng hầu như không hỗ trợ tiếng Việt nên ở đây e sẽ kết hợp các thư viện cơ bản để xử lý dữ liệu như: numpy, pandas ,.. Và các thuật toán có sẵn nhữ: lexrank, textrank , …

### ***Giới thiệu về thư viện học máy:***

* ***Sklearn:***

Sklearn là một thư viện Học máy mã nguồn mở hữu ích và mạnh mẽ trong Python. Dự án của David Cournapeau bắt đầu vào năm 2007 với tư cách là một dự án của Google Summer of Code. Hiện tại, Sklearn đang được duy trì bởi một đội ngũ các tình nguyện viên. Sklearn cung cấp một sự lựa chọn các công cụ hiệu quả cho Học máy và mô hình thống kê, bao gồm phân loại, hồi quy, phân cụm và giảm chiều dữ liệu với giao diện nhất quán trong Python. Thư viện này phần lớn được viết bằng Python, được xây dựng dựa trên NumPy, SciPy và Matplotlib**.**

Trang chủ của Sklearn: <https://scikit-learn.org/stable/>

Trang chủ mã nguồn Github:<https://github.com/scikit-learn/scikit-learn>

* ***TensorFlow:***

TF là một nền tảng Học máy mã nguồn mở, được thiết kế bởi đội ngũ Google Brain và tổ chức nghiên cứu trí tuệ máy của Google nhằm triển khai các ứng dụng của Học máy và Học sâu theo cách đơn giản. Nó là kết hợp giữa Đại số tính toán của các kỹ thuật tối ưu hoá để dễ dàng tính toán các biểu thức toán học. TF có một hệ sinh thái toàn diện, linh hoạt bao gồm các công cụ, thư viện và tài nguyên cộng đồng cho phép các nhà nghiên cứu xây dựng và triển khai các ứng dụng Học máy. Đây cũng là một trong những thư viện máy học lâu đời nhất.

TF có tài liệu đầy đủ và bao gồm nhiều thư viện Học máy nên rất phổ biến hiện nay. Vì là một sản phẩm của Google, nên hiện tại, TF đang được ứng dụng rất nhiều trong các sản phẩm của Google như phân loại chữ viết tay, nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên

Trang chủ của TF: <https://www.tensorflow.org/>

Trang mã nguồn Github: <https://github.com/tensorflow/tensorflow>

### ***Giới hiệu về một vài thư viện dùng trong tóm tắt văn bản:***

* ***LexRank:***

LexRank là một cách tiếp cận không giám sát để tóm tắt văn bản dựa trên cách tính trung tâm dựa trên đồ thị của câu. Ý chính là các câu "giới thiệu" các câu tương tự khác cho người đọc. Do đó, nếu một câu rất giống với nhiều câu khác, nó có thể sẽ là một câu có tầm quan trọng lớn. Tầm quan trọng của câu này cũng xuất phát từ tầm quan trọng của các câu "giới thiệu" nó. Như vậy, để được xếp hạng cao và đặt trong một bản tóm tắt, một câu phải tương tự như nhiều câu lần lượt cũng tương tự như nhiều câu khác. Điều này có ý nghĩa trực quan và cho phép các thuật toán được áp dụng cho bất kỳ văn bản mới tùy ý nào.

Cách sử dụng: có sẵn trong pip python. Cài đặt với pip install python

* ***NLTK:***

Thư viện [NLTK](https://www.nltk.org/) - Natural Language Toolkit là một trong những thư viện open-source xử lí ngôn ngữ tự nhiên. Được viết bằng Python và với ưu điểm là dễ dàng sử dụng nên thư viện này ngày càng trở nên phổ biến và có được một cộng đồng lớn mạnh. Thư viện cung cấp hơn 50 kho dữ liệu văn bản khác nhau (corpora) và nhiều chức năng để xử lí dữ liệu văn bản để phục vụ cho nhiều mục đích khác nhau.

Cài thư viện NLTK với pip pip install nltk.

## **Cách để xử lý văn bản đối với văn bản tiếng việt**

Hiện tại có rất nhiều bộ thư viện hỗ trợ xử lý văn bản dành riêng cho tiếng Việt, chương trình tóm tắt văn bản chương trình này sẽ chỉ sử dụng Underthesea.

* **Underthesea**
* **Một bộ công cụ NLP tiếng Việt.** Underthesea là một bộ dữ liệu mô-đun Python mã nguồn mở và hướng dẫn hỗ trợ nghiên cứu và phát triển trong [Xử lý ngôn ngữ tự nhiên tiếng Việt](https://github.com/undertheseanlp/underthesea). Chúng tôi cung cấp API cực kỳ dễ dàng để nhanh chóng áp dụng các mô hình NLP được đào tạo sẵn cho văn bản tiếng Việt của bạn, chẳng hạn như phân đoạn từ, gắn thẻ một phần giọng nói (PoS), nhận dạng thực thể được đặt tên (NER), phân loại văn bản và phân tích cú pháp phụ thuộc.
* **Một thư viện Pytorch.** Underthesea được hỗ trợ bởi một trong những thư viện deep learning phổ biến nhất, [Pytorch](https://pytorch.org/), giúp bạn dễ dàng đào tạo các mô hình deep learning và thử nghiệm các phương pháp tiếp cận mới bằng cách sử dụng các mô-đun và lớp underthesea.
* **Một phần mềm mã nguồn mở.** Underthesea được xuất bản theo giấy phép [GNU General Public License v3.0.](https://github.com/undertheseanlp/underthesea/blob/master/LICENSE) Quyền của giấy phép copyleft mạnh mẽ này có điều kiện là tạo mã nguồn hoàn chỉnh có sẵn của các tác phẩm và sửa đổi được cấp phép, bao gồm các tác phẩm lớn hơn sử dụng tác phẩm được cấp phép, theo cùng một giấy phép.

# **CHƯƠNG III: MÔ TẢ HỆ THỐNG**

## **Cách chương trình tóm tắt văn bản:**

* Đầu tiên chương trình sẽ nhận văn bản được truyền vào qua người sử dụng hoặc qua bất kỳ luồng nào. Vd: Sử dụng cho trang web sẽ nhận api của trang web.
* Tiếp theo chương trình sẽ tiền xử lý dữ liệu như: tách từ, tách câu, remove stopword,…
* Sau đó chương trình sẽ chạy qua các modul để tính điểm cho từng câu.Mỗi modul là một mô hình tóm tắt văn bản và đưa ra những đoạn tóm tắt.
* Chương trình nhận các đoạn tóm tắt và thực hiện gộp lại các đoạn tóm tắt tính trọng số và đưa ra văn bản được ghép lại với trọng số cao nhất.

A picture containing table

Description automatically generated

Chương 3 - h. 1 Biểu diễn cách chương trình tóm tắt văn bản

Ví dụ :

Chương trình sẽ nhận văn bản:

|  |
| --- |
| Bằng Kiều đã làm thỏa mãn những người yêu mến giọng ca của anh bằng một đêm nhạc ấn tượng. Xuyên suốt liveshow, khán giả một lần nữa được thưởng thức lại các ca khúc hay nhất của anh từ những ngày đầu ca hát cho đến hôm nay. Hàng loạt những ca khúc hit của Bằng Kiều được chọn thể hiện Hát live liên tục hơn 20 ca khúc nhưng với chất giọng cao và truyền cảm, Bằng Kiều đã không làm người nghe thất vọng. Ba vị khách mời Trọng Tấn, Thu Minh và Hồ Quỳnh Hương là phần q Hồ Quỳnh Hương bay bổng với " chìm trong âm nhạc đích thực qua ca khúc O sole mio . uà đặc biệt mà Bằng Kiều dành cho khán giả. Bằng Kiều Dẫu có lỗi lầm" Kết hợp cùng Thu Minh . và Phần kết hợp gây ấn tượng nhất trong đêm có lẽ là tiết mục của Bằng Kiều và nam ca sĩ Trọng Tấn. Cả hai đã khiến khán giả có những giây phút đắm một ca khúc nhạc cổ điển, Trọng Tấn hát tiếng Anh còn Bằng Kiều thể hiện bằng tiếng Việt. Màn kết hợp tuyệt với đã giành được những tràng pháo tay không ngớt từ phía khán giả. Màn kết hợp lần đầu tiên giữa Trọng Tấn và Bằng Kiều đã để lại tiếng vang lớn Sự hòa hợp của hai giọng ca đã đem lại cho khán giả những giây phút thăng hoa cùng âm nhạc Clip Bằng Kiều và Trọng Tấn ngẫu hứng hát ca khúc "Bài ca xây dựng" Hai khách mời nữ của đêm nhạc . là Thu Minh và Hồ Quỳnh Hương cũng cống hiến cho khán giả những ca khúc hit quen thuộc của mình như Thu Minh với Yêu mình anh và Bay, Hồ Quỳnh Hương với Anh và Tình yêu mãi mãi. |

Sau đó văn bản sẽ được xử lý như lower,remove stopword và tách thành từng câu:

|  |
| --- |
| bằng kiều đã làm thỏa mãn những người yêu mến giọng ca của anh bằng một đêm nhạc ấn tượng |
| xuyên suốt liveshow, khán giả một lần nữa được thưởng thức lại các ca khúc hay nhất của anh từ những ngày đầu ca hát cho đến hôm nay |
| … |

Tiếp theo chương trình sẽ chạy qua các modul là những thuật toán để tính toán điểm cho từng câu và sắp xếp lại đưa các câu có điểm cao lên đầu và trả về đoạn tóm tắt:

|  |  |  |
| --- | --- | --- |
| STT Câu | Nội dung | Điểm |
| 1 | Hàng loạt những ca khúc hit của Bằng Kiều được chọn thể hiện Hát Điểm live liên tục hơn 20 ca khúc nhưng với chất giọng cao và truyền cảm, Bằng Kiều đã không làm người nghe thất vọng. | 0.988 |
| 2 | Màn kết hợp lần đầu tiên giữa Trọng Tấn và Bằng Kiều đã để lại tiếng vang lớn Sự hòa hợp của hai giọng ca đã đem lại cho khán giả những giây phút thăng hoa cùng âm nhạc Clip Bằng Kiều và Trọng Tấn ngẫu hứng hát ca khúc "Bài ca xây dựng" Hai khách mời nữ của đêm nhạc là Thu Minh và Hồ Quỳnh Hương cũng cống hiến cho khán giả những ca khúc hit quen thuộc của mình như Thu Minh với Yêu mình anh và Bay, Hồ Quỳnh Hương với Anh và Tình yêu mãi mãi . | 0,875 |
| 3 | … | … |

Sau khi nhận về các đoạn tóm tắt của những thuật toán khác nhau chương trình sẽ gộp lại và tính toán lại điểm cho những đoạn tóm tắt. Cuối cùng chương trình sẽ đưa ra đoạn tóm tắt là kết quả của tất cả các thuật toán gộp lại.

|  |
| --- |
| Hàng loạt những ca khúc hit của Bằng Kiều được chọn thể hiện Hát live liên tục hơn 20 ca khúc nhưng với chất giọng cao và truyền cảm, Bằng Kiều đã không làm người nghe thất vọng…. |

## **Thuật toán, kỹ thuật được sử dụng trong chương trình:**

### ***Emberding***

Embedding là một kỹ thuật đưa một vector có số chiều lớn, thường ở dạng thưa, về một vector có số chiều nhỏ, thường ở dạng dày đặc. Phương pháp này đặc biệt hữu ích với những đặc trưng hạng mục có số phần tử lớn ở đó phương pháp chủ yếu để biểu diễn mỗi giá trị thường là một vector dạng one-hot. Một cách lý tưởng, các giá trị có ý nghĩa tương tự nhau nằm gần nhau trong không gian embedding.

1. Biểu diễn toán học

Giả sử một từ điển nào đó chỉ có sáu giá trị (Hà Nội, Hải Phòng, Tp HCM, Bình Dương, Lào Cai, Sóc Trăng).  Thể hiện cách biểu diễn của các giá trị này trong không gian one-hot và không gian embedding hai chiều.

A screenshot of a computer

Description automatically generated with low confidence

Chương\_3 - h. 2 Biểu diễn các word dưới dạng one-hot vector và embedding

Ở đây, các giá trị trong không gian embedding được lấy ví dụ bằng tay với chiều thứ nhất thể hiện dân số và chiều thứ hai thể hiện vĩ độ đã chuẩn hóa của mỗi giá trị. Vị trí của mỗi vector embedding trong không gian hai chiều được minh hoạ trong hình. Trong không gian này, Hà Nội, Hải Phòng và Hà Giang gần nhau về vị trí địa lý. Nếu chúng ta có một bài toán nào đó mà dân số có thể là một đặc trưng tốt, ta chỉ cần co trục tung và giãn trục hoành là có thể mang những tỉnh thành có dân số giống nhau gần với nhau hơn.

A screenshot of a computer screen

Description automatically generated with medium confidence

Chương\_3 - h. 3 Biểu diễn các vector embedding trong không gian

Với một từ điển bất kỳ với ***N*** từ . Giả sử số chiều của không gian embedding là ***d***, ta có thể biểu diễn toàn bộ các embedding cho ***N*** từ này dưới dạng một ma trận  với hàng thứ ***i*** là biểu diễn embedding cho từ .

Nếu vector  là biểu diễn one-hot của từ , ta có ngay là biểu diễn của từ đó trong không gian embedding.

* Cách tạo embedding :

Cách biểu diễn các tỉnh thành trong ví dụ trên đây chỉ là một ví dụ minh họa khi chúng ta có thể định nghĩa các trục một cách cụ thể dựa vào kiến thức nhất định đã có về dữ liệu. Cách làm này không khả thi với những dữ liệu vô cùng nhiều chiều và không có những ý nghĩa từng trục rõ ràng như trên. Việc tìm ra ma trận **E** cần thông qua một quá trình “học” dựa trên mối quan hệ vốn có của dữ liệu.

Ta có thể thấy rằng ma trận **E** có thể được coi là một ma trận trọng số của một tầng tuyến tính trong một mạng neural nhân tạo như trong hình dưới

A picture containing diagram

Description automatically generated

Chương\_3 - h. 4 Ma trận embedding(ma trận trọng số trong một neural nhân tạo)

Như vậy, ma trận này cũng có thể được xây dựng bằng cách đặt nó vào một mạng neural với một hàm mất mát nào đó.

Embedding có thể được học trong cả một bài toán tổng thể hoặc học riêng rẽ khác trước khi đưa vào một bài toán cụ thể. Embedding thu được có thể được dùng như một đặc trưng nhiều chiều và có thể trong các mô hình không phải học sâu.

**Word2vec** là một trong những phương pháp tiên phong về việc xây dựng embedding cho các từ dựa trên một mạng học sâu. Các vector embedding này được học chỉ dựa trên thứ tự các từ trong câu của một bộ dữ liệu lớn mà không cần biết ý nghĩa cụ thể của từng câu hay mối quan hệ đặc biệt nào giữa chúng. Các vector embedding này có thể được dùng để tạo các biểu diễn cho một câu hay một văn bản để giải quyết các bài toán khác.

* Độ tương tự giữa hai embedding:

Quay lại với mục đích chính của việc tạo embedding là đưa các giá trị hạng mục về một không gian số sao cho embedding của những giá trị tương tự nằm *gần* nhau trong không gian. Vậy *khoảng cách* này thường được tính như thế nào.

Có ba phép đo thường được sử dụng để tính khoảng cách giữa hai embedding là :

* + - Khoảng cách Euclid
    - Tích vô hướng của hai vector
    - Độ tương tự cosine.
* Khoảng cách Euclid:

Chart, line chart

Description automatically generated

Chương\_3 - h. 5 Biểu diễn khoảng cách Euclidean

Công thức tính khoảng cách Euclid giữa hai vector embedding:

Khoảng cách này không âm và càng nhỏ thì hai vector embedding càng gần nhau. Ở đây, là độ lớn của vector .

Để giảm sự phức tạp khi khai căn, bình phương khoảng cách Euclid thường được sử dụng. Việc lấy bình phương không ảnh hưởng tới việc so sánh khoảng cách vì bình phương là một hàm đồng biến.

* Tích vô hướng của hai vector:

Công thức tính độ tương tự theo tích vô hướng (dot product) giữa hai vector embedding:

Tính vô hướng giữa hai vector càng cao thể hiện các embedding càng giống nhau. Giá trị này lớn nếu góc giữa hai vector nhỏ và các vector này có độ dài lớn.

* Tương tự Cosine:

A picture containing chart

Description automatically generated

Chương\_3 - h. 6 Biểu diễn khoảng cách cosin

Tương tự cosin cũng được sử dụng để đo độ tương tự giữa hai vector:

Góc giữa hai vector càng nhỏ thì độ tương tự cosin càng cao. Độ tương tự cosin nhỏ nhất bằng -1 nếu hai vector này trái dấu nhau.

Trong ba độ đo trên đây, tích vô hướng có công thức đơn giản nhất và thường được sử dụng trong các bài toán quy mô lớn. Tương tự cosine không quan tâm tới độ lớn của hai vector mà chỉ xét tới góc giữa chúng, phép đo này phù hợp với các bài toán yêu cầu tìm sự trái ngược giữa các giá trị hạng mục. Nếu các vector embedding có cùng độ dài, ba phép đo này có ý nghĩa như nhau.

* Tìm embedding gần nhất:

Embedding được dùng nhiều trong bài toán tìm kiếm các điểm trong cơ sở dữ liệu (item embeddings) gần nhất với một embedding truy vấn (query embedding) nào đó.

Giả sử  và lần lượt là ma trận embedding của các giá trị trong cơ sở dữ liệu và vector truy vấn.

Với khoảng cách Euclid , khoảng cách giữa ***q*** và một embedding trong ***E*** được tính bởi:

Chỉ số của embedding gần ***q*** được tính như sau:

Với độ tương thích ***tích vô hướng*** , chỉ số embedding gần ***q*** được tihs như sau:

Với độ tương tự ***cosine***, chỉ số embedding gần ***q*** được tính như sau:

### ***Word2Vec***

Như đã nói phía trên : **Word2vec** là một trong những phương pháp tiên phong về việc xây dựng embedding cho các từ dựa trên một mạng học sâu. Word2vec là một mô hình đơn giản và nổi tiếng giúp tạo ra các biểu diễn embedding của từ trong một không gian có số chiều thấp hơn nhiều lần so với số từ trong từ điển. Ý tưởng của word2vec đã được sử dụng trong nhiều bài toán với dữ liệu khác xa với dữ liệu ngôn ngữ.

Bình thường muốn thể hiện 1 từ chúng ta sẽ dùng one-hot vector:A picture containing text, electronics, surge suppressor

Description automatically generated

Chương\_3 - h. 7 Ví dụ one-hot vector

* Độ lớn vector sẽ bằng đúng số lượng từ vựng. Vấn đề ở đây là làm thế nào để thể hiện mối liên kết gữa các từ , độ tương đồng của các từ như thế nào. Word2vec chính là giải pháp cho vấn đề này.

Có thể hiểu đơn giản ý tưởng cơ bản của word2vec như sau:

* + Hai từ xuất hiện trong những văn cảnh giống nhau thường có ý nghĩa gần với nhau.
  + Ta có thể đoán được một từ nếu biết các từ xung quanh nó trong câu. Ví dụ, với câu “Hà Nội là … của Việt Nam” thì từ trong dấu ba chấm khả năng cao là “thủ đô”.

Trong ví dụ dưới đây, từ “thủ đô” đang được xét và được gọi là *target word* hay *từ đích*. Những từ xung quanh nó được gọi là *context words* hay *từ ngữ cảnh*. Với mỗi từ đích trong một câu của cơ sở dữ liệu, các từ ngữ cảnh được định nghĩa là các từ trong cùng câu có vị trí cách từ đích một khoảng không quá ***C/2*** với ***C*** là một số tự nhiên dương. Như vậy, với mỗi từ đích, ta sẽ có một bộ không quá ***C*** từ ngữ cảnh.

Xét ví dụ sau đây với câu tiếng Anh: “The quick brown fox jump over the lazy dog” với ***C=4***

Graphical user interface, application, Teams

Description automatically generated

Chương\_3 - h. 8 Ví dụ w2v từ đích - từ ngữ cảnh

Khi “the” là từ đích, ta có cặp dữ liệu huấn luyện là (the, quick) và (the, brown). Khi “brown” là từ đích, ta có cặp dữ liệu huấn luyện là (brown, the), (brown, quick), (brown, fox) và (brown, jumps).

Word2vec định nghĩa hai embedding vector cùng chiều cho mỗi từ ***w*** trong từ điển. Khi nó là một từ đích, embedding vector của nó là ***u***; khi nó là một từ ngữ cảnh, embedding của nó là ***v***. Sở dĩ ta cần hai embedding khác nhau vì ý nghĩa của từ đó khi nó là từ đích và từ ngữ cảnh là khác nhau. Tương ứng với đó, ta có hai ma trận embedding ***U*** và ***V*** cho các từ đích và các từ ngữ cảnh.

Có hai cách khác nhau xây dựng mô hình word2vec:

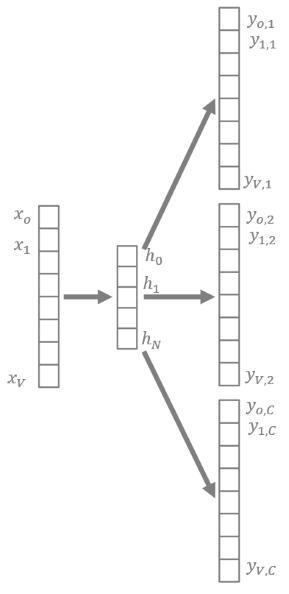
* Skip-gram: Dự đoán những từ ngữ cảnh nếu biết trước từ đích.
* CBOW (Continuous Bag of Words): Dựa vào những từ ngữ cảnh để dự đoán từ đích.

Mỗi cách có những ưu nhược điểm khác nhau và áp dụng với những loại dữ liệu khác nhau.

* Skip-gram làm việc tốt với dữ liệu nhỏ, nó có khả năng biểu diễn tốt những từ có tần suất thấp. Việc này hợp lý vì ta xây dựng được nhiều mẫu huấn luyện xung quanh từ có tần suất thấp này.
* CBOW phù hợp với các bộ dữ liệu lớn khi mà số mẫu huấn luyện được tạo ra từ mỗi ngữ cảnh (chỉ là một) ít hơn nhiều so với Skip-gram (tỉ lệ với kích thước cửa số ngữ cảnh). CBOW biểu diễn tốt hơn các từ xảy ra thường xuyên.

1. Skip-gram:

Hãy hiểu 1 cách đơn giản trước sau đó chúng ta sẽ đi đến cách mà skip gram xây dựng mô hình word2vec.



Chương\_3 - h. 9 Kiên trúc thần kinh của Skip gram với kích thước bối cảnh C=3

Với kiến trúc thần kinh đã biểu diễn chúng ta có thể hiểu đơn giản rằng với mô hình skip gram thì Input sẽ là từ cần tìm mối quan hệ và mô hình sẽ trả về output là các từ có quan hệ gần nhất với từ đó

* Xây dựng hàm mất mát:

Mọi tính toán trong mục này được xây dựng xung quanh một từ ngữ cảnh. Hàm mất mát tổng cộng sẽ là tổng của hàm mất mát tại mỗi từ ngữ cảnh. Việc tối ưu hàm mất mát có thể được thực hiện thông qua [Gradient Descent](https://machinelearningcoban.com/2017/01/12/gradientdescent/)(là phương pháp để tìm các điểm global minium trong hàm mất mát) trên từng từ ngữ cảnh hoặc một batch các từ ngữ cảnh.

Xét ví dụ bên trên với từ đích là “fox” và các từ ngữ cảnh là “quick”, “brown”, “jumps” và “over”. Việc dự đoán xác suất xảy ra các từ ngữ cảnh khi biết từ đích được mô hình hóa bởi:

**P("quick","brown","jumps","over"|"fox")**

Ta có thể giả sử rằng sự xuất hiện của một từ ngữ cảnh khi biết từ đích *độc lập* với các từ ngữ cảnh khác để xấp xỉ xác suất trên đây bởi:

**P("quick"|"fox")P("brown"|"fox")P("jumps"|"fox")P("over"|"fox")**

Giả sử từ đích có chỉ số***t*** trong từ điển ***V*** và tập hợp các chỉ số của các từ ngữ cảnh tương ứng là . Số lượng phần tử của  dao động từ ***C/2*** (nếu  đứng đầu hoặc cuối câu) tới ***C*** (nếu  đứng ở giữa câu và có đủ  ***C/2*** từ ngữ cảnh ở mỗi phía).

Từ dữ liệu đã có, ta cần một mô hình sao cho xác suất dưới đây càng lớn càng tốt với mỗi từ ngữ cảnh **:**

Để tránh các sai số tính toán khi nhân các số nhỏ hơn 1 với nhau, bài toán tối ưu này thường được đưa về bài toán tối thiểu đối số của ***log*** (thường được gọi là *negative log loss*):

Xác suất có điều kiện  được định nghĩa bởi:

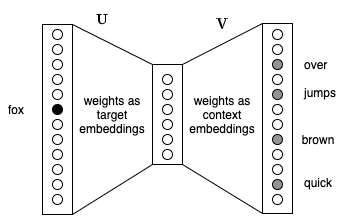
Với ***N*** là số phần tử của từ điển ***V***. Ở đây thể hiện mỗi quan hệ giữa từ đích  và từ ngữ cảnh . Biểu thức này càng cao thì xác suất thu được càng lớn. Tích vô hướngcũng thể hiện sự tương tự giữa hai vector.

Biểu thức này rất giống với công thức [Softmax](https://machinelearningcoban.com/2017/02/17/softmax/). Việc định nghĩa xác suất như biểu thức xác xuất có điều kiện ở trên đảm bảo rằng:

Tóm lại, hàm mất mát ứng với từ đích  theo ***U***,***V*** được cho bởi

***L***

* Biểu diễn dưới mạng neural



Chương\_3 - h. 10 Minh họa skip-gram dới dạng neural

Ở đây,  chính là kết quả của phép nhân vector one-hot tương ứng với  với ma trận trọng số U, vì vậy đây chính là giá trị đầu ra của của tầng ẩn ở giữa khi xét từ đích **.** Tiếp theo, đầu ra của tầng ẩn không hàm kích hoạt này được nhân trực tiếp với ma trận trọng số đầu ra V để được , đây chính là giá trị vector logit trước khi đi vào hàm kích hoạt softmax như trong biểu thức:

Có thể thấy rằng skip-gram word2vec là một mạng neural vô cùng đơn giản với chỉ một tầng ẩn không có hàm kích hoạt. Kiến trúc đơn giản này giúp word2vec hoạt động tốt ngay cả khi số lượng từ trong từ điển là cực lớn (có thể lên tới nhiều triệu từ). Lưu ý rằng kích thước đầu vào và đầu ra của mạng word2vec này bằng với số lượng từ trong từ điển.

* Tối ưu hàm mất mát

Việc tối ưu hai ma trận trọng số ***U*** và ***V*** được thực hiện thông qua các thuật toán [Gradient Descent](https://machinelearningcoban.com/2017/01/12/gradientdescent/). Các thuật toán tối ưu dạng này yêu cầu tính gradient cho từng ma trận.

Xét riêng số hạng

Đạo hàm theo :

Như vậy, mặc dù gradient này rất đẹp, chúng ta vẫn cần phải tính toán các xác suất . Mỗi xác suất này phụ thuộc toàn bộ ma trận trọng số ***V*** và vector  Như vậy ta cần cập nhập tổng cộng ***N\*d+d*** trọng số. Đây rõ ràng là một con số rất lớn với ***N*** lớn.

* Xấp xỉ hàm mất mát và lấy mẫu âm

Để tránh việc cập nhật rất nhiều tham số này trong một lượt, một phương pháp xấp xỉ được đề xuất giúp cải thiện tốc độ tính toán đáng kể. Mỗi xác suất  được mô hình bởi một hàm [sigmoid](https://machinelearningcoban.com/2017/01/27/logisticregression/#sigmoid-function) thay vì hàm softmax:

Lưu ý rằng tổng các xác suất không còn bằng 1 nữa. Tuy nhiên, nó vẫn mang ý nghĩa về xác suất có mặt của riêng từ ngữ cảnh  đi cùng với từ đích .

Lúc này, việc tính toán  chỉ còn phụ thuộc vào vector  và vector  (thay vì cả ma trận ***V***). Tương ứng với số hạng này, sẽ chỉ có **2d** trọng số cần được cập nhật cho mỗi cặp Số lượng trọng số này *không* phụ thuộc vào kích thước từ điển, khiến cho cách mô hình này có thể hoạt động tốt với ***N*** rất lớn.

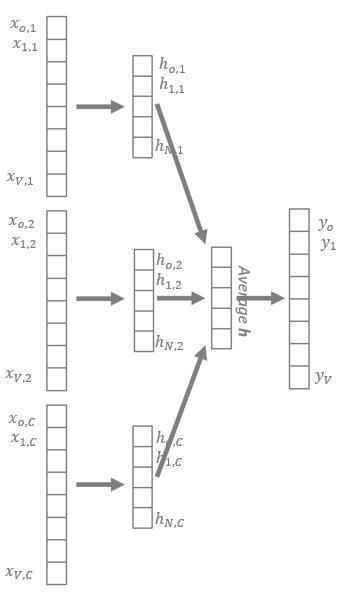
Có một vấn đề lớn với cách mô hình hóa này!

Vì không có sự ràng buộc giữa các xác suất  khi cố gắng tối đa hóa mỗi xác suất sẽ dẫn đến việc nghiệm thu được thỏa mãn mọi  đều cao. Điều này sẽ đạt được khi xấp xỉ 0. Chỉ cần toàn bộ các phần tử của ***U*** và ***V*** tiến tới dương vô cùng là thỏa mãn. Việc xấp xỉ này bây giờ trở nên tầm thường và vô nghĩa. Để tránh vấn đề này, ta cần thêm đưa thêm các ràng buộc sao cho tồn tại các xác suất khác cần được tối thiểu hóa khi xét tới từ đích .

Bản chất của bài toán tối ưu ban đầu là xây dựng mô hình sao cho với mỗi từ đích, xác suất của một từ ngữ cảnh xảy ra là cao trong khi xác suất của *toàn bộ* các từ ngoài ngữ cảnh đó là thấp – việc này được thể hiện trong hàm softmax. Để hạn chế tính toán, trong phương pháp này ta chỉ lấy mẫu ngẫu nhiên một vài từ ngoài ngữ cảnh đó để tối ưu. Các từ trong ngữ cảnh được gọi là “từ dương”, các từ ngoài ngữ cảnh được gọi là “từ âm”; vì vậy phương pháp này còn có tên gọi khác là “lấy mẫu âm” (*negative sampling*).

Khi đó, với mỗi từ đích, ta có một bộ các từ ngữ cảnh với nhãn là 1 và 0 tương ứng với các từ ngữ cảnh ban đầu (gọi là *ngữ cảnh dương*) và các từ *ngữ cảnh âm* được lấy mẫu từ ngoài tập ngữ cảnh dương đó. Với các từ ngữ cảnh dương,   tương tự với [hàm mất mát trong hồi quy logistic](https://machinelearningcoban.com/2017/01/27/logisticregression/#-ham-mat-mat-va-phuong-phap-toi-uu) với nhãn bằng 1. Tương tự, ta có thể dùng như là hàm mất mát cho các từ ngữ cảnh âm với nhãn bằng 0.

1. Continuous Bag of Words(CBOW):



Chương\_3 - h. 11 Kiến trúc thần kinh CBOW với C=3

Có thể nói rằng CBOW ngược lại so với skip gram , Input của nó sẽ là context và output là từ gần nhất với context đó.

Tiếp tục xét ví dụ trên Continous bag of Words đi tìm xác suất xảy ra từ đích khi biết các từ ngữ cảnh xung quanh. Ta cần mô hình hóa dữ liệu sao cho xác suất sau đây đạt giá trị lớn:

**P("fox"|"quick","brown","jumps","over")**

Vì có nhiều từ ngữ cảnh trong điều kiện, chúng thường được đơn giản hóa bằng cách lấy một từ “trung bình” làm đại diện.

với  là trung bình cộng của các từ trong ngữ cảnh của từ đích . Embedding của từ trung bình này là trung bình của embedding các từ ngữ cảnh. Xác xuất này cũng được định nghĩa tương tự như trong Skip-gram:

Biểu diễn mạng neural cho CBOW được thể hiện như sau:

A picture containing text, controller, remote, game

Description automatically generated

Chương\_3 - h. 12 Biểu diễn CBOW dưới dạng neural

Lưu ý rằng giá trị tại tầng ẩn là trung bình cộng của các embedding của các từ ngữ cảnh.

Kỹ thuật tối ưu likelihood này cũng tương tự như trong Skip-gram và phương pháp lấy mẫu âm với các từ đích cũng có thể được sử dụng một cách tương tự.

### ***TFIDF***

**TF-IDF** (Term Frequency – Inverse Document Frequency) là 1 kĩ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu. Một vài biến thể của tf-idf thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng.  Tf-idf cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân loại văn bản.

1. **TF là gì?**

**TF**: Term Frequency(Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản( tổng số từ trong một văn bản).

Trong đó:

* tf(t, d): tần suất xuất hiện của từ t trong văn bản d
* f(t, d): Số lần xuất hiện của từ t trong văn bản d
* max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

1. **IDF là gì?**

IDF: Inverse Document Frequency(Nghịch đảo tần suất của văn bản), giúp đánh giá tầm quan trọng của một từ . Khi tính toán TF , tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng  một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.

Trong đó:

* idf(t, D): giá trị idf của từ t trong tập văn bản
* |D|: Tổng số văn bản trong tập D
* |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

Cơ số logarit trong công thức này không thay đổi giá trị idf của 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). Việc sử dụng logarit nhằm giúp giá trị tf-idf của một từ nhỏ hơn, do chúng ta có công thức tính tf-idf của một từ trong 1 văn bản là tích của tf và idf của từ đó.

Cụ thể, chúng ta có công thức tính tf-idf hoàn chỉnh như sau: **tfidf(t, d, D) = tf(t, d) x idf(t, D)**

Khi đó:

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).

# **CHƯƠNG IV: THIẾT KẾ HỆ THỐNG**

## **Thu thập dữ liệu:**

Hiện tại cũng đã có datasets dành cho tóm tắt đa văn bản tiếng Việt hiện tại e sẽ sử dụng datasets của tác giả Nghiêm Quốc Minh

Link: [CLC-HCMUS/ViMs-Dataset (github.com)](https://github.com/CLC-HCMUS/ViMs-Dataset)

Sau khi tải về tập dữ liệu sẽ có dạng :

Diagram

Description automatically generated

Chương 4 - h. 1 Cấu tạo tập dữ liệu

Việc cần thiết bây giờ là phải chuyển và lấy hết các dữ liệu Info summarize ra tạo thành 1 dataframe hoàn chỉnh nhằm cho việc xử lý dữ liệu và huấn luyện mô hình.

Text

Description automatically generated

Chương 4 - h. 2 Biểu diễn cách lấy dữ liệu ra khỏi các folder

Text

Description automatically generated

Chương 4 - h. 3 Biểu diễn cách load từng folder dữ liệu chuyển đổi và lưu lại thành df.csv

## **Chuẩn bị dữ liệu:**

Với việc thu thập dữ liệu giờ đây chúng ta đã có datasets hoàn chỉnh bây giờ cúng ta phải đi tiền xử lý dữ liệu:

### ***B1: load datasets đã đc chuyển đổi sang csv***

**Text

Description automatically generated**

Chương 4 - h. 4 Load dữ liệu đã đc chuyển đổi sang csv

### ***B2: Xóa các thuộc tính ko liên quan đến chương trình:***

Ở đây Content chính là nội dung văn bản Cluster là văn bản tóm tắt do biên tập viên viết. Chúng ta phải xóa các cột không cần thiết cho chương trình.



Chương 4 - h. 5 Lấy nội dung văn bản và tóm tắt

### ***B3: Kiểm tra missing cho các thuộc tính đã tách***



Chương 4 - h. 6 dữ liệu missing

### ***B4: Tách cách dữ liệu missing***

Tách dữ liệu missing thành tập test và còn lại sẽ là train. Sau khi dugf cho tập train chúng ta sẽ chuyển sang dự đoán luôn cho tập test.

Text

Description automatically generated

Chương 4 - h. 7Chia tập dữ liệu

Vì là chương trình tóm tắt văn bản nên chỉ cần xử lý missing mà thôi. Vì ở đây tất cả đều là văn bản nên sẽ ko có dữ liệu nhiễu.

## **Tạo modul package huấn luyện cho chương trình:**

Như đã nói ở trên chương trình này sẽ sử dụng nhiều cách để tóm tắt văn bản rùi sẽ gộp lại lấy ra kết quả cuối cùng.

### ***3.1 Tạo word embedding:***

Việc này là cần thiết để chuyển hóa các văn bản thành vector vô hướng. Điều đó sẽ giúp cho máy hiểu và tính toán theo công thức đã nói ở chương 3.

B1: xử lý các dữ liệu để đưa vào chương trình học máy:

Text

Description automatically generated

Chương 4 - h. 8 hàm xử lý văn bản

B2: Huấn luyện mô hình skip gram



Chương 4 - h. 9 huấn luyện mô hình skip gram

Việc chúng ta đơn giản là gọi Skipgram còn đâu pakge đã làm hết cho chúng ta việc tính toán những công thức phái trên.

B3: Lưu lại mô hình



Chương 4 - h. 10 Lưu mô hình

Để cho bộ mô hình chính xác hơn thì tập dữ liệu phải rất lớn. Ở đây e sẽ sử dụng luôn mô hình đã được đào tạo sẵn với bộ dữ liệu cực lớn có sẵn. Link : [Kyubyong/wordvectors: Pre-trained word vectors of 30+ languages (github.com)](https://github.com/Kyubyong/wordvectors)

### ***3.2 Tạo các modul và package:***

Ở đây e sẽ tạo 1 package cho mô hình dự đoán tổng thể của e

Diagram

Description automatically generated

Chương 4 - h. 11 Biểu diễn pakage và modul tổng thể

Với mỗi modul là 1 cách để tóm tắt văn bản và đều theo mô hình:

Diagram

Description automatically generated

Chương 4 - h. 12 Biểu diễn cách các odul xử lý tóm tắt văn bản

Tránh cho việc dài dòng e sẽ cmt ngay cạnh code để dễ hiểu cho logic và cách tính toán trong code:

* + - Mô hình trọng số - modul trọng số

Text

Description automatically generated

Chương 4 - h. 13 Cách tóm tắt văn bản bằng trọng số

* + - Mô hình frenquency tần suất – modul tần suất

Text

Description automatically generated

Chương 4 - h. 14 Cách tóm tắt văn bản tần suất

* + - Mô hình count vector và cosin – modul countvec-cosin

Text

Description automatically generated

Chương 4 - h. 15 Tóm tắt văn bản với countvec – cosin

* + - Lexrank – modul lexrank

Text

Description automatically generated

Chương 4 - h. 16 Tóm tắt với lexrank

* + - Tfidf và cosin – modul tfidf-cosin

Text

Description automatically generated

Chương 4 - h. 17 Tóm tắt với tfidf và cosin

* + - Mô hình Machine learning kmeans- modul kmeans

Text

Description automatically generated

Chương 4 - h. 18 Mô hình kmeans+countvec+cosin

Text

Description automatically generated

Chương 4 - h. 19 mô hình kmeans + tfidf

Text

Description automatically generated

Chương 4 - h. 20 Mô hình kmeans + skipgram

Bộ xử lý của package chính :

* + - Xử lý văn bản :

Text

Description automatically generated

Chương 4 - h. 21Cách xử lý văn bản

* + - Chương trình chính:

Text

Description automatically generated

Chương 4 - h. 22 Chương trình chính \_\_init\_\_ package

## **Huấn luyện mô hình:**

**Sau khi tạo thành công packge việc đơn giản bây giờ của chúng ta là gọi nó ra và huấn luyện mô hình:**

****

Chương 4 - h. 23 Cách gọi chương trình + huấn luyện + lấy kết quả

* Chương trình sẽ trả về 1 list danh sách các tóm tắt của từng văn bản trong data frame

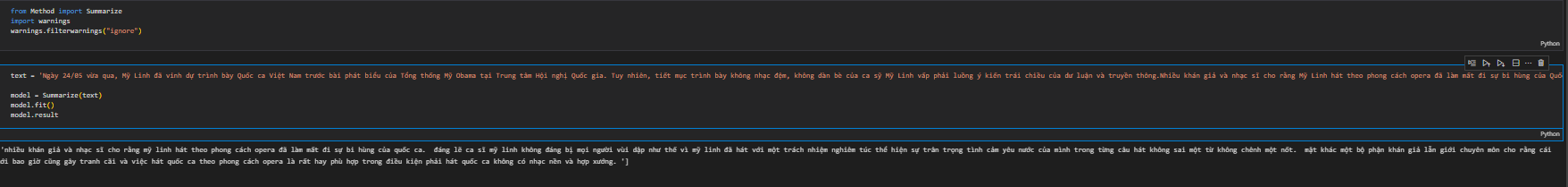
# **CHƯƠNG V: KẾT QUẢ THỰC NGHIỆM**

## **Thực hiện tóm tắt thử 1 đoạn văn bản:**

|  |
| --- |
| Ngày 24/05 vừa qua, Mỹ Linh đã vinh dự trình bày Quốc ca Việt Nam trước bài phát biểu của Tổng thống Mỹ Obama tại Trung tâm Hội nghị Quốc gia. Tuy nhiên, tiết mục trình bày không nhạc đệm, không dàn bè của ca sỹ Mỹ Linh vấp phải luồng ý kiến trái chiều của dư luận và truyền thông.Nhiều khán giả và nhạc sĩ cho rằng Mỹ Linh hát theo phong cách opera đã làm mất đi sự bi hùng của Quốc ca. Mặt khác, một bộ phận khán giả lẫn giới chuyên môn cho rằng ‘cái mới bao giờ cũng gây tranh cãi’ và việc hát quốc ca theo phong cách opera là rất hay, phù hợp trong điều kiện phải hát Quốc ca không có nhạc nền và hợp xướng.Giữa tâm bão dư luận, Mỹ Linh đã thẳng thắn lên tiếng trước những lời chê bai về việc phần trình diễn Quốc ca của mình: "Đã qua rồi thời chiến tranh bom đạn nên tôi hát quốc ca với thông điệp của thời bình, với tinh thần hàn gắn và chia sẻ yêu thương chứ không hừng hực kháng chiến như ngày nào". Nữ ca sĩ cũng thừa nhận, cô đã gặp "tai nạn nghề nghiệp" tại phần trình diễn này bởi lên cô bắt tone hơi thấp. Mỹ Linh cảm thấy rất tiếc vì có thể làm tốt hơn nếu có cơ hội hát lại.Cách đây ít phút, Diva Hồng Nhung đã đăng tải một dòng trạng thái dài lên tiếng bảo vệ cho Mỹ Linh – người đồng nghiệp đồng thời cũng là người bạn thân thiết của cô. Theo cô Bống, câu chuyện này đã bị thổi phồng quá xa trong những tranh cãi gay gắt mang tính tiêu cực.Diva Hồng Nhung cho rằng việc chọn Mỹ Linh hát Quốc ca trước bài phát biểu của Tổng thống Obama ‘là lựa chọn tối ưu’. Vì việc đứng hát trực tiếp, không nhạc đệm, không hợp xướng trước Tổng thống Obama – người đàn ông quyền lực nhất thế giới, trước hàng trăm ống kính máy quay, được phát trực tiếp không chỉ trong nước mà còn ở quốc tế đòi hỏi một bản lĩnh sân khấu thật sự vững vàng. Nên ‘Nếu không phải là một ca sĩ đẳng cấp cỡ Mỹ Linh thì... khả năng "cấm khẩu", không hát nên lời là hoàn toàn có thể xảy ra.’ – Hồng Nhung nhận xét.Hồng Nhung cho biết chị rất buồn vì ‘chuyện phê bình, chê bai gay gắt vượt ra quá xa trong khung văn hóa ứng xử cho phép của một số đông dành cho Mỹ Linh, đã trở nên không công bằng, rồi đến mức ác, cố ý vùi dập’. Đáng lẽ ca sĩ Mỹ Linh không đáng bị mọi người vùi dập như thế vì ‘Mỹ Linh đã hát với một trách nhiệm nghiêm túc, thể hiện sự trân trọng, tình cảm yêu nước của mình trong từng câu hát, không sai một từ, không chênh một nốt’.Kết lại tâm thư dài, cô Bống Hồng Nhung không quên dành lời động viên cho người đồng nghiệp, cũng là người bạn thân thiết của mình: ‘Tôi hãnh diện vì Mỹ Linh, một tài năng đủ lớn để sống vượt lên mọi vỏ bọc bên ngoài, trung thành với giá trị nhân văn mà mình tin tưởng, mặc những thị phi’.Trước Hồng Nhung, không ít nhạc sĩ như nhạc sĩ Hồ Hoài Anh, nhạc sĩ Thanh Phương, nhạc sĩ Trí Minh, nhạc sĩ Ngọc Châu, ca sĩ Thái Thùy Linh... đã lên tiếng ủng hộ và bảo vệ màn trình diễn Quốc ca phá cách mới mẻ của Mỹ Linh. |

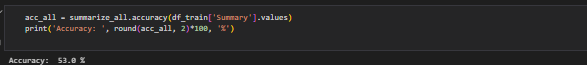
## **Kết quả trả về:**

|  |
| --- |
| nhiều khán giả và nhạc sĩ cho rằng mỹ linh hát theo phong cách opera đã làm mất đi sự bi hùng của quốc ca. đáng lẽ ca sĩ mỹ linh không đáng bị mọi người vùi dập như thế vì mỹ linh đã hát với một trách nhiệm nghiêm túc thể hiện sự trân trọng tình cảm yêu nước của mình trong từng câu hát không sai một từ không chênh một nốt. mặt khác một bộ phận khán giả lẫn giới chuyên môn cho rằng cái mới bao giờ cũng gây tranh cãi và việc hát quốc ca theo phong cách opera là rất hay phù hợp trong điều kiện phải hát quốc ca không có nhạc nền và hợp xướng. |

****

Chương 4 - h. 24 vd tóm tắt 1 văn bản

So sánh độ chính xác trên datasets đã tạo :



Chương 4 - h. 25 Độ chính xác của mô hình

Độ chính xác của mô hình không quá cao vì đây là phương pháp extraction nên hiệu quả có lẽ ko cao khi so sánh với tập dữ liệu do các chuyên gia tự tóm tắt lại. Nhưng 50% là con số khá cao hầu hết đều là các câu đúng trọng điểm. Và 1 phần cũng do cách xử lý tính toán điểm còn nhiều lỗi do tự thực hiện.

* Có thể đưa vào trang web tóm tắt tin tức hoặc nhưng phần mềm trích dẫn văn bản.

# **KẾT LUẬN**

Chương trình tóm tắt văn bản cho tiếng việt là 1 chương trình thiết thực nó không chỉ có thể áp dụng vào các trang web tin tức mà còn có thể ap dụng vào nhiều công việc khác nhau. Cách tiếp cận extraction là cách tiếp cận phổ thông chỉ tóm tắt lại văn bản có sẵn . Việc tiếp cận theo hướng Abstraction có lẽ sẽ tối ưu và dễ hiểu hơn máy sẽ tự sinh ra từ ngữ và tóm tắt lại như cách con người tóm tắt. Hiện tại rất ít bài tóm theo cách này và độ khó của hưởng tiếp cận Abstraction khá là cao yêu cầu kinh nghiệm và vốn kiến thức rất lớn về cả toán học và học máy.

Qua đồ án đã cho e biết cách tạo ra một chương trình học máy tóm tắt văn bản một cách cơ bản. Chương trình tóm tắt văn bản này vẫn còn rất nhiều thiếu sót với các công thức cơ bản.

# **TÀI LIỆU THAM KHẢO**

## **Tài liệu trong trường :**

1. Bài giảng Kỹ nghệ tri thức và học máy. Ts. Đặng Văn Nam, Trường Đại học Mỏ - Địa chất.
2. Bài giảng Xử lý ngôn ngữ. Thầy Nguyễn thế lộc, Trường Đại Học Mỏ Địa Chất
3. Bài giảng khai phá dữ liệu. Cô Nguyễn Thị Phương Bắc,Trường Đại học Mỏ Địa Chất

## **Tài liệu trên internet:**

* + - [TF-IDF là gì? Code demo thuật toán TF-IDF với dữ liệu tiếng Việt (luyencode.net)](https://blog.luyencode.net/tf-idf-la-gi/)
    - [Text Summarization In NLP (topcoder.com)](https://www.topcoder.com/thrive/articles/text-summarization-in-nlp)
    - [5 Powerful Text Summarization Techniques in Python. (turing.com)](https://www.turing.com/kb/5-powerful-text-summarization-techniques-in-python)
    - [Khoa học dữ liệu (phamdinhkhanh.github.io)](https://phamdinhkhanh.github.io/2019/04/29/ModelWord2Vec.html)
    - [LexRank method for Text Summarization (opengenus.org)](https://iq.opengenus.org/lexrank-text-summarization/)
    - [word2vec  |  TensorFlow Core](https://www.tensorflow.org/tutorials/text/word2vec#setup)
    - [Word nhúng: Word2Vec giải thích (helpex.vn)](https://helpex.vn/article/word-nhung-word2vec-giai-thich-5c664097ae03f6012876544a)