

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM  
KHOA CÔNG NGHỆ THÔNG TIN



TRẦN THÀNH QUANG: 19133047  
CAO ANH VĂN: 19133067

**Đề Tài:**

**TÌM HIỂU VỀ KỸ THUẬT EMBEDDING CHO  
XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**TIỂU LUẬN CHUYÊN NGÀNH**

**GIÁO VIÊN HƯỚNG DẪN**

**Th.S QUÁCH ĐÌNH HOÀNG**

TP.HCM, ngày Tháng năm 2022

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**

---



**TRẦN THÀNH QUANG: 19133047**

**CAO ANH VĂN: 19133067**

**Đề Tài:**

**TÌM HIỂU VỀ KỸ THUẬT EMBEDDING CHO  
XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**TIỂU LUẬN CHUYÊN NGÀNH**

**GIÁO VIÊN HƯỚNG DẪN**

**Th.S QUÁCH ĐÌNH HOÀNG**

**TP.HCM, ngày Tháng năm 2022**

## PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên 1: Trần Thành Quang MSSV: 19133047

Họ và tên sinh viên 2: Cao Anh Văn MSSV: 19133067

Ngành: Kỹ thuật dữ liệu

Tên đề tài: Tìm hiểu kỹ thuật embedding cho xử lý ngôn ngữ tự nhiên

Họ và tên giáo viên hướng dẫn: Th.S Quách Đình Hoàng

**NHẬN XÉT:**

1. Về nội dung và đề tài khối lượng thực hiện:

.....

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

.....

3. Khuyết điểm:

.....

.....

.....

.....

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

*Tp.Hồ Chí Minh, ngày...tháng...năm 2022*

Giáo viên hướng dẫn

*Ký & ghi rõ họ tên*

## LỜI CẢM ƠN

Trong quá trình nghiên cứu đề tài, nhóm đã được giảng viên hướng dẫn đã luôn hỗ trợ và góp ý các sai sót của nhóm, vì thế chúng tôi xin được bày tỏ lòng biết ơn sâu sắc đến thầy giáo hướng dẫn đề tài của nhóm là Ths.Quách Đình Hoàng.

Đầu tiên, chúng tôi xin gửi lời cảm ơn sâu sắc nhất đến Ban giám hiệu trường Đại học Sư phạm Kỹ Thuật Thành phố Hồ Chí Minh đã xây dựng cơ sở vật chất và một môi trường học tập hiện đại, chất lượng phục vụ nhóm trong quá trình hoàn thiện đề tài.

Đồng thời, chúng tôi xin gửi lời cảm ơn đến Ban chủ nhiệm khoa Công nghệ Thông tin và các Thầy Cô khoa Công nghệ Thông tin - Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo môi trường học tập và làm việc hiệu quả. Các thầy cô đã nhiệt tình giảng dạy để chúng tôi thực hiện tốt đề tài này.

Một lần nữa, chúng tôi xin gửi lời cảm ơn chân thành nhất đến Thầy Quách Đình Hoàng là giáo viên hướng dẫn tiểu luận chuyên ngành của nhóm, đã hướng dẫn, quan tâm, góp ý và luôn đồng hành cùng chúng tôi trong suốt các giai đoạn của đề tài.

Trong quá trình hoàn thiện đề tài, nhóm sẽ không thể tránh khỏi các sai sót và hạn chế nhất định. Kính mong nhận được những phản hồi, đóng góp ý kiến từ Quý Thầy Cô, để nhóm có thể nhận ra được lỗi sai, tiếp thu thêm các kiến thức và hoàn thiện dự án.

*Xin chân thành cảm ơn*

## KẾ HOẠCH THỰC HIỆN

Tuần	Thời gian	Nội dung công việc	Ghi chú
1, 2	22/08 – 04/09	Lựa chọn và xác định đề tài tiểu luận chuyên ngành.	
3, 4	05/09 – 18/09	Tìm hiểu sơ lược về đề tài.	
5, 6	19/09 – 02/10	Tìm hiểu về xử lý ngôn ngữ tự nhiên.	
7,8	03/10 – 16/10	Tìm hiểu word embedding cho xử lý ngôn ngữ tự nhiên.	
9, 10	17/10 – 30/10	Tìm hiểu sâu hơn về các kỹ thuật trong word embedding (word2vec, TF-IDF).	
11, 12	31/10 – 13/11	Tìm kiếm demo cho dự án.	
13, 14	14/11 – 27/11	Triển khai demo tóm tắt văn bản.	
15, 16	28/11 – 11/12	Tìm hiểu độ đo và hoàn thiện demo.	
17, 18	12/12 – 25/12	Hoàn thiện báo cáo	

## MỤC LỤC

DANH SÁCH HÌNH ẢNH.....	8
CHƯƠNG 1. MỞ ĐẦU .....	10
1.1. Tính cấp thiết của đề tài [1] .....	10
1.2. Mục tiêu và nhiệm vụ nghiên cứu.....	11
1.3. Cách tiếp cận và phương pháp nghiên cứu .....	11
1.4. Kết quả dự kiến đạt được .....	13
CHƯƠNG 2: NỘI DUNG.....	14
2.1. Cơ sở lý thuyết .....	14
2.1.1. Tổng quan về xử lý ngôn ngữ tự nhiên .....	14
2.1.1.1. Định nghĩa xử lý ngôn ngữ tự nhiên .....	14
2.1.1.2. Tầm quan trọng của xử lý ngôn ngữ tự nhiên.....	14
2.1.1.3. Sự phát triển của xử lý ngôn ngữ tự nhiên [2].....	15
2.1.2. Tổng quan về Embedding trong xử lý ngôn ngữ tự nhiên .....	18
2.1.2.1 Embedding [3] .....	18
2.1.2.2 Word embedding.....	19
2.2. Xây dựng ứng dụng.....	27
2.2.1. Tổng quan về tóm tắt văn bản.....	27
2.2.1.1. Khái niệm tóm tắt văn bản.....	27
2.2.1.2. Tầm quan trọng của tóm tắt văn bản .....	27
2.2.2. Thuật toán.....	28
2.2.2.1. Tổng quan về thuật toán K-means .....	28
2.2.3 Độ đo.....	30
2.2.3.1 Độ đo Rouge [4].....	30
2.2.3.2 Độ đo bert score [5] .....	32
CHƯƠNG 3. THỰC NGHIỆM .....	35
3.1 Bài toán .....	35
3.2 Dữ liệu [6] .....	35
3.3 Phương pháp và kết quả.....	39
3.3.1.Phương pháp tạo model .....	39

3.3.2. Quy trình tạo ra kết quả từ model đã xây dựng .....	39
3.3.3 . Kết quả .....	42
CHƯƠNG 4: KẾT LUẬN .....	52
4.1. Kết quả đạt được .....	52
4.1.1. Ý nghĩa khoa học .....	52
4.1.2. Ý nghĩa thực tiễn.....	52
4.2. Hạn chế.....	52
4.3. Hướng phát triển .....	53
TÀI LIỆU THAM KHẢO.....	54

## DANH SÁCH HÌNH ẢNH

Hình 2.1.2.1.1 Ví dụ minh họa One-hot encoding.[3].....	18
Hình 2.1.2.2.1 Mô phỏng các hoạt động của CBOW và Skip-gram.[3]	24
Hình 2.1.2.2.2 Mô phỏng câu input với CBOW.[9] .....	25
Hình 2.1.2.2.3 Mô phỏng câu input với phương pháp Skip-gram .[10]..	26
Hình 2.2.2.1.1 Minh họa phân cụm sử dụng thuật toán K-means.[8]....	28
Hình 2.2.3.2.1 Hình ảnh minh họa bert score. [5] .....	34
Hình 3.2.1 Các tập dữ liệu trong folder thứ nhất .....	36
Hình 3.2.2 Các file dữ liệu txt trong một chủ đề. ....	36
Hình 3.2.3 Nội dung một file txt trong chủ đề khoa học. ....	37
Hình 3.2.4 Các tập dữ liệu trong folder thứ 2. ....	37
Hình 3.2.5 Các file dữ liệu txt trong một chủ đề. ....	38
Hình 3.2.6 Nội dung một file txt trong chủ đề âm nhạc. ....	38
Hình 3.3.1.1 Code xây dựng model. ....	39
Hình 3.3.2.1 Mô hình luồng xử lý sử dụng model đã xây dựng.....	40
Hình 3.3.3.1 Nội dung của đoạn văn bản được tóm tắt.....	43
Hình 3.3.3.2 Các câu được tách từ đoạn nội dung ban đầu.....	43
Hình 3.3.3.3 Kết quả phân cụm các câu. ....	44
Hình 3.3.3.4 Kết quả tóm tắt.....	45
Hình 3.3.3.5 Hình ảnh máy ảo trên aws .....	46
Hình 3.3.3.6 Kết quả thực thi API.....	47
Hình 3.3.3.7 Số câu tóm tắt và độ đo Precision tốt nhất.....	48
Hình 3.3.3.8 Web page tóm tắt văn bản sử dụng chế độ Default.....	48
Hình 3.3.3.9 Kết quả tóm tắt văn bản sử dụng web page với chế độ default.....	49
Hình 3.3.3.10 Kết quả độ đo tóm tắt văn bản sử dụng web page với chế độ default .....	49
Hình 3.3.3.11 Demo tóm tắt văn bản sử dụng web page với chế độ custom.....	50



Hình 3.3.3.12 Kết quả tóm tắt văn bản sử dụng web page với chế độ custom.....	51
Hình 3.3.3.13 Kết quả độ đo tóm tắt văn bản sử dụng web page với chế độ custom.....	51

## CHƯƠNG 1. MỞ ĐẦU

### 1.1. Tính cấp thiết của đề tài [1]

Bất kỳ mô hình tính toán trên máy tính nào đều làm việc với các con số. Vậy làm thế nào để các mô hình tính toán có thể làm việc với ngôn ngữ tự nhiên? Mặt khác, từ là đơn vị ngôn ngữ nhỏ nhất mang ý nghĩa hoàn chỉnh. Do đó, để các mô hình làm việc được với ngôn ngữ tự nhiên thì việc số hóa các từ là cách tiếp cận đơn giản nhất.

Chúng ta có thể có một vài cách biểu diễn từ như sau:

Biểu diễn mỗi từ bằng một con số: đây là cách đơn giản nhất, tuy nhiên, có thể làm sai lệch mối quan hệ ngữ nghĩa giữa các từ. Nếu bạn biểu diễn “mèo” là số 1 và “chó” là số 2. Như vậy, ở một khía cạnh nào đó ta có: “mèo”+”mèo”=”chó”, “chó” > “mèo”. Mặt khác nếu một từ được biểu diễn là một vector thì dẫn đến sự bùng nổ của thư viện từ, những từ có nghĩa tương đồng nhau chưa được thể hiện trong phương pháp này.

Sử dụng “one-hot vector”: đây là vector có số chiều bằng số từ vựng. Vector này có duy nhất một chiều có giá trị bằng 1 ứng với từ đang biểu diễn, các vị trí khác có giá trị 0. Ví dụ [1,0,0,0...0]. Biểu diễn này giải quyết được các yếu điểm của biểu diễn bằng số. Tuy nhiên, nhược điểm của phương pháp này là số chiều vector rất lớn, ảnh hưởng đến quá trình xử lý cũng như lưu trữ.

Sử dụng vector ngẫu nhiên: với cách này, mỗi từ được biểu thị bằng một vector có giá trị của các chiều là ngẫu nhiên. Do đó, số lượng chiều chúng ta cần sử dụng ít hơn nhiều so với sử dụng one-hot. Ví dụ: nếu chúng ta có 1 triệu từ, chúng ta có thể biểu thị tất cả các từ đó trong không gian 3D, mỗi từ là một điểm trong không gian 3 chiều.

Sử dụng word embedding: đây được coi là cách tốt nhất để thể hiện các từ trong văn bản. Kỹ thuật này gán mỗi từ với một vector, nhưng ưu việt hơn kỹ thuật vector ngẫu nhiên vì các vector này được tính toán để biểu diễn quan hệ tương đồng giữa các từ.

## 1.2. Mục tiêu và nhiệm vụ nghiên cứu

Mục tiêu của đề tài là tập trung nghiên cứu cơ sở lý thuyết của kỹ thuật embedding cho xử lý ngôn ngữ tự nhiên, từ đó áp dụng vào bài toán tóm tắt văn bản. Chúng tôi sẽ tập trung tìm hiểu các cấp độ của bài toán, khai thác chiều sâu bài toán, các cách giải quyết bài toán phổ biến hiện nay.

Trong đề tài này, chúng tôi muốn tìm hiểu về kỹ thuật embedding cho xử lý ngôn ngữ tự nhiên, từ đó áp dụng vào mô hình tóm tắt văn bản để đưa ra bức tranh tổng quan về bài toán và ý nghĩa của nó. Để đạt được điều đó, chúng tôi tập trung vào tìm hiểu một số vấn đề sau:

1. Tìm hiểu cơ sở lý thuyết của kỹ thuật embedding cho xử lý ngôn ngữ tự nhiên.
2. Tìm hiểu cơ sở lý thuyết của bài toán tóm tắt văn bản.
3. Tìm hiểu các thuật toán tóm tắt cơ bản và các thuật toán dùng để xử lý các cấp độ của bài toán.
4. Ứng dụng mô hình tóm tắt để xây dựng nên một trang web đơn giản, nơi mà người dùng có thể nhập vào đoạn văn bản cần tóm tắt và kết quả sẽ nhận được đoạn văn bản đã được tóm tắt.
5. Đánh giá và giải thích kết quả.

## 1.3. Cách tiếp cận và phương pháp nghiên cứu

Word embedding là một trong những kỹ thuật được sử dụng nhiều nhất trong các bài toán xử lý ngôn ngữ tự nhiên.

Các thành công của các mô hình học sâu tân tiến nhất trong xử lý ngôn ngữ tự nhiên chính là nhờ một phần của kỹ thuật nhúng từ.

Word embedding sử dụng continuous vector, continuous vector có thể học được, vì thế biểu diễn được mối tương quan giữa các từ.

Có 2 loại chính của Word embedding là:

1. Frequency-based embedding:
  - + Dựa trên tần số xuất hiện của các từ trong Corpus.

- + Tiêu biểu ở phương pháp này là Glove, TF-IDF.

2. Prediction-based embedding: là phương pháp vector hóa dựa trên kết quả của một mô hình dự đoán.

- + Thường là mô hình mạng Neural.

- + Ví dụ tiêu biểu là Word2vec.

Sau khi tìm hiểu lý thuyết của kỹ thuật embedding cho xử lý ngôn ngữ tự nhiên, chúng tôi tiến hành áp dụng một phương pháp vào bài toán tóm tắt văn bản.

Hai cách tiếp cận chính để tóm tắt văn bản:

1. Tóm tắt dựa trên trích xuất

Cách tiếp cận này chọn những đoạn văn chính để tạo ra một bản tóm tắt. Cách tiếp cận này sẽ tìm ra phần quan trọng của tài liệu và xếp hạng chúng dựa trên mức độ quan trọng và độ giống nhau giữa các tài liệu. Kỹ thuật tóm tắt văn bản trích xuất bao gồm việc kéo các cụm từ chính từ tài liệu nguồn và kết hợp chúng để tạo thành một bản tóm tắt.

2. Tóm tắt dựa trên trừu tượng

Phương pháp trừu tượng chọn từ dựa trên sự hiểu biết ngữ nghĩa, thậm chí những từ đó không xuất hiện trong các tài liệu nguồn.

Do đó, trừu tượng thực hiện tốt hơn trích xuất. Tuy nhiên, các thuật toán tóm tắt văn bản được yêu cầu để thực hiện trừu tượng khó phát triển hơn và đó là lý do tại sao việc sử dụng trích xuất vẫn còn phổ biến.

Trong dự án này chúng tôi sẽ sử dụng tóm tắt dựa trên trích xuất cho bài toán của mình.

#### **1.4. Kết quả dự kiến đạt được**

Nhóm chúng tôi mong muốn sau khi thực hiện quá trình nghiên cứu nhiều công trình, cũng như các ứng dụng từ các tác giả đi trước, nhóm có thể học hỏi và đúc kết thành một bài báo cáo khai thác sâu về nội dung lý thuyết kỹ thuật embedding cho xử lý ngôn ngữ tự nhiên và áp dụng vào bài toán tóm tắt văn bản.

Về phần ứng dụng, để trực quan hóa bài toán, nhóm sẽ xây dựng một ứng dụng web đơn giản cho phép người dùng nhập vào đoạn văn bản và nhận lại được kết quả tóm tắt, để trực quan hóa kết quả của mô hình sau khi phân tích từ tập dữ liệu nhằm có cái nhìn cụ thể hơn cũng như thấy được sự hữu ích khi áp dụng vào thực tế.

## CHƯƠNG 2: NỘI DUNG

### 2.1. Cơ sở lý thuyết

#### 2.1.1. Tổng quan về xử lý ngôn ngữ tự nhiên

##### 2.1.1.1. Định nghĩa xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo, giúp máy tính hiểu, diễn giải và vận dụng ngôn ngữ của con người. NLP nhằm mục đích rút ngắn khoảng cách giao tiếp của con người và máy tính.

##### 2.1.1.2. Tầm quan trọng của xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên giúp máy tính có thể giao tiếp với con người bằng chính ngôn ngữ mà chúng ta sử dụng và nhiều vấn đề khác về ngôn ngữ.

Xử lý ngôn ngữ tự nhiên giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Mọi hoạt động của con người đều thông qua giao tiếp, một khi máy tính có thể hiểu được ngôn ngữ của chúng ta thì máy tính sẽ đem lại rất nhiều lợi ích. Tận dụng các đặc điểm của máy tính mà con người không thể so sánh như làm việc không biết mệt mỏi, công bằng tuyệt đối, ... thì việc máy tính hiểu được ngôn ngữ của chúng ta sẽ đem lại nhiều lợi ích, áp dụng vào nhiều lĩnh vực bài toán như:

- + Phân loại văn bản (text classification): ví dụ như phân loại bình luận của khách hàng về một sản phẩm trên sàn thương mại điện tử.

- + Trích xuất thông tin (information extraction): tìm kiếm bộ dữ liệu liên quan đến một truy vấn dữ liệu của chúng ta, ví dụ thông qua Google chúng ta tìm kiếm một từ khóa thì kết quả trả ra sẽ là nhiều bộ dữ liệu liên quan đến từ khóa đó.
- + Tác tử phần mềm hội thoại (conversational agent): các ứng dụng quen thuộc như các trợ lý ảo AI: Siri, Google home, ... các ứng dụng chatbox được nhiều doanh nghiệp sử dụng để tăng sự tương tác với khách hàng.
- + Tóm tắt văn bản (text summarization): từ một tập dữ liệu lớn ta sẽ rút ngắn lại thành một tập dữ liệu con mà nội dung chính ban đầu vẫn được đảm bảo.
- + Hỏi đáp (question answering): xây dựng các hệ thống có thể tự động trả lời các câu hỏi của con người. Chatbox được sử dụng ở các doanh nghiệp là một ví dụ điển hình.
- + Dịch máy (machine translation): là một nhánh con của xử lý ngôn ngữ học tính toán, liên quan đến việc chuyển đổi từ một ngôn ngữ này sang một ngôn ngữ khác. Một ví dụ điển hình là Google dịch.
- + Mô hình hóa chủ đề (topic modelling): là một kỹ thuật học máy không giám sát giúp khám phá chủ đề của một bộ tài liệu lớn. Mô hình hóa chủ đề có thể được mô tả như việc gán một chủ đề chung cho một tập hợp các tài liệu mô tả tốt nhất và phù hợp nhất với nội dung của các tài liệu đó.

### **2.1.1.3. Sự phát triển của xử lý ngôn ngữ tự nhiên [2]**

Mặc dù xử lý ngôn ngữ tự nhiên không phải là một lĩnh vực mới trong trí tuệ nhân tạo, nhưng lại đang phát triển nhanh chóng nhờ mối quan tâm ngày càng tăng trong giao tiếp giữa người với máy, cộng với sự

sẵn có của dữ liệu lớn, máy tính ngày càng mạnh mẽ hơn và các thuật toán ngày càng nâng cao.

Lịch sử phát triển của xử lý ngôn ngữ tự nhiên:

- **NLP tượng trưng (những năm 1950 - đầu những năm 1990)**

Tiền đề của NLP tượng trưng được tóm tắt kỹ lưỡng bởi thí nghiệm Chinese room của John Searle: đưa ra một bộ sưu tập các quy tắc (ví dụ: một cuốn sách từ vựng tiếng Trung, với các câu hỏi và câu trả lời phù hợp), máy tính mô phỏng hiểu ngôn ngữ tự nhiên (hoặc các nhiệm vụ NLP khác) bằng cách áp dụng các quy tắc đó cho dữ liệu mà nó nhận được.

**Những năm 1950:** thí nghiệm Georgetown năm 1954 liên quan đến việc dịch hoàn toàn tự động hơn 60 câu tiếng Nga sang tiếng Anh. Các tác giả tuyên bố rằng trong vòng ba hoặc năm năm, dịch máy sẽ là một vấn đề được giải quyết. Tuy nhiên, các thí nghiệm và nghiên cứu sau này không thực sự đem lại kết quả khả quan.

**Những năm 1960:** một số hệ thống xử lý ngôn ngữ tự nhiên thành công đáng chú ý được phát triển vào những năm 1960 là SHRDLU, một hệ thống ngôn ngữ tự nhiên hoạt động trong các "blocks worlds" với hạn chế về mặt từ vựng, có thể đưa ra các câu trả lời chung chung với các tình huống mà hệ thống không thể giải quyết.

**Những năm 1970:** trong những năm 1970, nhiều lập trình viên bắt đầu viết "conceptual ontologies", cấu trúc thông tin trong thế giới thực thành dữ liệu máy tính có thể hiểu được. Ví dụ như MARGIE (Schank, 1975), SAM (Cullingford, 1978), PAM (Wilensky, 1978), TaleSpin (Meehan, 1976), ... Trong thời gian này, các chatbot đầu tiên đã được viết.



**Những năm 1980:** những năm 1980 và đầu những năm 1990 đánh dấu thời kỳ hoàng kim của các phương pháp biểu tượng trong NLP. Các lĩnh vực trọng tâm của thời gian bao gồm nghiên cứu về phân tích cú pháp dựa trên quy tắc, hình thái học, ngữ nghĩa, tài liệu tham khảo và các lĩnh vực hiểu biết ngôn ngữ tự nhiên khác. Các dòng nghiên cứu khác đã được tiếp tục, ví dụ: sự phát triển của chatbots với Racter và Jabberwacky. Một bước phát triển quan trọng (cuối cùng dẫn đến sự thay đổi thống kê vào những năm 1990) là tầm quan trọng ngày càng tăng của đánh giá định lượng trong giai đoạn này.

- **NLP thống kê (những năm 1990 – 2010)**

Bắt đầu từ cuối những năm 1980, đã có một cuộc cách mạng trong xử lý ngôn ngữ tự nhiên với sự ra đời của các thuật toán học máy để xử lý ngôn ngữ.

**Những năm 1990:** nhiều thành công ban đầu đáng chú ý về các phương pháp thống kê trong NLP đã xảy ra trong lĩnh vực dịch máy. Nghiên cứu tại IBM Research nên các hệ thống này đã có thể tận dụng kho tài liệu văn bản đa ngôn ngữ hiện có do Nghị viện Canada và Liên minh Châu Âu tạo ra.

**Những năm 2000:** với sự phát triển mạnh mẽ của website dẫn đến lượng dữ liệu thô (chưa được gán nhãn) sinh ra rất nhiều. Các nghiên cứu tập trung vào các thuật toán học không có giám sát hoặc bán giám sát trên dữ liệu thô này nhưng thường cho ra kết quả không cao.

- **Neural NLP (hiện tại):** trong những năm 2010, representation learning - học biểu diễn và deep neural network - phương pháp học máy kiểu mạng nơron đã trở nên phổ biến trong xử lý ngôn ngữ tự nhiên. Sự phổ biến đó một phần là do một loạt các kết quả cho thấy

rằng các kỹ thuật như vậy có thể đạt được kết quả tốt trong nhiều nhiệm vụ ngôn ngữ tự nhiên.

## 2.1.2. Tổng quan về Embedding trong xử lý ngôn ngữ tự nhiên

### 2.1.2.1 Embedding [3]

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.

Ví dụ:

Document	Index	One-hot encoding
a	1	[1, 0, 0, ..., 0](9999 số 0)
b	2	[0, 1, 0, ..., 0]
c	3	[0, 0, 1, ..., 0]
....	.....	.....
mẹ	9999	[0, 0, 0, ..., 1, 0]
vân	10000	[0, 0, 0, ..., 0, 1]

Hình 2.1.2.1.1 Ví dụ minh họa One-hot encoding.[3]

Nhìn vào ảnh 2.1.2.1.1, ta thấy có 3 vấn đề khi ta biểu diễn dữ liệu dạng text dưới dạng **one-hot**.

- **Chi phí tính toán lớn:** nếu data có 100 từ, độ dài của vector one-hot là 100. Nếu data có 10000 từ, độ dài của vector one-hot là

10000. Tuy nhiên, để mô hình có độ khái quát cao thì trong thực tế dữ liệu có thể lên đến hàng triệu từ, lúc đó độ dài vector one-hot sẽ phình to gây khó khăn cho việc tính toán, lưu trữ.

- **Mang ít giá trị thông tin:** các vector hầu như toàn số 0. Chúng ta có thể thấy, đối với dữ liệu dạng text thì giá trị chứa trong các pixel (nếu input dạng ảnh) hay các dạng khác là rất ít, chủ yếu nằm trong vị trí tương đối giữa các từ với nhau và quan hệ về mặt ngữ nghĩa. Tuy nhiên, one-hot vector không thể biểu diễn điều đó vì kỹ thuật chỉ đánh index theo thứ tự từ điển đầu vào chứ không phải vị trí các từ trong một context cụ thể. Để khắc phục điều đó, trong mô hình thường dùng một lớp RNN hoặc LSTM để có thể trích xuất được thông tin về vị trí. Có một cách khác như trong mô hình transformer, được bỏ hoàn toàn lớp word embedding hay RNN và thêm vào đó lớp positional encoding và self-attention.
- **Độ khái quát yếu:** ví dụ ta có ba từ cùng chỉ *người mẹ*: mẹ, má, bầm. Tuy nhiên, từ bầm tương đối hiếm gặp trong tiếng Việt. Khi biểu diễn bằng one-hot encoding, khi đưa vào model train thì từ **bầm** mặc dù cùng nghĩa so với hai từ còn lại nhưng lại bị phân vào class khác nhau do cách biểu diễn khác nhau. Còn nếu dùng word embedding, do biểu diễn được cả thông tin về vị trí, ngữ nghĩa nên từ **bầm** sẽ có vị trí gần với hai từ còn lại.

### 2.1.2.2 Word embedding

Word embedding có 2 phương pháp là based method và predictive method. Nếu từ nào có cùng xuất hiện trong một văn bản hoặc một ngữ nghĩa thì chúng sẽ được sắp xếp cùng nhau.

Count-Based method: phương pháp này tính toán mức liên quan về mặt ngữ nghĩa giữa các từ bằng cách thống kê số lần đồng xuất hiện của một từ so với các từ khác.

- Mèo ăn cơm
- Mèo ăn cá
- Sai trong trường hợp nếu trong stop word tiếng việt như: mà, là , thì , do đó,... xuất hiện nhiều thì sẽ dẫn đến lu mờ các giá trị có nghĩa trong câu.
- Giải quyết vấn đề này chúng ta cần đánh dấu trọng số.

Thuật toán TF\_IDF\_Transform là một cách giải quyết vấn đề này. Trong đó:

TF: là tần suất xuất hiện của từ trong data.

IDF: là một hệ số giúp làm giảm trọng số của những từ hay xuất hiện trong data.

Nhờ thuật toán TF\_IDF\_Transform mà chúng ta có thể giúp làm giảm bớt trọng số của những từ xuất hiện nhiều nhưng lại không có nhiều thông tin.

TF-IDF (Term Frequency – Inverse Document Frequency) là 1 kỹ 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 được sử dụng để lọc những từ không mang quá nhiều ý nghĩa trong các bài toán như tóm tắt văn bản và phân loại văn bản.

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).

$$TF(t, d) = \frac{f(t, d)}{\max \{f(w, d) : w \in d\}}$$

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  $d$ .
- +  $\max(f(w, d) : w \in d)$ : số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản.

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 “thì”, “là” và “của” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Do đó chúng ta cần giảm độ quan trọng của những từ này xuống.

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

Trong đó:

- +  $IDF(t, D)$ : giá trị IDF của từ  $t$  trong tập văn bản.
- +  $|D|$ : tổng số văn bản trong tập  $D$ .

$+ |\{d \in D: t \in d\}|$  : thể hiện số văn bản trong tập  $D$  có chứa từ  $t$ .

Cơ số logarit trong công thức này sẽ thu hẹp khoảng giá trị của từ. Việc 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 đó chúng ta có công thức tính  $TF - IDF$  của một từ trong 1 văn bản là tích của  $TF$  và  $IDF$  của từ đó.

Công thức tổng quát:

$$TFIDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

Khi đó những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều và có tầm quan trọng cao trong văn bản. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao

Predictive methods (word2vec): ứng dụng của công cụ word2vec để khảo sát phản hồi phân tích, nhận xét, công cụ đề xuất.

Word2vec là một phương pháp cổ điển tạo ra các nhúng từ trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), được phát triển bởi Tomas Mikolov và nhóm của anh ấy tại Google vào năm 2013. Word2vec lấy từ vựng từ một kho văn bản lớn làm đầu vào và học cách đưa ra biểu diễn vector của chúng.

Tương tự như cách CNN trích xuất các tính năng từ hình ảnh, thuật toán word2vec trích xuất các tính năng từ văn bản cho các từ cụ thể. Sử dụng các tính năng đó, word2vec tạo ra các vector đại diện cho một từ trong

không gian vector. Các vector này được chọn bằng cách sử dụng hàm tương tự cosine, cho biết sự giống nhau về ngữ nghĩa giữa các từ.

Tương tự,  $\cosine = 1$  có nghĩa là góc giữa hai từ là 0 và sẽ biểu thị rằng các từ tương tự nhau. Hai từ giống nhau sẽ chiếm các vị trí gần nhau trong không gian vector đó, trong khi các từ rất khác nhau sẽ chiếm các khoảng cách xa nhau. Bằng cách đó, sử dụng khả năng đó với đại số tuyến tính, thuật toán có thể nhận ra ngữ cảnh và các từ có nghĩa tương tự.

Ví dụ: các từ “intelligent” và “smart” sẽ xuất hiện gần nhau hơn trong không gian vector này, trong khi các từ “engine” và “car” sẽ cách xa nhau. Điều này là do những từ này có sự hiểu biết theo ngữ cảnh trong không gian vector.

Có hai mô hình kiến trúc khác nhau của mô hình word2vec để tạo ra các đại diện nhúng từ.

Đó là:

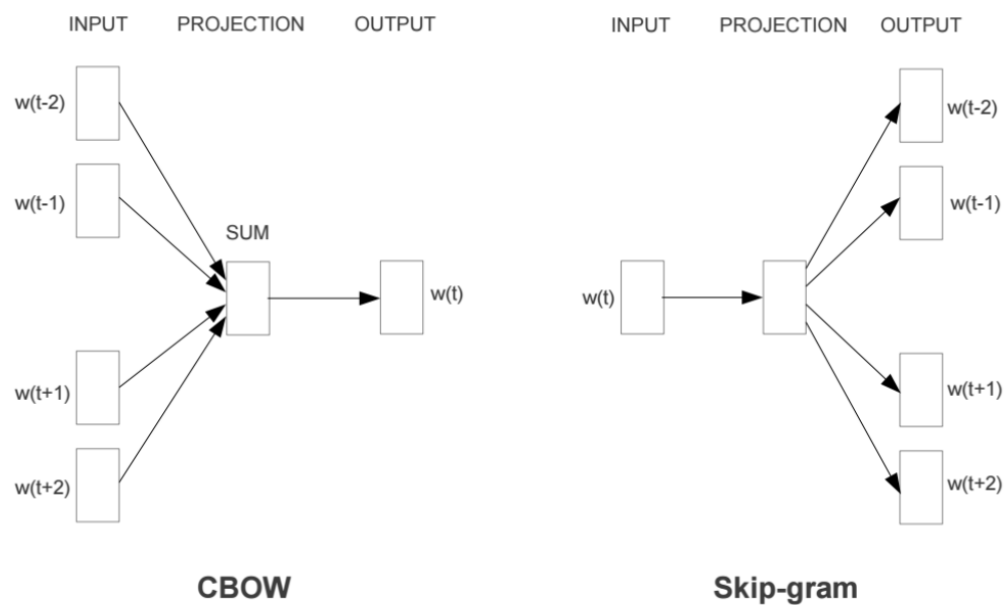
- + Mô hình skip-gram.
- + Mô hình continuous bag of words (CBOW).

Kiến trúc tổng quát:

Mô hình word2vec sử dụng một kiến trúc neural network đơn giản với một hidden layer duy nhất. Mục tiêu của mô hình là tìm hiểu ma trận trọng số cho các từ. Các trọng số này chính là các vector từ.

Mô hình này gồm 3 lớp chính:

- + Input layer: vector từng từ đầu vào được biểu diễn dạng one-hot.
- + Hidden layer: lớp nhúng.
- + Output layer: là một vector phân bố xác.



Hình 2.1.2.2.1 Mô phỏng các hoạt động của CBOW và Skip-gram.[3]  
 Ở hình trên ta thấy sự khác biệt kiến trúc giữa 2 mô hình CBOW và Skip-gram.

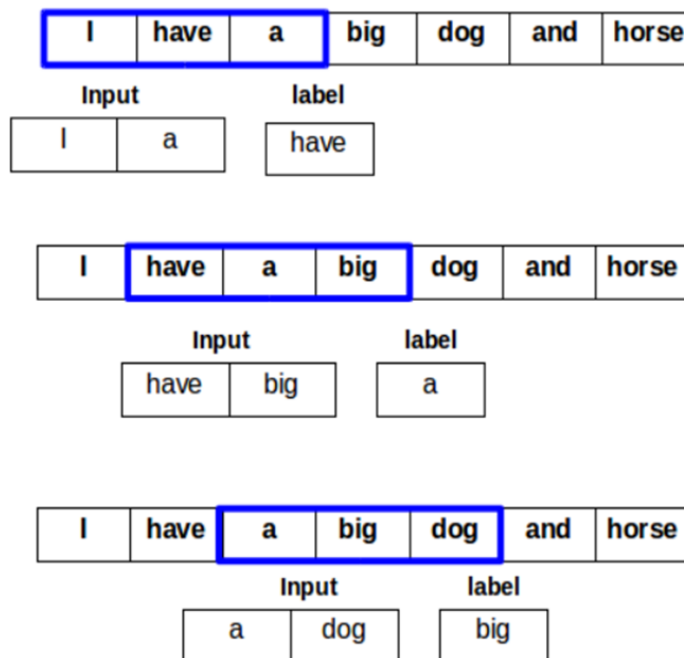
- CBOW

+ Ý tưởng của mô hình CBOW là dự đoán từ mục tiêu dựa trên ngữ cảnh xung quanh nó trong phạm vi nhất định. Cho từ mục tiêu tại  $w_c$ , tại vị trí  $C$  trong văn bản, khi đó đầu vào là các từ ngữ cảnh  $\{w_{c-m}, \dots, w_{c-1}, w_{c+1}, \dots, w_{c+m}\}$  xung quanh từ  $w_c$  trong phạm vi  $m$ .

Ví dụ:

Ta có một câu : **I have a big dog and horse**





Hình 2.1.2.2.2 Mô phỏng câu input với CBOW.[9]

Nhìn vào hình 2.1.2.2.2 ta thấy với  $m = 1$  và label là have thì input của CBOW: “I,a”

Với  $m=1$  và label là “a” thì input của CBOW: “have,big”

- Skip-gram
  - + Ngược lại với CBOW thì Skip-gram sẽ đưa vào một từ và dự đoán các từ xung quanh từ đó.

Ta có một sentence (câu văn) =  $\{w_1, w_2, \dots, w_n\}$

Khi đó chúng ta có thể biểu diễn toán học học

$$\left\{ w_{i_1}, w_{i_1}, \dots, w_{i_1} \left| \sum_{j=1}^n i_j - i_{j-1} < k \right. \right\}$$

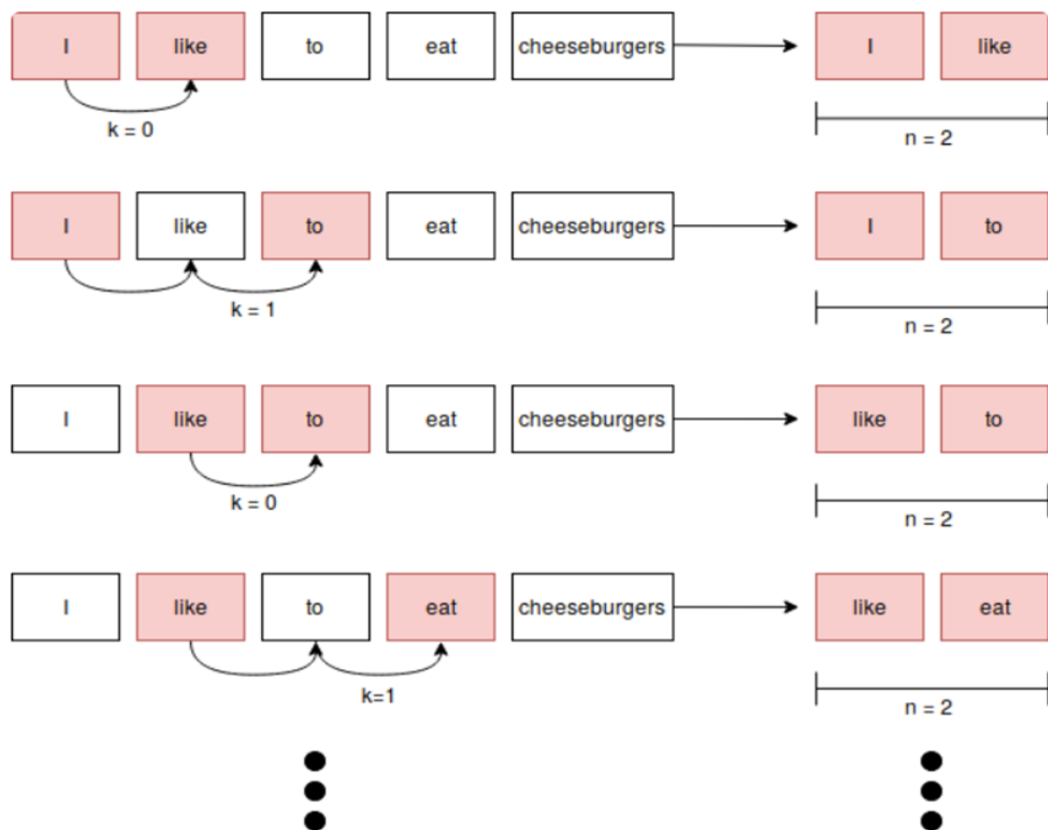
Với tham số  $K$  đề cập đến khoảng cách max skip và  $n$  là độ dài của dãy con.

Một giải thích đơn giản hơn: chúng tôi đang tìm kiếm một tập hợp các dãy con có độ dài  $n$  trong đó mỗi từ trong mỗi dãy con nhỏ hơn hoặc bằng một khoảng cách  $k$  (khi các từ cạnh nhau có khoảng cách bằng không).

Ví dụ: Chúng ta có câu bên dưới và đặt  $k = 1$  và  $n = 2$ .

**I Like to eat cheeseburgers**

**I like to eat cheeseburgers.**



Hình 2.1.2.2.3 Mô phỏng câu input với phương pháp Skip-gram .[10]

Ở trong hình 2.1.2.2.3 ta thấy nếu sử dụng skip-gram với  $k=1$  và  $n=2$  thì kết quả của Skip-gram sẽ là: {I to, like eat, to cheeseburgers}

## 2.2. Xây dựng ứng dụng

### 2.2.1. Tổng quan về tóm tắt văn bản

#### 2.2.1.1. Khái niệm tóm tắt văn bản

Tóm tắt văn bản là quá trình rút trích những thông tin quan trọng nhất từ một văn bản để tạo ra phiên bản ngắn gọn, súc tích mang lại 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 của văn bản.

#### 2.2.1.2. Tầm quan trọng của tóm tắt văn bản

Trong thời đại bùng nổ của thông tin dẫn đến rất nhiều thông tin được phát sinh và số hóa thành văn bản trên các trang web hay tài liệu. Mặt khác, con người ngày càng bận rộn với nhiều công việc thì việc tiếp cận và tiếp nhận thông tin một cách nhanh chóng là thực sự cần thiết.

Vì thế việc tóm tắt văn bản sẽ đem lại nhiều lợi ích rõ ràng cho chúng ta. sau đây là một số lợi ích mà tóm tắt văn bản sẽ đem lại :

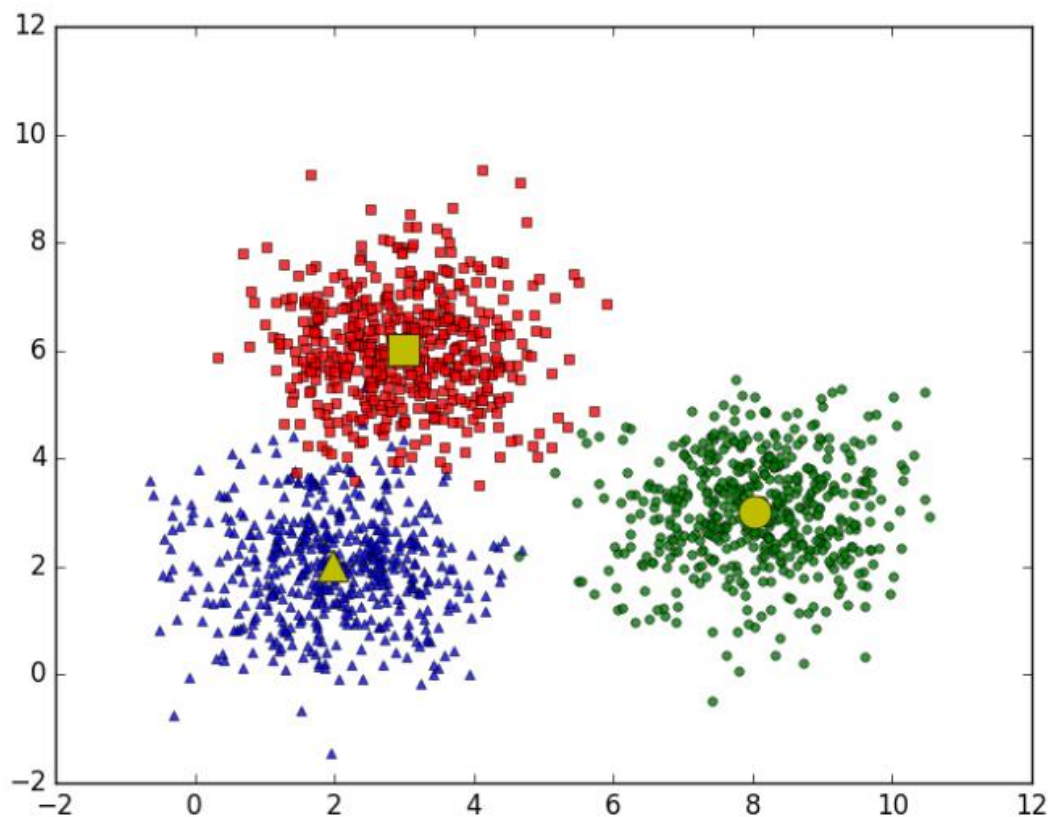
- + Làm cho việc đọc một cách dễ dàng hơn.
- + Tiết kiệm thời gian đọc hơn.
- + Dễ dàng ghi nhớ thông tin hơn.
- + Tăng hiệu suất công việc.

## 2.2.2. Thuật toán

### 2.2.2.1. Tổng quan về thuật toán K-means

K-means là một trong những thuật toán cơ bản nhất trong unsupervised learning. K-means sẽ phân cụm các điểm dữ liệu của chúng ta, sao cho các điểm trong cùng một cụm sẽ có tính chất giống nhau.

Mỗi cụm dữ liệu được đặc trưng bởi một tâm (centroid). Tâm là điểm đại diện nhất cho một cụm. Chúng ta sẽ dựa vào khoảng cách từ mỗi quan sát tới các tâm để xác định nhãn cho từng điểm dữ liệu sẽ thuộc cụm nào.



Hình 2.2.2.1.1 Minh họa phân cụm sử dụng thuật toán K-means.[8]

Trong hình minh họa 2.4.1.1 ta đang phân tập dữ liệu thành 3 cụm, các hình khối màu vàng là centroid của cụm.

Giả sử chúng ta có  $N$  điểm dữ liệu  $X = [x_1, x_2, \dots, x_n]$  và  $K < N$  là số cluster mà chúng ta muốn phân cụm. Các điểm centroid là  $M =$

$[m_1, m_2, \dots, m_k]$  của từng cụm.  $Y = [y_1, y_2, \dots, y_N]$  là ma trận được tạo bởi các label vector của mỗi điểm dữ liệu.

Nếu một điểm dữ liệu  $x_i$  được phân vào cụm có centroid là  $m_k$  thì sẽ bị sai số là  $(x_i - m_k)$ , chúng ta muốn sai số này có giá trị tuyệt đối nhỏ nhất.

Từ đó ta có thể suy ra được sai số cho toàn bộ tập dữ liệu là:

$$\zeta(Y, M) = \sum_{i=1}^N \sum_{j=1}^k y_{ij} \|x_i - m_j\|_2^2$$

Trong đó  $\|x_i - m_j\|_2^2$  chính là bình phương khoảng cách tính từ điểm  $x_i$  đến centroid  $m_j$ .

Việc cần làm tiếp theo là tối ưu cho hàm số ở trên.

$$\zeta(Y, M) = \arg \min \sum_{i=1}^N \sum_{j=1}^K y_{ij} \|x_i - m_j\|_2^2$$

Trong đó  $\arg \min$  là tìm biến số sao cho giá trị của hàm số đạt giá trị nhỏ nhất. Một cách đơn giản để giải bài toán là xen kẽ giải Y và M khi biến còn lại được cố định. Đây là một thuật toán lặp, cũng là kỹ thuật phổ biến khi giải bài toán tối ưu. Chúng ta sẽ lần lượt giải 2 bài toán:

- Cố định M, tìm Y: giả sử đã tìm được các centers, hãy tìm các label vector để hàm mất mát đạt giá trị nhỏ nhất. Điều này tương đương với việc tìm cluster cho mỗi điểm dữ liệu.
- Cố định Y, tìm M: giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.

Các bước thực hiện phân cụm:

Bước 1: chọn K điểm bất kỳ làm các centroid ban đầu.

Bước 2: phân mỗi điểm dữ liệu vào cluster có centroid gần nó nhất.

Bước 3: nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.

Bước 4: cập nhật centroid cho từng cluster bằng cách lấy trung bình cộng của tất cả các điểm dữ liệu đã được gán vào cluster đó sau bước 2.

Bước 5: quay lại bước 2.

Chúng ta có thể đảm bảo rằng thuật toán sẽ dừng lại sau một số hữu hạn vòng lặp. Thật vậy, vì hàm mất mát là một số dương và sau mỗi bước 2 hoặc 3, giá trị của hàm mất mát bị giảm đi. Theo kiến thức về dãy số nếu một dãy số giảm và bị chặn dưới thì nó hội tụ. Hơn nữa, số lượng cách phân nhóm cho toàn bộ dữ liệu là hữu hạn nên đến một lúc nào đó, hàm mất mát sẽ không thể thay đổi, và chúng ta có thể dừng thuật toán tại đây.

### 2.2.3 Độ đo

Để đánh giá một bài tóm tắt văn bản ta cần nhận xét một số tiêu chí sau:

- Sẽ có 4 tiêu chí để đánh giá một bài tóm tắt:

- Comprehensive - toàn diện: tóm gọn được các ý chính.
- Concise - ngắn gọn: bài tóm tắt phải ngắn gọn hơn bài gốc.
- Coherent - mạch lạc: bài tóm tắt nên có ý nghĩa mạch lạc, không phải là một tập hợp rời rạc các ý được lấy từ đoạn văn bản gốc.
- Not hallucinate - không ảo giác: bài tóm tắt không nên sinh ra các ý không nằm trong bài gốc.

#### 2.2.3.1 Độ đo Rouge [4]

Đây là chỉ số dựa trên việc tính toán sự trùng lặp cú pháp giữa bản tóm tắt và văn bản gốc (hoặc bất kỳ văn bản nào khác) .

Rouge-1: là tính toán sự chồng chéo của unigram (từ riêng lẻ) giữa các phần văn bản ứng viên và văn bản tham chiếu.

Rouge -2: tính toán sự chồng chéo của bigram (các cặp từ) giữa ứng cử viên và các văn bản tham khảo.

Rouge - L: tính toán sự trùng lặp của sự đồng xuất hiện dài nhất trong t dãy n gram giữa các thành phần văn bản ứng viên và tham chiếu.

Rouge-S: tính toán sự trùng lặp của skip-gram-bigram (*của bất kỳ cặp từ nào trong thứ tự câu của chúng*) giữa ứng viên và các đoạn văn bản tham chiếu

Ta có công thức:

$$R1 - recall \rightarrow \frac{\text{number} - \text{of} - \text{overlapping} - \text{words}}{\text{total} - \text{words} - \text{in} - \text{reference} - \text{summary}}$$

$$R1 - precision \rightarrow \frac{\text{number} - \text{of} - \text{overlapping} - \text{words}}{\text{total} - \text{words} - \text{in} - \text{system} - \text{summary}}$$

R1 - recall sẽ được tính bằng thương số của số từ được lặp lại trong câu tham khảo và tổng số từ trong câu tham khảo. R1- precision sẽ được tính bằng thương số của số từ được lặp lại và tổng số từ của câu ban đầu.

Ví dụ:

Ta có câu sau:

**The cat was under the bed**

Một câu sau khi biến đổi từ câu trên:

**The cat was found under the bed.**

Giả sử nếu ta xem xét các từ riêng rẽ, số lượng từ chồng chéo giữa tóm tắt hệ thống và văn bản người dùng bình thường. Để đánh giá độ chính xác (precision) và thu hồi (recall).

Ta có :

$$\text{Recal} = \frac{6}{6} = 1$$

$$\text{Precision} = \frac{6}{7} = 0.86$$

### 2.2.3.2 Độ đo bert score [5]

Đối với độ đo BLUE và ROUGE thì ta cần có bản tóm tắt tham khảo để phục vụ việc tính toán độ chính xác. Nhưng đối với Bert score thì không, nó sẽ so sánh bản tóm tắt tạo ra với bản gốc. Hai độ đo này chỉ thể hiện được sự hiện diện chính xác của từ trong bản tóm tắt có nằm trong bản tham khảo hay không mà không diễn giải được ngữ nghĩa của từ.

Bert score tính điểm tương đồng với mỗi token trong câu dự đoán (candidate sentence) với mỗi token trong câu tham khảo (reference sentence). Thay vì matches chính xác thì bert score sẽ tính toán sự tương đồng sử dụng những từ theo ngữ cảnh - contextual embeddings.

BertScore tính toán độ giống nhau của 2 câu dưới dạng tổng cosine tương đồng giữa các lần embedding. Độ tương đồng cosine của một token của câu ban đầu và ứng viên được tính theo công thức bên dưới.

$$\cos(x_i, x_j) = \frac{x_i^T \hat{x}_j}{\|x_i\| \|\hat{x}_j\|}$$

Trong công thức trên  $x_i x_j$ , lần lượt là các token trong câu ban đầu (reference sentence) và câu được sinh ra từ tóm tắt (candidate sentence).

Reference sentence có dạng  $x = \langle x_1, \dots, x_k \rangle$ , candidate sentence có dạng



$\hat{x} = \langle \hat{x}_1, \dots, \hat{x}_m \rangle$ . Công thức trên, chúng ta hiểu là cos của 2 vecto trong không gian.

Công thức tính các độ đo như recall, precision, F1 score được biểu diễn bên dưới.

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max x_i^T \hat{x}_j, \quad P_{BERT} = \frac{1}{|\hat{x}|} \sum_{x_i \in x} \max x_i^T \hat{x}_j$$

$$F_{BERT} = 2 \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}}$$

Mỗi token ở reference sentence được khớp với token tương đồng nhất ở candidate sentence. Sau đó, độ đo F1 sẽ được tính bằng việc kết hợp recall và precision.

Trọng số thể hiện độ quan trọng của các từ với nhau.

$$idf(\omega) = -\log \frac{1}{M} \sum_{i=1}^M \Pi[\omega \in x^{(i)}]$$

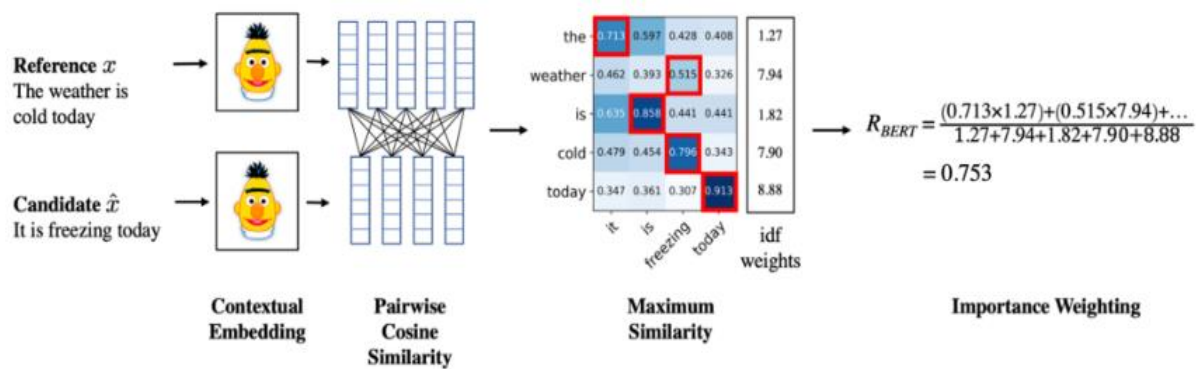
Trong công thức trên  $\Pi[.]$  là một hàm chỉ thị.

Thực tế cho thấy rằng các từ ít phổ biến sẽ thể hiện 2 câu có tương đồng chính xác hơn là các từ phổ biến trong các câu, vì thế cần có một trọng số độ quan trọng để giảm thiểu sự ảnh hưởng của các từ phổ biến đến điểm số tương đồng giữa 2 câu.

Khi đó các công thức tính độ đo ở trên sẽ được nhân thêm hệ số quan trọng. Ví dụ công thức của recall sẽ được viết lại như bên dưới.

$$R_{BERT} = \frac{\sum_{x_i \in x} idf(x_i) \max_{\hat{x}_j \in \hat{x}} x_i^T \hat{x}_j}{\sum_{x_i \in x} idf(x_i)}$$

Hình ảnh minh họa cách hoạt động của bert score.



Hình 2.2.3.2.1 Hình ảnh minh họa bert score. [5]

Với hình minh họa bên trên, cho câu ban đầu (reference sentence)  $x$  và câu được sinh ra từ tóm tắt (candidate sentence)  $\hat{x}$ , độ tương đồng của các từ trong câu sẽ được thể hiện và những từ có độ tương đồng cao nhất sẽ được tô viền màu đỏ. Trọng số quan trọng idf sẽ được tính ở cột bên phải của ma trận, sau đó áp dụng công thức tính  $R_{BERT}$ , ta sẽ tính được giá trị mong muốn.

## CHƯƠNG 3. THỰC NGHIỆM

### 3.1 Bài toán

Trong một thời đại mà mỗi ngày, mỗi giờ, mỗi phút đều có một lượng thông tin khổng lồ được sinh ra, nhưng giới hạn về thời gian, về khả năng đọc và tiếp thu của con người là có hạn, việc hiểu và nắm bắt nhiều thông tin một cách nhanh chóng không phải là vấn đề đơn giản với bất kỳ ai. Vì vậy vấn đề được đặt ra phải chuyển những dữ liệu văn bản này thành các bản tóm tắt ngắn hơn, tập trung nắm bắt các chi tiết nổi bật, để ta có thể điều hướng hiệu quả hơn cũng như kiểm tra xem các tài liệu lớn hơn có chứa thông tin mà ta đang tìm kiếm. Vì vậy nhóm muốn tạo một hệ thống tóm tắt văn bản tự động giúp người đọc giảm thời gian đọc tài liệu, khi nghiên cứu tài liệu nó sẽ giúp quá trình lựa chọn dễ dàng hơn, tăng hiệu quả của quá trình sắp xếp các kết quả tìm kiếm tài liệu của người dùng.











### 3.2 Dữ liệu [6]

Tập dữ liệu của nhóm được nhóm lấy từ bài báo “A Comparative Study on Vietnamese Text Classification Methods” của các tác giả Cong Duy Vu Hoang, Dien Dinh, Le Nguyen Nguyen, Quoc Hung Ngo, link bài viết

[https://www.researchgate.net/publication/4251746\\_A\\_Comparative\\_Study\\_on\\_Vietnamese\\_Text\\_Classification\\_Methods](https://www.researchgate.net/publication/4251746_A_Comparative_Study_on_Vietnamese_Text_Classification_Methods). Được tổng hợp lại ở link <https://github.com/duyvuleo/VNTC>, của tác giả Cong Duy Vu Hoang. Tập dữ liệu gồm nhiều file.txt chứa các đoạn văn bản tiếng việt với nhiều chủ đề khác nhau .















Trong dữ liệu sẽ chia thành 2 folder khác nhau với mỗi folder sẽ chia thành nhiều folder nhỏ chứa nhiều file txt với chủ đề khác nhau.

Folder thứ nhất gồm 10 chủ đề lớn như : chính trị - xã hội, đời sống, khoa học, kinh doanh, pháp luật, sức khỏe, thể thao, văn hóa, vi tính, được thể hiện như hình ảnh bên dưới.

	Chính trị Xã hội	4/9/2006 8:55 AM	File folder
	Đời sống	5/6/2006 3:03 PM	File folder
	Khoa học	4/9/2006 8:46 AM	File folder
	Kinh doanh	4/9/2006 8:46 AM	File folder
	Pháp luật	4/9/2006 8:47 AM	File folder
	Sức khỏe	4/9/2006 8:48 AM	File folder
	The giới	4/9/2006 8:49 AM	File folder
	The thao	4/9/2006 8:50 AM	File folder
	Văn hóa	5/22/2006 9:54 PM	File folder
	Vi tính	4/9/2006 8:54 AM	File folder

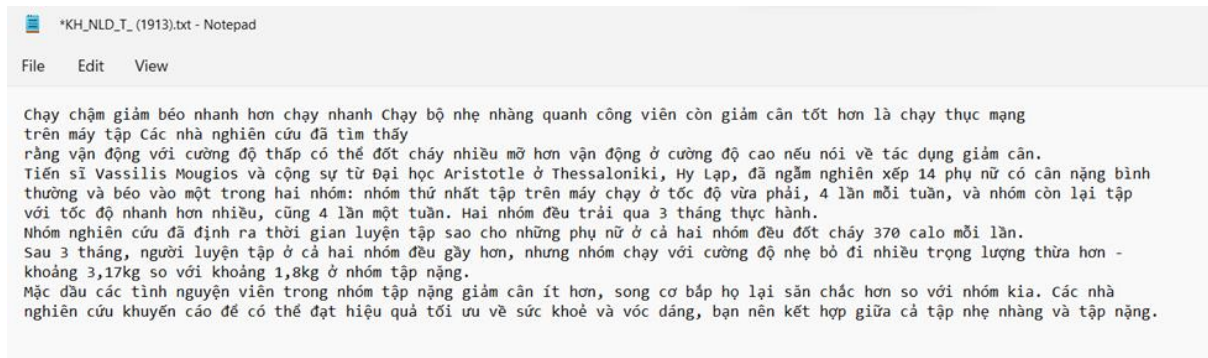
Hình 3.2.1 Các tập dữ liệu trong folder thứ nhất.

Trong mỗi chủ đề sẽ có nhiều file txt chứa nội dung của chủ đề đó, được thể hiện như hình ảnh bên dưới.

	KH_NLD_T_ (1909).txt	6/1/2006 9:18 PM	Text Document	2 KB
	KH_NLD_T_ (1910).txt	6/1/2006 9:18 PM	Text Document	2 KB
	KH_NLD_T_ (1911).txt	6/1/2006 9:18 PM	Text Document	3 KB
	KH_NLD_T_ (1912).txt	6/1/2006 9:18 PM	Text Document	3 KB
	KH_NLD_T_ (1913).txt	6/1/2006 9:18 PM	Text Document	3 KB
	KH_NLD_T_ (1915).txt	6/1/2006 9:18 PM	Text Document	2 KB
	KH_NLD_T_ (1916).txt	6/1/2006 9:18 PM	Text Document	2 KB
	KH_NLD_T_ (1917).txt	6/1/2006 9:18 PM	Text Document	9 KB
	KH_NLD_T_ (1918).txt	6/1/2006 9:18 PM	Text Document	10 KB
	KH_NLD_T_ (1919).txt	6/1/2006 9:18 PM	Text Document	5 KB
	KH_NLD_T_ (1920).txt	6/1/2006 9:18 PM	Text Document	12 KB
	KH_NLD_T_ (1921).txt	6/1/2006 9:18 PM	Text Document	12 KB
	KH_NLD_T_ (1922).txt	6/1/2006 9:18 PM	Text Document	12 KB
	KH_NLD_T_ (1923).txt	6/1/2006 9:18 PM	Text Document	9 KB

Hình 3.2.2 Các file dữ liệu txt trong một chủ đề.

Ví dụ về nội dung của một file trong chủ đề khoa học, được thể hiện như hình ảnh bên dưới.



Hình 3.2.3 Nội dung một file txt trong chủ đề khoa học.

Nội dung file sẽ là một đoạn văn có nội dung liên quan một sự vật, sự việc, hiện tượng, ... bất kỳ trong chủ đề khoa học. Với file ở trên là nội dung liên quan về “Chạy chậm giảm béo nhanh hơn chạy nhanh”.















Folder thứ 2 gồm 27 chủ đề cụ thể: ẩm thực, âm nhạc, bất động sản, bóng đá, chứng khoán,... được thể hiện như hình bên dưới.



Hình 3.2.4 Các tập dữ liệu trong folder thứ 2.

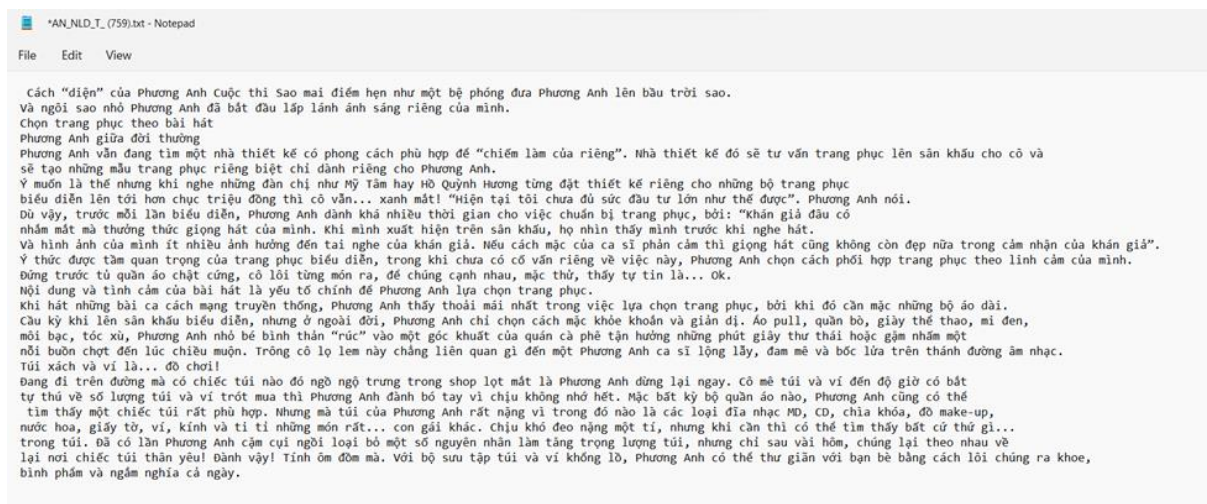
Hình ảnh bên dưới thể hiện các file txt trong chủ đề Âm nhạc. Mỗi chủ đề sẽ gồm nhiều file với định dạng .txt.

## Embedding cho xử lý ngôn ngữ tự nhiên

name	Date modified	Type	Size
 AN_NLD_T_ (759).txt	5/26/2006 1:52 AM	Text Document	6 KB
 AN_TN_T_ (761).txt	5/26/2006 1:44 AM	Text Document	2 KB
 AN_TN_T_ (762).txt	5/26/2006 1:52 AM	Text Document	5 KB
 AN_TN_T_ (763).txt	5/26/2006 1:44 AM	Text Document	9 KB
 AN_TN_T_ (764).txt	5/26/2006 1:44 AM	Text Document	2 KB
 AN_TN_T_ (765).txt	5/26/2006 1:44 AM	Text Document	7 KB
 AN_TN_T_ (766).txt	5/26/2006 1:44 AM	Text Document	6 KB
 AN_TN_T_ (767).txt	5/26/2006 1:44 AM	Text Document	8 KB
 AN_TN_T_ (768).txt	5/26/2006 1:44 AM	Text Document	2 KB
 AN_TN_T_ (769).txt	5/26/2006 1:44 AM	Text Document	10 KB
 AN_TN_T_ (770).txt	5/26/2006 1:44 AM	Text Document	8 KB
 AN_TN_T_ (771).txt	5/26/2006 1:44 AM	Text Document	3 KB
 AN_TN_T_ (772).txt	5/26/2006 1:44 AM	Text Document	8 KB
 AN_TN_T_ (773).txt	5/26/2006 1:44 AM	Text Document	6 KB

Hình 3.2.5 Các file dữ liệu txt trong một chủ đề.

Ví dụ về nội dung của một file trong chủ đề âm nhạc, được thể hiện như hình bên dưới.



Hình 3.2.6 Nội dung một file txt trong chủ đề âm nhạc.

Trong hình trên đây là nội dung của một bài nhận xét về ca sĩ Phương Anh trong chương trình âm nhạc Sao mai điểm hẹn.

### 3.3 Phương pháp và kết quả

#### 3.3.1. Phương pháp tạo model

Xây dựng mô hình word embedding – mô hình chuyển một tập từ vựng về một không gian vector mà vẫn giữ được đặc trưng ý nghĩa của từ với bộ dữ liệu ở trên .

*Input:* file đã được xây dựng từ tập dữ liệu trên gồm 2.385.532 câu tiếng Việt

*Công cụ:* thư viện gensim để tạo mô hình

*Output:* mô hình word2vec:

```
1 model = Word2Vec(input_gensim, vector_size=128, window=5, min_count=0, workers=4, sg=1)
2 model.wv.save("w2v.model")
```

Hình 3.3.1.1 Code xây dựng model.

Trong hình 3.3.1.1 , các tham số trong hàm Word2Vec:

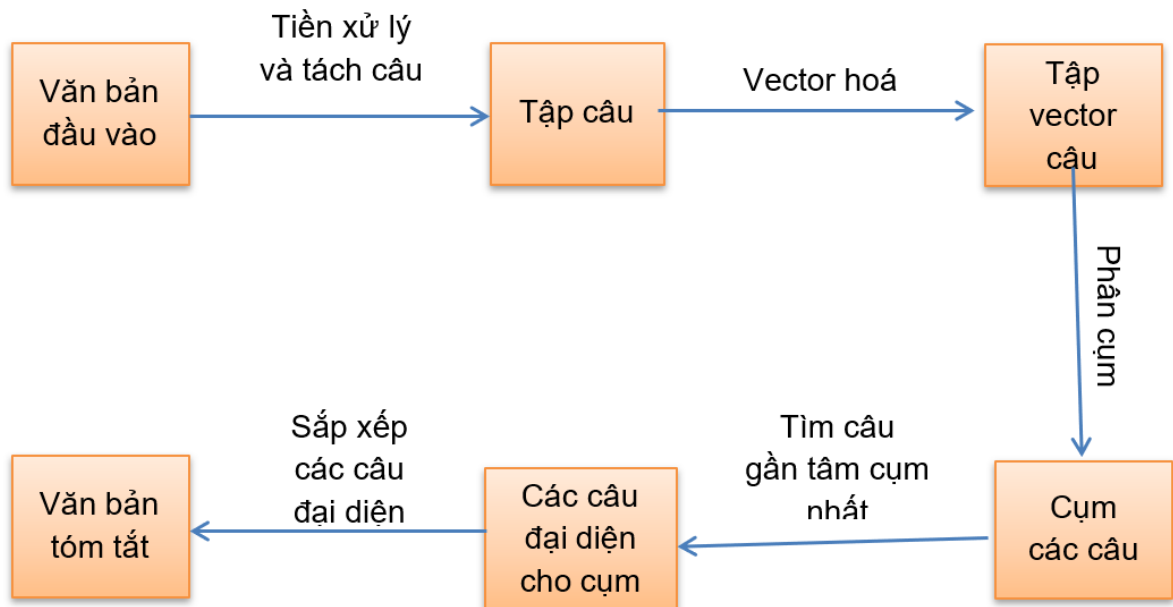
- input\_gensim: là dữ liệu từ tập input đầu vào.
- vector\_size: số chiều (kích thước) của vector tạo ra.
- window: khoảng cách tối đa từ hiện tại và từ dự đoán trong câu.
- min\_count: loại bỏ các từ có tần suất xuất hiện thấp hơn mức này.
- workers: số core của máy tính.
- sg: có 2 giá trị 1 (mô hình Skip-gram) và 0 cho CBOW.

Kích thước từ điển được tạo ra: 173444 .

Chức năng: biến đổi một từ có trong bộ từ điển thành một vector 128 chiều.

#### 3.3.2. Quy trình tạo ra kết quả từ model đã xây dựng





Hình 3.3.2.1 Mô hình luồng xử lý sử dụng model đã xây dựng.  
Trong ảnh trên là các bước của quá trình xây dựng mô hình tóm tắt văn bản sử dụng word2vec

#### Tiền xử lý văn bản

- Biến đổi về chữ thường.
- Loại bỏ khoảng trắng thừa.
- Tách từ (sử dụng công cụ tách từ ViTokenizer).

Ví dụ:

Trước khi xử lý:

*Giá xăng dầu được dự báo giảm nhẹ vào ngày mai*

Sau khi xử lý:

*giá\_xăng\_dầu\_được\_dự\_báo\_giảm\_nhẹ\_vào\_ngày\_mai*

#### Tách câu

Sử dụng thư viện nltk để tách câu.



Ví dụ:

Trước khi tách câu:

*Giá xăng dầu được báo giảm nhẹ vào ngày mai. Hôm nay là chủ nhật.*

Sau khi tách câu:

Câu 1: *giá xăng dầu được báo giảm nhẹ vào ngày mai.*

Câu 2: *hôm nay là chủ nhật.*

### **Vector hoá câu**

**Bước 1:** Tiến hành vector hoá câu:

*Input:* Câu đã được tiền xử lý và tách từ.

*Phương pháp:* Vector hoá từng từ của câu (bỏ qua những từ không có trong từ điển của mô hình word2vec). Sau đó cộng các vector đó lại.

*Output:* Vector 128 chiều đại diện cho câu.

Sau bước này, ta có N vector 128 chiều là biểu diễn của N câu trong đoạn văn bản đầu vào.

### **Phân cụm các câu**

N là số câu của văn bản ban đầu.

K là số câu của văn bản tóm tắt.

Sau bước trước, ta có N vector 128 chiều biểu diễn cho N câu trong đoạn văn input, tiến hành phân N vector này vào K cụm, sử dụng thuật toán phân cụm: K-Means Clustering.

### **Tìm câu đại diện cho cụm**

Xác định K câu gần K tâm cụm nhất. Các câu đó mang ý nghĩa tổng quát nhất cho cụm, có thể đại diện cho K cụm.

### Sắp xếp các câu đại diện cụm thành đoạn tóm tắt

Tính thứ tự trung bình của từng cụm. Xếp k câu đại diện theo đúng thứ tự trung bình của cụm chứa câu đó.

$$\text{Thứ tự trung bình} = \frac{\text{Tổng số thứ tự các câu trong cụm}}{\text{Số câu của cụm}}$$

**Ví dụ:** thứ tự các câu trong input: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.

Kết quả phân cụm (câu **in đậm và gạch chân** là câu đại diện cho cụm):

Cụm 1: [ 2, **8** ]      => Thứ tự trung bình:  $T1 = (2 + 8)/2 = 5.0$

Cụm 2: [ **0**, 1, 10 ]      => Thứ tự trung bình:  $T2 = (0 + 1 + 10)/3 = 3.67$

Cụm 3: [ 3, 4, **5**, 6, 7, 9, 11 ]

=> Thứ tự trung bình:  $T3 = (3+4+5+6+7+9+11)/7=6.43$

Ta thấy:  $T2 < T1 < T3$

Do đó, thứ tự sắp xếp: Cụm 2 => Cụm 1 => Cụm 3

Vậy đoạn tóm tắt là: Câu 0 => Câu 8 => Câu 5

### 3.3.3 . Kết quả

Sau khi tạo xong model “w2v.model”. Nhóm bắt đầu thử tóm tắt một đoạn trong bài báo đơn giản bằng thuật toán Kmean. Nội dung bài báo như hình bên dưới:

```

2 content = ""
3 Cụ thể, năm 2023, ngày 1/1 rơi vào ngày Chủ nhật trong tuần, do đó, người lao động sẽ
4 được sắp xếp nghỉ bù theo quy định tại khoản 3 Điều 111 Bộ luật Lao động 2019:
5 Nếu ngày nghỉ hằng tuần trùng với ngày nghỉ lễ, tết quy định tại khoản 1 Điều 112 của
6 Bộ luật này thì người lao động được nghỉ bù ngày nghỉ hằng tuần vào ngày làm việc kế tiếp.
7 Theo đó, dịp Tết Dương lịch 2023, người lao động sẽ được nghỉ bù vào ngày 2/1/2023
8 Như vậy, nếu làm việc theo chế độ nghỉ thứ Bảy, Chủ nhật thì lịch nghỉ Tết Dương lịch 2023 sẽ
9 kéo dài 31/12/2022 đến hết ngày 2/1/2023, tức nghỉ 3 ngày.
10 Trường hợp làm việc theo chế độ nghỉ 1 ngày/tuần là ngày Chủ nhật thì lịch nghỉ Tết Dương lịch 2023 c
11 ủa người lao động là từ 1/1/2023 đến hết ngày 2/1/2023, tức kéo dài 2 ngày
12 Còn theo Điểm a, Khoản 1 Điều 112 Bộ luật Lao động năm 2019, thời gian nghỉ Tết Dương lịch được quy định:
13 Người lao động được nghỉ làm việc, hưởng nguyên lương Tết Dương lịch 1 ngày (ngày 1 tháng 1 dương lịch).
14 Như vậy, dịp Tết Dương lịch năm 2023, người lao động được nghỉ làm, hưởng nguyên lương trong ngày 1/1.
15 Còn ngày nghỉ thứ 7 và ngày nghỉ bù 2/1/2023 nghỉ theo chế độ thông thường.
16 Dựa theo lịch nghỉ Tết Dương lịch năm 2023, người lao động có thể bố trí thời gian về quê
17 hoặc tham gia các chương trình du lịch, dã ngoại theo khung thời gian trên.
18 ""

```

### Hình 3.3.3.1 Nội dung của đoạn văn bản được tóm tắt.

Trong hình 3.3.3.1 chứa nội dung của một văn bản mà chúng ta cần tóm tắt

Sau khi lấy nội dung từ content: ta bắt đầu cắt những câu trong văn bản thành các câu khác nhau (phương pháp cắt câu từ đầu tới dấu “.” sẽ kết thúc câu) sau đó tiền xử lý đầu vào và vector hóa ta phân được thành các câu như sau.

```

===
0 === cụ thể, năm 2023, ngày 1/1 rơi vào ngày chủ nhật trong tuần, do đó, người lao động sẽ được sắp xếp nghỉ bù theo quy định tại khoản 3 điều 111 bộ luật lao động 2019:
nếu ngày nghỉ hằng tuần trùng với ngày nghỉ lễ, tết quy định tại khoản 1 điều 112 của bộ luật này thì người lao động được nghỉ bù ngày nghỉ hằng tuần vào ngày làm việc kế tiếp.
1 === theo đó, dịp tết dương lịch 2023, người lao động sẽ được nghỉ bù vào ngày 2/1/2023 như vậy, nếu làm việc theo chế độ nghỉ thứ bảy, chủ nhật thì lịch nghỉ tết
dương lịch 2023 sẽ kéo dài 31/12/2022 đến hết ngày 2/1/2023, tức nghỉ 3 ngày.
2 === trường hợp làm việc theo chế độ nghỉ 1 ngày/tuần là ngày chủ nhật thì lịch nghỉ tết dương lịch 2023 c ủa người lao động là từ 1/1/2023 đến hết ngày 2/1/2023,
tức kéo dài 2 ngày còn theo điểm a, khoản 1 điều 112 bộ luật lao động năm 2019, thời gian nghỉ tết dương lịch được quy định:
người lao động được nghỉ làm việc, hưởng nguyên lương tết dương lịch 1 ngày (ngày 1 tháng 1 dương lịch).
3 === như vậy, dịp tết dương lịch năm 2023, người lao động được nghỉ làm, hưởng nguyên lương trong ngày 1/1.
4 === còn ngày nghỉ thứ 7 và ngày nghỉ bù 2/1/2023 nghỉ theo chế độ thông thường.
5 === dựa theo lịch nghỉ tết dương lịch năm 2023, người lao động có thể bố trí thời gian về quê hoặc tham gia các chương trình du lịch, dã ngoại theo khung thời gian trên.
===

```

### Hình 3.3.3.2 Các câu được tách từ đoạn nội dung ban đầu.

Trong hình 3.3.3.2 sau khi chúng ta lấy nội dung đầu vào , chúng ta sẽ cắt nó thành các câu khác nhau , riêng biệt nhau.

Chi tiết các câu sau khi cắt

```

{
0 === cụ thể, năm 2023, ngày 1/1 rơi vào ngày chủ nhật trong tuần, do đó, người lao động sẽ
được sắp xếp nghỉ bù theo quy định tại khoản 3 điều 111 bộ luật lao động 2019: nếu ngày
nghỉ hằng tuần trùng với ngày nghỉ lễ, tết quy định tại khoản 1 điều 112 của bộ luật này thì
người lao động được nghỉ bù ngày nghỉ hằng tuần vào ngày làm việc kế tiếp.

```

- 1 === theo đó, dịp tết dương lịch 2023, người lao động sẽ được nghỉ bù vào ngày 2/1/2023 như vậy, nếu làm việc theo chế độ nghỉ thứ bảy, chủ nhật thì lịch nghỉ tết dương lịch 2023 sẽ kéo dài 31/12/2022 đến hết ngày 2/1/2023, tức nghỉ 3 ngày.
  - 2 === trường hợp làm việc theo chế độ nghỉ 1 ngày/tuần là ngày chủ nhật thì lịch nghỉ tết dương lịch 2023 của người lao động là từ 1/1/2023 đến hết ngày 2/1/2023, tức kéo dài 2 ngày còn theo điểm a, khoản 1 điều 112 bộ luật lao động năm 2019, thời gian nghỉ tết dương lịch được quy định: người lao động được nghỉ làm việc, hưởng nguyên lương tết dương lịch 1 ngày (ngày 1 tháng 1 dương lịch).
  - 3 === như vậy, dịp tết dương lịch năm 2023, người lao động được nghỉ làm, hưởng nguyên lương trong ngày 1/1.
  - 4 === còn ngày nghỉ thứ 7 và ngày nghỉ bù 2/1/2023 nghỉ theo chế độ thông thường.
  - 5 === dựa theo lịch nghỉ tết dương lịch năm 2023, người lao động có thể bố trí thời gian về quê hoặc tham gia các chương trình du lịch, dã ngoại theo khung thời gian trên.
- }

Sau khi xử lý xong đoạn văn bản đầu vào ta dùng thuật toán Kmean với tham số  $k = 3$  (nhóm chọn ngẫu nhiên) để phân cụm các câu văn đã cắt ở trên.

```

Cụm 1 [3 4 5]
Cụm 2 [0 2]
Cụm 3 [1]
=====
Thứ tự trung bình = 4.0
Câu gần tâm cụm nhất: 3
=====
như vậy, dịp tết dương lịch năm 2023, người lao động được nghỉ làm, hưởng nguyên lương trong ngày 1/1.
=====
Thứ tự trung bình = 1.0
Câu gần tâm cụm nhất: 0
=====
cụ thể, năm 2023, ngày 1/1 rơi vào ngày chủ nhật trong tuần, do đó, người lao động sẽ được sắp xếp nghỉ bù theo quy định tại khoản 3 điều 111 bộ luật lao động 2019: nếu ngày nghỉ hằng tuần trùng với ngày nghỉ lễ, tết quy định tại khoản 1 điều 112 của bộ luật này thì người lao động được nghỉ bù ngày nghỉ hằng tuần vào ngày làm việc kế tiếp.
=====
Thứ tự trung bình = 1.0
Câu gần tâm cụm nhất: 1
=====
theo đó, dịp tết dương lịch 2023, người lao động sẽ được nghỉ bù vào ngày 2/1/2023 như vậy, nếu làm việc theo chế độ nghỉ thứ bảy, chủ nhật thì lịch nghỉ tết dương lịch 2023 sẽ kéo dài 31/12/2022 đến hết ngày 2/1/2023, tức nghỉ 3 ngày.
=====

```

### Hình 3.3.3.3 Kết quả phân cụm các câu.

Trong hình 3.3.3.3 kết quả của các câu được cắt ra sau khi qua thuật toán kmean nó sẽ được phân thành các cụm khác nhau như hình trên.

Sau khi tính toán thứ tự của các câu ta sẽ được kết quả ta nhận được là:

Kết quả tóm tắt:

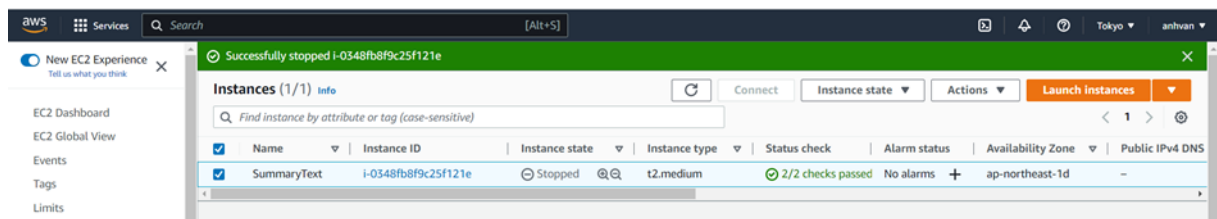
(Câu 0) cụ thể, năm 2023, ngày 1/1 rơi vào ngày chủ nhật trong tuần, do đó, người lao động sẽ được sắp xếp nghỉ bù theo quy định tại khoản 3 điều 111 bộ luật lao động 2019: nếu ngày nghỉ hàng tuần trùng với ngày nghỉ lễ, tết quy định tại khoản 1 điều 112 của bộ luật này thì người lao động được nghỉ bù ngày nghỉ hàng tuần vào ngày làm việc kế tiếp. (Câu 1) theo đó, dịp tết dương lịch 2023, người lao động sẽ được nghỉ bù vào ngày 2/1/2023 như vậy, nếu làm việc theo chế độ nghỉ thứ bảy, chủ nhật thì lịch nghỉ tết dương lịch 2023 sẽ kéo dài 31/12/2022 đến hết ngày 2/1/2023, tức nghỉ 3 ngày. (Câu 3) như vậy, dịp tết dương lịch năm 2023, người lao động được nghỉ làm, hưởng nguyên lương trong ngày 1/1.

### Hình 3.3.3.4 Kết quả tóm tắt.

Trong hình 3.3.3.4 kết quả sau khi chúng ta tóm tắt môn văn bản đầu vào.

## Deploy ứng dụng demo web lên dịch vụ EC2 của AWS

Chúng tôi tiến hành sử dụng dịch vụ ec2 của aws để depoly trang web của mình. Chúng tôi sử dụng một máy ảo ubuntu server với dung lượng 4gb ram và 15gb ổ đĩa. Sau đó tiến hành chuyển file project của nhóm lên máy ảo và run app.

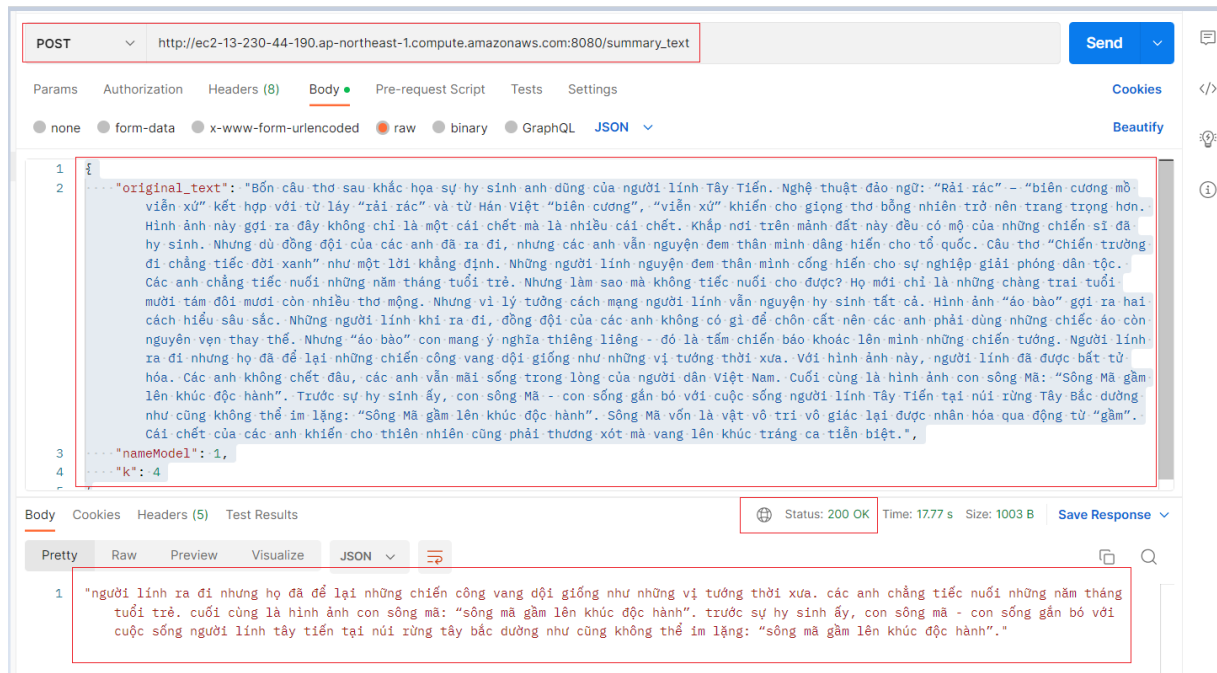


### Hình 3.3.3.5 Hình ảnh máy ảo trên aws

## Xây dựng API

Chúng tôi thực hiện xây dựng một API tóm tắt văn bản, phù hợp để tích hợp vào các hệ thống website đã có sẵn. Ở API này, người dùng sẽ truyền vào nội dung cần tóm tắt và kết quả trả ra sẽ là nội dung đã tóm tắt.

Chúng tôi sử dụng ứng dụng postman để thử nghiệm API của mình. Kết quả được thể hiện ở hình ảnh bên dưới.



Hình 3.3.3.6 Kết quả thực thi API.

Trong hình trên các tham số là :

- + `original_text` chứa nội dung của văn bản gốc
- + `nameModel` là giá trị của model, 1 là skip-gram, giá trị khác là CBOW.
- + `k` là số câu của đoạn văn bản tóm tắt.

Chi tiết các tham số và kết quả như sau.

## Tham số

{

**"original\_text":** "Bốn câu thơ sau khắc họa sự hy sinh anh dũng của người lính Tây Tiến. Nghệ thuật đảo ngữ: "Rải rác" – "biên cương mờ viễn xứ" kết hợp với từ láy "rải rác" và từ Hán Việt "biên cương", "viễn xứ" khiến cho giọng thơ bỗng nhiên trở nên trang trọng hơn. Hình ảnh này gợi ra đây không chỉ là một cái chết mà là nhiều cái chết. Khắp nơi trên mảnh đất này đều có mộ của những chiến sĩ đã hy sinh. Nhưng dù đồng đội của các anh đã ra đi, nhưng các anh vẫn nguyện đem thân mình dâng hiến cho tổ quốc. Câu thơ "Chiến trường đi chẳng tiếc đời xanh" như một lời khẳng định. Những người lính nguyện đem thân mình cống hiến cho sự nghiệp giải phóng dân tộc. Các anh chẳng tiếc nuôi những năm tháng tuổi trẻ. Nhưng làm sao mà không tiếc nuôi cho được? Họ mới chỉ là những chàng trai tuổi mười tám đôi mươi còn nhiều thơ mộng. Nhưng vì lý tưởng cách mạng người lính vẫn nguyện hy sinh tất cả. Hình ảnh "áo bào" gợi ra hai cách hiểu sâu sắc. Những người lính khi ra đi, đồng đội của các anh không có gì để chôn cất nên các anh phải dùng những chiếc áo còn nguyên vẹn thay thế. Nhưng "áo bào" còn mang ý nghĩa thiêng liêng - đó là tấm chiến bào khoác lên mình những chiến tướng. Người lính ra đi nhưng họ đã để lại những chiến công vang dội giống như những vị tướng thời xưa. Với hình ảnh này, người lính đã được bất tử hóa. Các anh không chết đâu, các anh vẫn mãi sống trong lòng của người dân Việt Nam.

Cuối cùng là hình ảnh con sông Mã: “Sông Mã gầm lên khúc độc hành”. Trước sự hy sinh ấy, con sông Mã - con sông gắn bó với cuộc sống người lính Tây Tiến tại núi rừng Tây Bắc dường như cũng không thể im lặng: “Sông Mã gầm lên khúc độc hành”. Sông Mã vốn là vật vô tri vô giác lại được nhân hóa qua động từ “gầm”. Cái chết của các anh khiến cho thiên nhiên cũng phải thương xót mà vang lên khúc tráng ca tiễn biệt.”,

```
"nameModel": 1,  
"k": 4  
}
```

### Kết quả

“nghệ thuật đảo ngữ: “rải rác” – “biên cương mờ viễn xứ” kết hợp với từ láy “rải rác” và từ Hán Việt “biên cương”, “viễn xứ” khiến cho giọng thơ bỗng nhiên trở nên trang trọng hơn. hình ảnh này gợi ra đây không chỉ là một cái chết mà là nhiều cái chết. với hình ảnh này, người lính đã được bất tử hóa. nhưng dù đồng đội của các anh đã ra đi, nhưng các anh vẫn nguyện đem thân mình dâng hiến cho tổ quốc. những người lính khi ra đi, đồng đội của các anh không có gì để chôn cất nên các anh phải dùng những chiếc áo còn nguyên vẹn thay thế. nhưng “áo bào” con mang ý nghĩa thiêng liêng - đó là tấm chiến bào khoác lên mình những chiến tướng. cuối cùng là hình ảnh con sông mã: “sông mã gầm lên khúc độc hành”. trước sự hy sinh ấy, con sông mã - con sông gắn bó với cuộc sống người lính tây tiến tại núi rừng tây bắc dường như cũng không thể im lặng: “sông mã gầm lên khúc độc hành”

Với kết quả đoạn văn bản tóm tắt trên thì sẽ cho kết quả độ đo Precision của bert score cao nhất là 0.7702.

```
done in 1.30 seconds, 0.77 sentences/sec  
Số cụm cho kết quả tốt nhất : 8 với score là : tensor([0.7702])  
171.252.154.249 - - [27/Dec/2022 05:23:42] "POST /summary_text HTTP/1.1" 200 -
```

Hình 3.3.3.7 Số câu tóm tắt và độ đo Precision tốt nhất.

### Xây dựng Web page

Chúng tôi tiến hành xây dựng một web page đơn giản, ở đó người dùng có thể nhập vào nội dung cần tóm tắt, chọn được thuật toán và số câu của đoạn văn bản đầu ra mong muốn. Sẽ có 2 option mà người dùng có thể lựa chọn là Default hoặc Custom. Đối với Default thì người dùng chỉ cần nhập vào đoạn văn bản cần tóm tắt, việc chọn số câu của văn bản đầu ra sẽ được lấy dựa trên kết quả tốt nhất của độ đo. Đối với option Custom, người dùng cần chọn các tham số như thuật toán và số câu của đoạn văn bản tóm tắt đầu ra mà người dùng mong muốn.

Chúng tôi sẽ tiến hành demo tóm tắt văn bản ở chế độ Default. Người dùng sẽ lần lượt thực hiện nhập đoạn văn bản vào ô input sau đó nhấn nút Default, các thông tin như số lượng câu của input và các tham số dùng cho việc tóm tắt văn bản sẽ được chọn và hiển thị như hình bên dưới.

The screenshot shows a web browser window with the URL `ec2-13-230-44-190.ap-northeast-1.compute.amazonaws.com:3080`. The page title is "TÓM TẮT VĂN BẢN TIẾNG VIỆT" (Vietnamese Text Summarization) and the subtitle is "Sử dụng Word2Vec và thuật toán K-mean". Below the title, there is a text input field with a "Choose File" button and a "No file chosen" status. The input field contains a sample text about a child named Lai. To the right of the input field is an "Output" section. Below the input field, there are several controls: a "Số câu đầu vào của đoạn văn" (Number of input sentences of the text) field with the value "13", a "Chọn model" (Choose model) dropdown menu with "Default" selected, and a "Số lượng câu" (Number of sentences) field with the value "default". At the bottom, there are "Submit" and "Reset" buttons.

Hình 3.3.3.8 Web page tóm tắt văn bản sử dụng chế độ Default.

Với demo trên, chúng tôi nhập vào input có độ dài của đoạn là 13 câu. Khi chúng tôi chọn chế độ Default thì các tham số như model sẽ nhận giá trị default - mặc định, số lượng câu sẽ được chọn tương ứng với độ đo mang giá trị cao nhất. Chi tiết input như sau:

“ Nó tên là Lai, cái tên mà em đặt cho nó khi nó còn bé tí tẹo. Nó là con vật mà em vô cùng yêu quý.

Ba bảo: “Giống chó này quý lắm con ạ! Ba dặn đi dặn lại nhiều lần, với lại ở chỗ thân quen bác ấy mới ưu tiên cho mình con Lai này đây, ráng mà nuôi dạy cho kĩ!”.

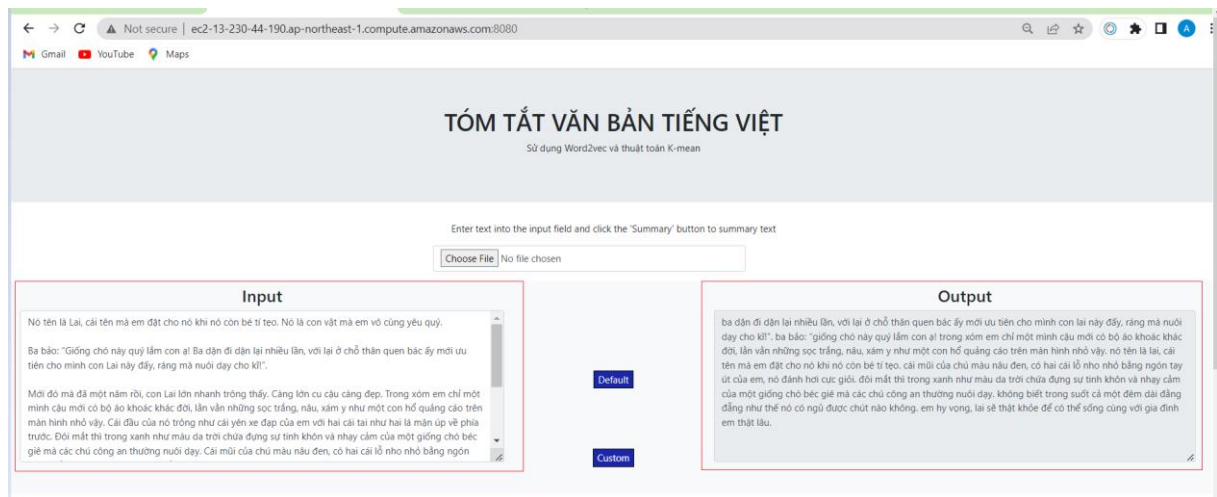
Mới đó mà đã một năm rồi, con Lai lớn nhanh trông thấy. Càng lớn cu cậu càng đẹp. Trong xóm em chỉ một mình cậu mới có bộ áo khoác khác đời, lần vắn những sọc trắng, nâu, xám y như một con hổ quảng cáo trên màn hình nhỏ vậy. Cái đầu của nó trông như cái yên xe đạp của em với hai cái tai như hai lá mạn úp về phía trước. Đôi mắt thì trong xanh như màu da trời chứa đựng sự tinh khôn và nhạy cảm của một giống chó béc giê mà các chú công an thường nuôi dạy. Cái mũi của chú màu nâu đen, có hai cái lỗ nho nhỏ bằng ngón tay út của em, nó đánh hơi cực giỏi.

Tối đến, Lai thường nằm ngủ ở bậc thềm ngoài hiên để canh chừng kẻ trộm. Không biết trong suốt cả một đêm dài đằng đẵng như thế nó có ngủ được chút nào không. Bất kì một tiếng động nhỏ nào chú cũng đều phát hiện được cả

Lai khôn ngoan và lanh lợi nên cả nhà em ai cũng quý nó. Em hy vọng, Lai sẽ thật khỏe để có thể sống cùng với gia đình em thật lâu.”



Sau khi nhấn nút Submit và chờ trong giây lát, chúng ta sẽ nhận được kết quả được thử hiện như hình bên dưới.

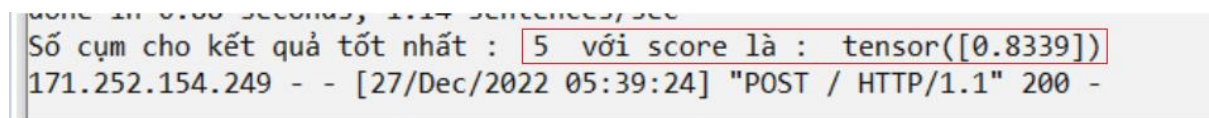


Hình 3.3.3.9 Kết quả tóm tắt văn bản sử dụng web page với chế độ default.

Kết quả của văn bản tóm tắt sẽ được hiển thị ở ô output. Chi tiết của output như sau:

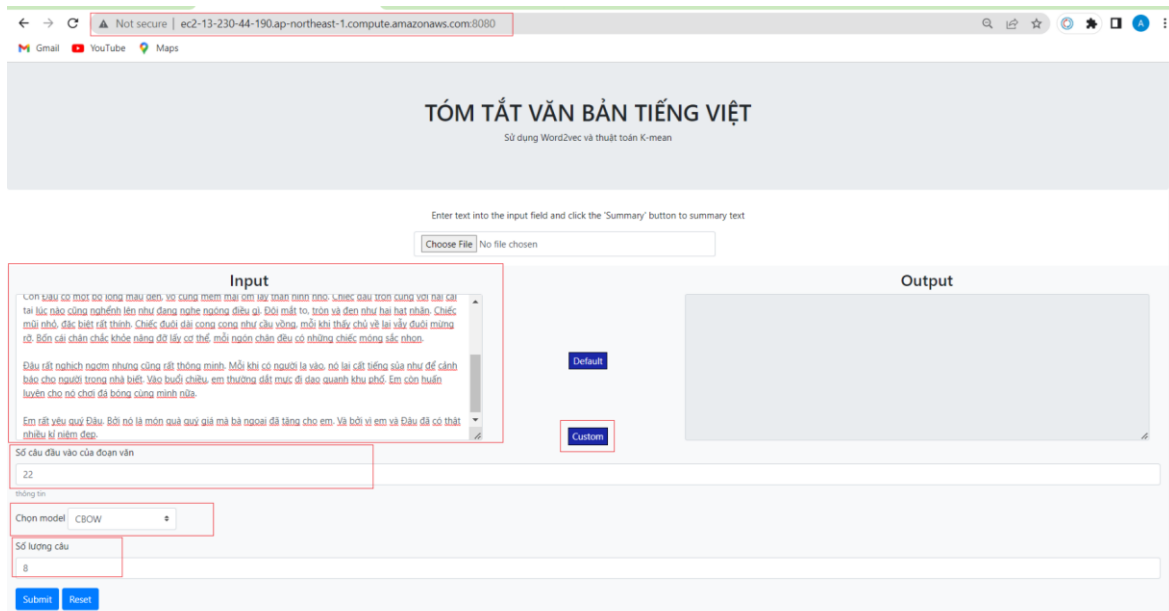
“ba dặn đi dặn lại nhiều lần, với lại ở chỗ thân quen bác ấy mới ưu tiên cho mình con lai này đây, rắng mà nuôi dạy cho kĩ!”. ba bảo: “giống chó này quý lắm con ạ! trong xóm em chỉ một mình cậu mới có bộ áo khoác khác đời, lần vẫn những sọc trắng, nâu, xám y như một con hổ quảng cáo trên màn hình nhỏ vậy. nó tên là lai, cái tên mà em đặt cho nó khi nó còn bé tí tẹo. cái mũi của chú màu nâu đen, có hai cái lỗ nhỏ nhỏ bằng ngón tay út của em, nó đánh hơi cực giỏi. đôi mắt thì trong xanh như màu da trời chứa đựng sự tinh khôn và nhạy cảm của một giống chó béc giê mà các chú công an thường nuôi dạy. không biết trong suốt cả một đêm dài đằng đẳng như thế nó có ngủ được chút nào không. em hy vọng, lai sẽ thật khỏe để có thể sống cùng với gia đình em thật lâu.”

Với demo trên ta nhận được kết với quả độ đo chính xác là 0.8339.



Hình 3.3.3.10 Kết quả độ đo tóm tắt văn bản sử dụng web page với chế độ default.

Chúng tôi tiến hành demo tóm tắt văn bản ở chế độ Custom. Ở chế độ này người dùng sẽ nhập vào input, sau đó nhấn nút custom và nhập và các tham số mong muốn như model và số lượng câu của đoạn văn bản tóm tắt sinh ra như ý muốn.



Hình 3.3.3.11 Demo tóm tắt văn bản sử dụng web page với chế độ custom. Với demo trên, chúng tôi nhập vào input có độ dài của đoạn là 22 câu. Khi chọn chế độ này người dùng cần chọn model và nhập vào số lượng câu và ở đây chúng tôi chọn model CBOW và số lượng câu là 8. Chi tiết input như sau:

“Gia đình em có nuôi một con chó tên là Đậu. Nó đã sống cùng em ba năm rồi, nên đối với em, giống như một người bạn vô cùng thân thiết vậy.

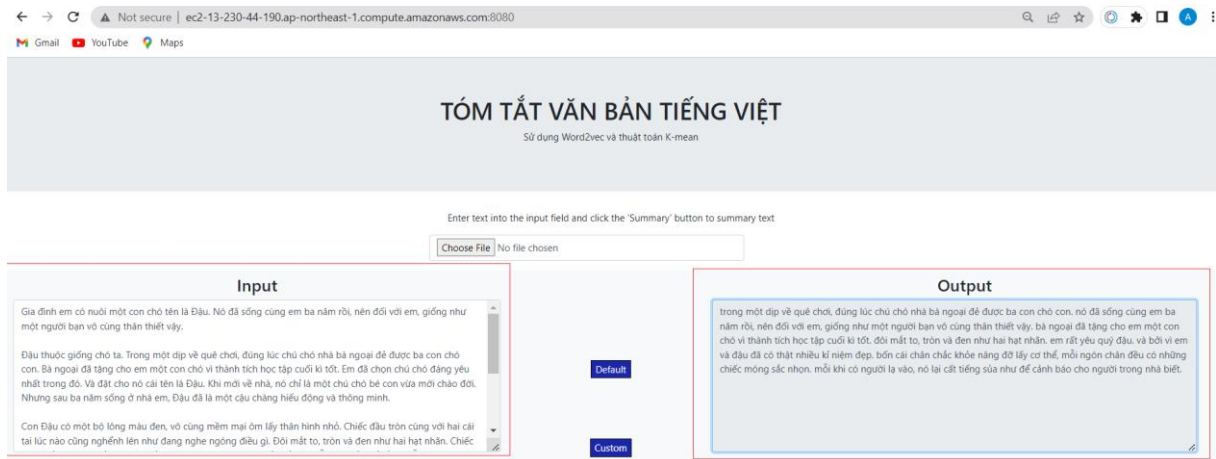
Đậu thuộc giống chó ta. Trong một dịp về quê chơi, đúng lúc chú chó nhà bà ngoại đẻ được ba con chó con. Bà ngoại đã tặng cho em một con chó vì thành tích học tập cuối kì tốt. Em đã chọn chú chó đáng yêu nhất trong đó. Và đặt cho nó cái tên là Đậu. Khi mới về nhà, nó chỉ là một chú chó bé con vừa mới chào đời. Nhưng sau ba năm sống ở nhà em, Đậu đã là một cậu chàng hiếu động và thông minh.

Con Đậu có một bộ lông màu đen, vô cùng mềm mại ôm lấy thân hình nhỏ. Chiếc đầu tròn cùng với hai cái tai lúc nào cũng ngẩng lên như đang nghe ngóng điều gì. Đôi mắt to, tròn và đen như hai hạt nhãn. Chiếc mũi nhỏ, đặc biệt rất thính. Chiếc đuôi dài cong cong như cầu vồng, mỗi khi thấy chủ về lại vẫy đuôi mừng rỡ. Bốn cái chân chắc khỏe nâng đỡ lấy cơ thể, mỗi ngón chân đều có những chiếc móng sắc nhọn.

Đậu rất nghịch ngợm nhưng cũng rất thông minh. Mỗi khi có người lạ vào, nó lại cất tiếng sủa như để cảnh báo cho người trong nhà biết. Vào buổi chiều, em thường dắt mực đi dạo quanh khu phố. Em còn huấn luyện cho nó chơi đá bóng cùng mình nữa.

Em rất yêu quý Đậu. Bởi nó là món quà quý giá mà bà ngoại đã tặng cho em. Và bởi vì em và Đậu đã có thật nhiều kỉ niệm đẹp.”

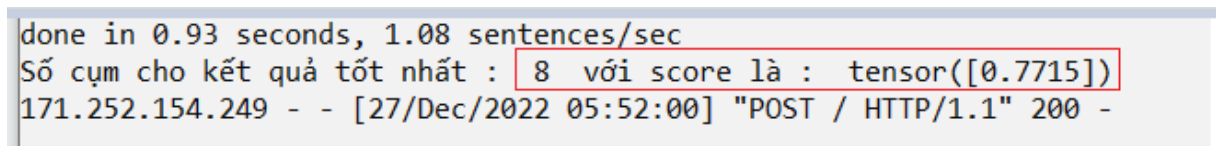
Sau khi nhất nút Submit và chờ trong giây lát, chúng ta sẽ nhận được kết quả được thử hiện như hình bên dưới.



Hình 3.3.3.12 Kết quả tóm tắt văn bản sử dụng web page với chế độ custom. Kết quả của văn bản tóm tắt sẽ được hiển thị ở ô output. Chi tiết của output như sau:

“trong một dịp về quê chơi, đúng lúc chú chó nhà bà ngoại đẻ được ba con chó con. nó đã sống cùng em ba năm rồi, nên đối với em, giống như một người bạn vô cùng thân thiết vậy. bà ngoại đã tặng cho em một con chó vì thành tích học tập cuối kì tốt. đôi mắt to, tròn và đen như hai hạt nhãn. em rất yêu quý đậu. và bởi vì em và đậu đã có thật nhiều kỉ niệm đẹp. bốn cái chân chắc khỏe nâng đỡ lấy cơ thể, mỗi ngón chân đều có những chiếc móng sắc nhọn. mỗi khi có người lạ vào, nó lại cất tiếng sủa như để cảnh báo cho người trong nhà biết.”

Với demo trên ta nhận được kết với quả độ đo chính xác là 0.7715.



Hình 3.3.3.13 Kết quả độ đo tóm tắt văn bản sử dụng web page với chế độ custom.

## **CHƯƠNG 4: KẾT LUẬN**

### **4.1. Kết quả đạt được**

#### **4.1.1. Ý nghĩa khoa học**

Báo cáo đã trình bày các cơ sở lý thuyết về xử lý ngôn ngữ tự nhiên (NLP), kỹ thuật word embedding, lý thuyết và áp dụng demo của bài toán tóm tắt văn bản tiếng việt. Nội dung chính của đề tài là trình bày cơ sở lý thuyết về xử lý ngôn ngữ tự nhiên, word embedding và bài toán tóm tắt văn bản tiếng việt. Ngoài ra còn cho thấy ứng dụng và vai trò của bài toán tóm tắt văn bản. Thông qua đề tài, chúng tôi nắm bắt được những lý thuyết của bài toán xử lý ngôn ngữ tự nhiên, word embedding nói chung và bài toán tóm tắt văn bản nói riêng. Thuật toán và các độ đo áp dụng vào bài toán tóm tắt văn bản. Thông qua ứng dụng demo, chúng tôi đã học thêm được các kiến thức và kỹ năng sử dụng python, các thư viện dùng cho xử lý ngôn ngữ tự nhiên, các kiến thức về lập trình web và deploy ứng dụng web lên aws. Bên cạnh đó, chúng tôi còn nâng cao thêm khả năng đọc hiểu tài liệu, khả năng làm việc nhóm và khả năng trình bày báo cáo khoa học.

#### **4.1.2. Ý nghĩa thực tiễn**

Thông qua tìm hiểu đề tài, chúng tôi biết được thêm các kiến thức về xử lý ngôn ngữ tự nhiên, word embedding, từ đó áp dụng vào bài toán tóm tắt văn bản. Chúng tôi biết được các lợi ích của bài toán tóm tắt văn bản mang lại, đặc biệt là về vấn đề thời gian.

### **4.2. Hạn chế**

Do sự hạn chế về nguồn lực và thời gian, chúng tôi chưa thực sự tìm hiểu sâu và rộng với kỹ thuật word embedding. Đối với bài toán tóm tắt văn bản, chúng tôi đang sử dụng thuật toán phân cụm khá đơn giản nên cho kết quả chưa thực sự cao, chưa tìm hiểu và áp dụng nhiều độ đo của bài toán.

### 4.3. Hướng phát triển

Các kết quả được trình bày trong báo cáo có thể được áp dụng để giải quyết các bài toán về xử lý ngôn ngữ tự nhiên nói chung và đặc biệt là bài toán tóm tắt văn bản. Chúng tôi dự định sẽ tìm hiểu thêm các thuật toán phân cụm tốt hơn và các độ đo cho kết quả chính xác hơn để đưa ra được kết quả tóm tắt tốt nhất.

Về mặt phát triển lâu dài cho ứng dụng demo, chúng tôi dự định sẽ phát triển thành một trang web thương mại, nơi mà tóm tắt các bản tin thành các tin vắn tắt.

## TÀI LIỆU THAM KHẢO

- [1] trituenhantao.io, Word embedding là gì? Tại sao nó quan trọng? 02/04/2019. , truy cập lần cuối ngày 13/11/2022, <https://trituenhantao.io/kien-thuc/word-embedding-la-gi-tai-sao-no-quan-trong/>
- [2] W. guan, "Natural language processing,," wikipedia, 9/11/2022, truy cập lần cuối ngày 27/12/2022, [https://en.wikipedia.org/wiki/Natural\\_language\\_processing](https://en.wikipedia.org/wiki/Natural_language_processing).
- [3] B. Q. Manh, Word Embedding - Tìm hiểu khái niệm cơ bản trong NLP, 14/09/2020, truy cập lần cuối ngày 11/12/2022, <https://viblo.asia/p/word-embedding-tim-hieu-khai-niem-co-ban-trong-nlp-1Je5E93G5nL>
- [4] P. Mishra, Automated metrics for evaluating the quality of text generation, 01/07/2022, truy cập lần cuối ngày 27/12/2022, <https://blog.paperspace.com/automated-metrics-for-evaluating-generated-text/>
- [5] Tianyi Zhang et al, BERTSCORE: EVALUATING TEXT GENERATION WITH BERT, ICRL 2020.
- [6] duyvuleo, VNTEC, 01/01/2019, truy cập lần cuối ngày 27/12/2022, <https://github.com/duyvuleo/VNTEC>.
- [7] Jose Camacho-Collados - Mohammad Taher Pilehvar, Embeddings in Natural Language Processing, Morgan & Claypool publishers , 2020.
- [8] machinelearningcoban, Bài 4: K-means Clustering, 1/1/2017, truy cập lần cuối ngày 17/12/2022, <https://machinelearningcoban.com/2017/01/01/kmeans/>
- [9] N.T.Long, Mô hình CBOW ( continuous bag of words), 10/03/2019, truy cập lần cuối ngày 27/12/2022 , <https://nguyentruonglong.net/mo-hinh-cbow-continuous-bag-of-words.html>
- [10] Michael Struwig, what is a skipgram , 02/01/2019, truy cập lần cuối ngày 26/12/2022 , <https://www.ntsobigdatablog.com/2019/01/02/what-is-a-skipgram/>