

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

— * —

ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC
NGÀNH CÔNG NGHỆ THÔNG TIN

XÂY DỰNG HỆ THỐNG TÌM TẬP TỪ
TIẾNG ANH TỐT NHẤT BIỂU DIỄN
NGHĨA CỦA ĐOẠN MÔ TẢ TIẾNG ANH

Sinh viên thực hiện : **Nguyễn Ngọc Cường**
Lớp CNTT2.02 – K57
Giáo viên hướng dẫn: **ThS Ngô Văn Linh**

HÀ NỘI 05 - 2017

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

Họ và tên sinh viên: Nguyễn Ngọc Cường

Điện thoại liên lạc: 0169354432

Email: ngoccuongbka94@gmail.com

Lớp: CNTT2.02

Hệ đào tạo: Đại học chính quy

Đồ án tốt nghiệp được thực hiện tại: Viện CNTT&TT, Đại học Bách Khoa Hà Nội.

Thời gian làm ĐATN: Từ ngày 31/01/2017 đến ngày 26/05/2017

2. Mục đích nội dung của ĐATN

Xây dựng hệ thống tìm tập từ tiếng Anh tốt nhất biểu diễn nghĩa của đoạn mô tả tiếng Anh.

3. Các nhiệm vụ cụ thể của ĐATN

- Tìm hiểu mạng neural nhân tạo, công cụ word2vec
- Đề xuất phương pháp tính độ tương đồng giữa hai câu tiếng Anh.
- Ứng dụng xây dựng hệ thống tìm tập từ tiếng Anh tốt nhất biểu diễn nghĩa của đoạn mô tả tiếng Anh.
- Xây dựng bộ thử nghiệm và chạy thử nghiệm để kiểm tra độ chính xác của hệ thống.

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Ngọc Cường* - cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Ngô Văn Linh*.

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

Hà Nội, ngày tháng năm

Tác giả ĐATN

Nguyễn Ngọc Cường

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

Hà Nội, ngày tháng năm

Giáo viên hướng dẫn

ThS. Ngô Văn Linh

MỤC LỤC

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	1
TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP	4
ABSTRACT OF THESIS	5
LỜI CẢM ƠN	6
Danh mục hình vẽ	7
Danh mục bảng	8
I. Đặt vấn đề, định hướng giải quyết và các nghiên cứu liên quan	9
1.1. Đặt vấn đề	9
1.2. Định hướng giải pháp	9
1.3. Các nghiên cứu liên quan.....	10
1.3.1. Mạng nơ ron nhân tạo.....	10
1.3.2. Ứng dụng của mạng nơ ron trong xử lý ngôn ngữ tự nhiên	11
II. Cơ sở lý thuyết.....	13
2.1. Công cụ Word2vec	13
2.1.1. Mô hình CBOW (Continuous Bag Of Words).....	13
2.1.2. Mô hình Skip-Gram.....	16
2.1.3. Hierarchical Softmax.....	18
2.1.4. Negative Sampling	20
III. Phân tích và thiết kế hệ thống	21
3.1. Hệ thống đề xuất.....	21
3.1.1. Xây dựng dữ liệu từ điển Anh Anh	22
3.1.2. Thu thập các bộ dữ liệu word2vec đã được training sẵn.....	22
3.1.3. Chức năng tính độ tương đồng giữa hai câu	22
3.2. Các công nghệ được sử dụng.....	24
3.2.1. Angular 2	24
3.2.2. Django Framework	25
3.3. Phân tích hệ thống.....	25
3.3.1. Mô tả hệ thống	25
3.3.2. Mô tả Use Case.....	27
3.3.3. Đặc tả các chức năng của hệ thống.....	29

3.4. Thiết kế hệ thống	35
3.4.1. Biểu đồ trình tự	35
3.2.2. Thiết kế biểu đồ lớp.....	38
3.2.3. Thiết kế cơ sở dữ liệu	50
3.2.4. Thiết kế giao diện	53
IV. Cài đặt và đánh giá hệ thống.....	56
4.1. Cài đặt	56
4.2. Đánh giá hệ thống	56
4.2.1. Môi trường thực nghiệm.....	57
4.2.2. Kết quả đánh giá	57
4.3. Kết quả đạt được	59
V. Kết luận	64

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Đồ án “Xây dựng hệ thống tìm tập từ tiếng Anh biểu diễn tốt nhất đoạn mô tả tiếng Anh” giúp khắc phục được vấn đề suy nghĩ bằng tiếng mẹ đẻ khi sử dụng tiếng Anh của những người không phải người Anh bản xứ. Chức năng chính của hệ thống là khi người dùng muốn sử dụng một từ nào đó mà người dùng không biết trong tiếng Anh, nhưng biết nghĩa của nó, biết mô tả nó như thế nào, thì người dùng có thể mô tả từ đó và hệ thống sẽ đưa ra tập từ biểu diễn nghĩa tốt nhất của đoạn mô tả đó, từ đây người dùng có thể xem nghĩa của những từ trong tập từ hệ thống đưa ra và chọn ra từ nào phù hợp với ngữ cảnh mà người dùng muốn nhất. Người dùng có thể đánh giá từ nào trong tập từ phù hợp với đoạn mô tả, hệ thống sẽ dựa vào đánh giá của người dùng để đưa ra kết quả tốt hơn trong các lần tìm kiếm sau.

Đồ án đã đi sâu vào tìm hiểu mạng neural nhân tạo, các thuật toán, mô hình bên trong công cụ word2vec, một công cụ do Google nghiên cứu, chuyển mỗi từ thành một vector, có ý nghĩa rất lớn trong lĩnh vực xử lý ngôn ngữ tự nhiên và đã đạt được những kết quả rất ấn tượng.

Đồ án cũng đề xuất hai phương pháp dựa vào word2vec để tính độ tương đồng giữa hai câu tiếng Anh. Bên cạnh đó cũng đã triển khai cài đặt hai phương pháp này.

Cuối cùng đồ án mô tả các kết quả đánh giá hệ thống dựa trên các dữ liệu thử nghiệm đã được xây dựng bởi người dùng thật. Từ đó đưa ra các ưu, nhược điểm của hệ thống và các hướng phát triển trong tương lai để hệ thống có thể đi vào hoạt động phục vụ cộng đồng.

Bộ cục của đồ án gồm 5 phần chính:

- Phần 1 nêu tổng quan về hệ thống, đặt vấn đề, định hướng giải pháp và các nghiên cứu liên quan
- Phần 2 nêu rõ cơ sở lý thuyết về các thuật toán, mô hình bên trong công cụ word2vec và hai phương pháp đề xuất cho hệ thống để tính độ tương đồng giữa hai câu tiếng Anh.
- Phần 3 là phần phân tích và thiết kế hệ thống
- Phần 4 là phần cài đặt, đánh giá hệ thống.
- Phần 5 là phần kết luận, tổng kết các ưu nhược điểm của hệ thống, đưa ra các hướng phát triển trong tương lai.

ABSTRACT OF THESIS

Thesis “Build an application to find the best words set which represent the meaning of the description in English” helps to change a bad habit of most of people who are not English native speaker that is thinking in native mother tongue when using English. The main functionality of this application is to give user the best words when they give the description of the word they want to express which they do not know in English. And the user can see the meaning of each word in words set to choose the best word that is suitable in the context. And they can also give the feedback to improve the search system for the next query.

Thesis researches on word2vec, related works and proposes two methods to calculate the similarity of two English sentences and implements those as well.

Finally, Thesis describes the results which were tested on application based on the test data which was created by the real users. And then Thesis gives the advantages, disadvantages of the application and the solutions to develop the application in the future as well.

Thesis has 5 main sections:

- Section 1: The Summary of application and related work
- Section 2: Neural network, Algorithms, models in Word2vec, describes 2 methods to calculate the similarity of two English sentences.
- Section 3: System analysis and design
- Section 4: Implementing and Experiment System
- Section 5: Conclusion, advantages, disadvantages and solutions to improve the application in the future.

LỜI CẢM ƠN

Đầu tiên, tôi xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới ThS. Ngô Văn Linh, người đã tận tình hướng dẫn, giúp đỡ, động viên tôi trong suốt quá trình thực hiện đồ án này, giúp tôi hoàn thành tốt đồ án này.

Đồng thời, tôi xin bày tỏ lòng biết ơn đến các thầy cô trong Viện Công Nghệ Thông Tin đã giảng dạy, truyền đạt cho tôi những kiến thức cơ bản, làm nền tảng cho việc thực hiện đồ án này.

Tôi cũng xin gửi lời cảm ơn tới chị em, bạn bè tôi đã luôn động viên giúp đỡ tôi trong quá trình thực hiện đồ án.

Cuối cùng, tôi xin cảm ơn chân thành đến gia đình tôi, một nơi luôn làm chỗ dựa tinh thần, tạo mọi điều kiện tốt nhất để tôi thực hiện tốt đồ án này.

Hà Nội, ngày tháng năm 2017

Sinh viên

Nguyễn Ngọc Cường

Danh mục hình vẽ

Hình 1.	Mô hình mạng neural nhân tạo	11
Hình 2.	Mô hình Continuous Bag of Words.....	14
Hình 3.	Mô hình Skip-Gram.....	16
Hình 4.	Cây nhị phân	18
Hình 5.	Mô hình hệ thống.....	21
Hình 6.	Biểu diễn cách tính độ tương đồng 2 câu theo phương pháp 2	23
Hình 7.	Kiến trúc Angular 2	24
Hình 8.	Biểu đồ usecase tổng quát	27
Hình 9.	Biểu đồ use case tác nhân User	28
Hình 10.	Biểu đồ use case Admin	28
Hình 11.	Biểu đồ trình tự đăng nhập	35
Hình 12.	Biểu đồ trình tự tìm nghĩa của từ.....	35
Hình 13.	Biểu đồ trình tự tìm tập từ dựa vào mô tả.....	36
Hình 14.	Biểu đồ trình tự chấp thuận các cặp từ và đoạn mô tả từ đánh giá của người dùng	36
Hình 15.	Biểu đồ trình tự crawl dữ liệu từ điển cho hệ thống	37
Hình 16.	Biểu đồ lớp tìm nghĩa của từ	38
Hình 17.	Biểu đồ lớp tìm tập từ dựa vào mô tả	42
Hình 18.	Biểu đồ lớp cho chức năng đăng nhập.....	45
Hình 19.	Biểu đồ lớp cho các chức năng của Admin	47
Hình 20.	Mô hình liên hệ giữa các bảng trong cơ sở dữ liệu của hệ thống.....	52
Hình 21.	Mô hình liên hệ giữa các bảng lưu trữ dữ liệu từ điển Anh Anh của hệ thống	53
Hình 22.	Thiết kế giao diện trang chủ	53
Hình 23.	Thiết kế giao diện trang tìm kiếm bằng mô tả.....	54
Hình 24.	Thiết kế giao diện trang tìm kiếm bằng từ.....	54
Hình 25.	Thiết kế giao diện trang admin chấp thuận đánh giá của người dùng	55
Hình 26.	Thiết kế giao diện admin chức năng crawling.....	55
Hình 27.	Thiết kế giao diện trang admin cập nhật hệ thống.....	56
Hình 28.	Ví dụ của một từ trong bộ dữ liệu word2vec training sẵn	56
Hình 29.	Bộ dữ liệu để thực nghiệm.....	57
Hình 30.	Giao diện trang chủ hệ thống.....	59
Hình 31.	Giao diện trang tìm kiếm	60
Hình 32.	Giao diện chức năng tìm kiếm bằng mô tả	60
Hình 33.	Giao diện chức năng hiển thị nghĩa của từ khi người dùng click vào từ được trả về	60
Hình 34.	Màn hình hiển thị chức năng tìm kiếm nghĩa của từ	61
Hình 35.	Giao diện chức năng chia sẻ mạng xã hội	61
Hình 36.	Giao diện chức năng đăng nhập.....	62
Hình 37.	Giao diện admin chức năng chấp thuận đánh giá của người dùng	62
Hình 38.	Giao diện chức năng crawling của admin	63
Hình 39.	Giao diện chức năng hiển thị quá trình crawling.....	63
Hình 40.	Giao diện hiển thị chức năng cập nhật hệ thống.....	63

Danh mục bảng

Bảng 1.	Đặc tả use case UC01 « Đăng nhập ».....	30
Bảng 2.	Mô tả dữ liệu đầu vào khi đăng nhập	30
Bảng 3.	Đặc tả use case UC02 « Tìm nghĩa của từ ».....	30
Bảng 4.	Đặc tả use case UC03 « Tìm tập từ bằng mô tả »	31
Bảng 5.	Đặc tả use case UC04 « Nghe phát âm của từ».....	32
Bảng 6.	Đặc tả use case UC05 « Crawl từ điển Anh Anh »	32
Bảng 7.	Đặc tả use case UC06 « Xem danh sách cặp từ và đoạn mô tả được đánh giá»	33
Bảng 8.	Đặc tả use case UC07 « Chấp thuận các cặp từ và đoạn mô tả của người dùng »	33
Bảng 9.	Đặc tả use case UC08 « Cập nhật hệ thống tìm kiếm »	34
Bảng 10.	Đặc tả use case UC09 « Đăng xuất ».....	34
Bảng 11.	Mô tả bảng Word trong cơ sở dữ liệu.....	50
Bảng 12.	Mô tả bảng Definition trong cơ sở dữ liệu	50
Bảng 13.	Mô tả bảng Example trong cơ sở dữ liệu.....	50
Bảng 14.	Mô tả bảng Phonetic trong cơ sở dữ liệu.....	51
Bảng 15.	Mô tả bảng auth_user trong cơ sở dữ liệu	51
Bảng 16.	Kết quả đánh giá từng bộ dữ liệu word2vec training sẵn, kích thước context bằng 10 với phương pháp 1	58
Bảng 17.	Kết quả đánh giá từng bộ dữ liệu word2vec training sẵn, kích thước context bằng 10 với phương pháp 2	59

Danh mục thuật ngữ

Thuật ngữ, từ viết tắt	Tên đầy đủ, ý nghĩa
crawl	Một kỹ thuật thu thập dữ liệu trên nền tảng web.
RESTful API	Viết tắt của từ Representation State Transfer, ful chỉ là một suffix trong tiếng Anh, là một kiểu kiến trúc được sử dụng trong việc giao tiếp giữa các máy tính trong việc quản lý tài nguyên trên internet [15]

I. Đặt vấn đề, định hướng giải quyết và các nghiên cứu liên quan

1.1. Đặt vấn đề

Trong thời đại ngày nay, khi mà toàn cầu hóa, hội nhập quốc tế đang diễn ra rất nhanh, ai cũng phải trang bị cho mình một hành trang để bắt kịp, tiếp cận với cơ hội, tri thức thế giới. Trong đó việc học tiếng Anh là một điều tất yếu khó tránh khỏi. Thêm nữa, thế giới công nghệ phát triển bùng nổ như hiện nay, con người càng ngày càng ứng dụng các công nghệ để giải quyết các vấn đề một cách hiệu quả nhanh chóng. Điều này cũng đúng trong việc học tiếng Anh, một trong những ngôn ngữ thông dụng nhất thế giới, được rất nhiều người không phải người Anh bản xứ theo học. Tuy nhiên việc học không phải dễ, gặp nhiều khó khăn rào cản, nhiều người đã bỏ cuộc hoặc không đạt được đến trình độ nâng cao trong tiếng Anh, vì vậy việc áp dụng công nghệ vào việc học Tiếng Anh được nhiều người nghiên cứu, các ý tưởng mới ra đời và được thực hiện. Một người có thể học tốt tiếng Anh không cần đến sự trợ giúp của công nghệ nhưng điều đó cần nhiều công sức, thời gian, sự kiên trì và không phải ai cũng làm được điều đó. Vì vậy có sự trợ giúp của công nghệ sẽ giúp phần nào giải quyết các rào cản cho người học. Trong quá trình học tiếng Anh tôi nhận thấy có một rào cản lớn không chỉ với tôi mà còn với hầu hết người học tiếng Anh mà tiếng Anh không phải tiếng mẹ đẻ, đó là rào cản suy nghĩ bằng tiếng mẹ đẻ khi sử dụng tiếng Anh. Đây là một thói quen khó bỏ khi học một ngôn ngữ mới. Để giải quyết thói quen này thì cần luyện tập suy nghĩ bằng tiếng Anh, có thể ví dụ như dùng từ điển Anh Anh để tra từ, hiểu nghĩa của từ bằng tiếng Anh, từ đó khi nói, viết hoặc đọc sẽ không cần phải suy nghĩ từ tiếng mẹ đẻ nữa mà sử dụng tiếng Anh luôn vì câu từ tiếng Anh đã được suy nghĩ rồi. Nhưng việc suy nghĩ bằng tiếng Anh không phải dễ, không phải khi nào cũng nhớ được hết từ đã học, hoặc có nhiều từ muốn diễn đạt nhưng không biết những từ đó trong tiếng Anh mặc dù biết nghĩa của những từ đó như thế nào, mô tả như thế nào, mà việc tra từ tiếng mẹ đẻ sang tiếng Anh thì sẽ phá vỡ việc muốn suy nghĩ bằng tiếng Anh, dễ bị suy nghĩ bằng tiếng mẹ đẻ khi sử dụng tiếng Anh. Từ đây mà ý tưởng xây dựng hệ thống này ra đời. Hệ thống này sẽ giúp giải quyết vấn đề này, giúp người dùng tìm ra từ mà người dùng muốn khi mô tả nghĩa của nó bằng tiếng Anh. Nhận thấy chưa có hệ thống nào giúp giải quyết vấn đề này nên hệ thống này rất có ý nghĩa thực tiễn. Nhưng việc tìm ra từ biểu diễn nghĩa của đoạn mô tả không phải dễ vì đây là máy tính không phải người thật. Việc dạy cho máy tính không chỉ hiểu được cú pháp của một ngôn ngữ mà còn ngữ nghĩa của các câu sử dụng trong từng ngữ cảnh đặc biệt mà không mắc phải nháp nhằng là một bài toán khó và phức tạp. Vậy vấn đề đặt ra là làm sao để máy tính hiểu nghĩa của đoạn mô tả trong tiếng Anh để từ đó tìm từ biểu diễn nghĩa của đoạn mô tả này. Từ đó đưa ra tập từ biểu diễn nghĩa tốt nhất cho đoạn mô tả được mô tả bởi người dùng.

1.2. Định hướng giải pháp

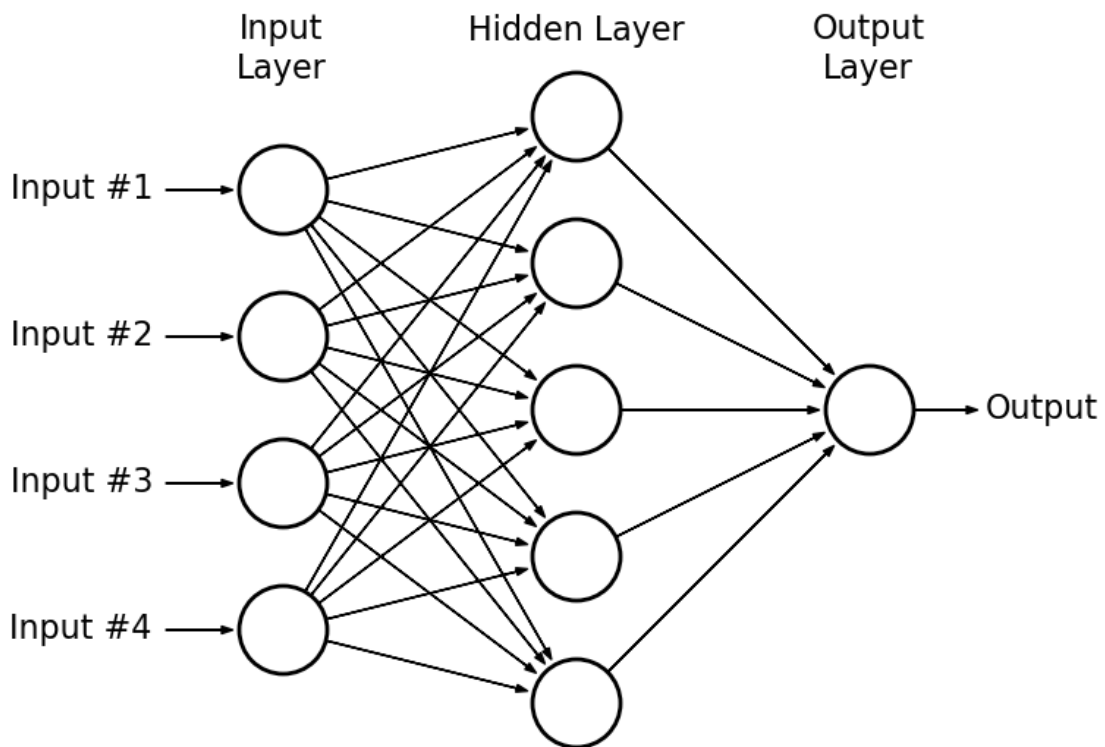
Với vấn đề được đặt ra là làm sao để máy tính hiểu được nghĩa của đoạn mô tả trong tiếng Anh, nhiều nghiên cứu cho thấy với sự phát triển của mạng neural trong lĩnh vực xử lý ngôn ngữ tự nhiên, một vài khoảng cách tri thức giữa con người và máy tính dần được xóa bỏ, khó khăn phần nào được giải quyết. Trong nghiên cứu

này, hệ thống sẽ tìm ra tập từ biểu diễn nghĩa tốt nhất từ một đoạn mô tả bởi người dùng cung cấp bằng cách tìm ra các từ có nghĩa tương đồng với đoạn mô tả. Mỗi từ tiếng Anh đều có nghĩa của nó được lưu trữ trong một từ điển (cụ thể đây là dữ liệu từ điển Oxford Anh Anh). Vì vậy chỉ cần tìm ra nghĩa của từ nào tương đồng với đoạn mô tả thì sẽ giải quyết được bài toán này. Để biểu diễn được từ ngữ làm sao cho máy tính hiểu được thì ta sẽ chuyển từ thành vector, có nhiều cách biểu diễn từ bởi vector trong đó có one hot vector và word embedding (word2vec), one hot vector có số chiều bằng kích thước từ điển sẽ không có lợi có việc tính toán, và one hot vector không biểu diễn được sự tương đồng ngữ nghĩa của các từ nhưng với công cụ word2vec, một công cụ được Tomas Mikolov[1] và các cộng sự tại google nghiên cứu khắc phục được các yếu điểm trên của one hot vector, word2vec có nhiều ứng dụng quan trọng trong xử lý ngôn ngữ tự nhiên và đã đạt được những kết quả rất ấn tượng. Với cách biểu diễn từ (word) thành vector bởi word2vec như vậy, các vector mang lại cả cú pháp và ngữ nghĩa ở một mức độ nào đó, một vài ví dụ kinh điển mà word2vec đã làm được là “ $\text{vec}(\text{king}) - \text{vec}(\text{man}) + \text{vec}(\text{woman}) = \text{vec}(\text{queen})$ ”, “ $\text{vec}(\text{Paris}) - \text{vec}(\text{France}) + \text{vec}(\text{Germany}) = \text{vec}(\text{Berlin})$ ”[1]. Word2vec có thể tính được độ tương đồng của hai từ bằng cách thực hiện các phép tính trên các vector của chúng. Vì vậy việc chọn để áp dụng công cụ word2vec vào hệ thống của tôi là rất khả quan.

1.3. Các nghiên cứu liên quan

1.3.1. Mạng nơ ron nhân tạo

Mạng nơ ron nhân tạo là mô hình xử lý thông tin được mô phỏng dựa trên mạng nơ ron sinh học, gồm một nhóm các nơ ron nhân tạo (nút) liên kết với nhau theo một cấu trúc. Các nhà khoa học máy tính đã có một thời gian dài được truyền cảm hứng bởi bộ não người. Năm 1943, Warren S. McCulloch và Walter Pitts đã phát triển một mô hình khái niệm đầu tiên về mạng neural nhân tạo. Trong bài báo “A logical calculus of the ideas imminent in nervous activity” họ mô tả khái niệm của một nơ ron, một tế bào đơn trong một mạng các tế bào nhận dữ liệu đầu vào, xử lý và sinh ra một giá trị đầu ra. Mục đích công việc của họ và các nhà nghiên cứu và nhà khoa học sau này không phải là để mô tả cách thức làm việc của bộ não mà là nghiên cứu về mạng nơ ron nhân tạo được thiết kế như một mô hình tính toán dựa trên bộ não để giải quyết các vấn đề cụ thể như nhận dạng mẫu, phân loại, dịch máy, xử lý tín hiệu, xử lý ảnh và computer vision...



Hình 1. Mô hình mạng neural nhân tạo [12]

Mạng nơ ron nhân tạo gồm nhiều tầng, tầng đầu vào, các tầng ẩn và tầng đầu ra, mỗi tầng gồm nhiều nơ ron, các nơ ron được kết nối với nhau, mỗi kết nối có một vector trọng số, giá trị đầu vào tổng thể của một nơ ron sẽ được tính bởi một hàm tích hợp của các tín hiệu đầu vào và vector trọng số, giá trị đầu ra của một nơ ron sẽ được tính bởi một hàm tác động (activation function) của giá trị đầu vào tổng thể của nơ ron. Mạng nơ ron có khả năng tự học, là một hệ thống tự thích nghi nghĩa là nó sẽ tự điều chỉnh cấu trúc mạng bên trong hoặc các trọng số của các liên kết trong mạng nơ ron dựa vào thông tin được đưa vào mạng. Nếu mạng sinh ra một giá trị đầu ra tốt, các trọng số sẽ không cần phải điều chỉnh, ngược lại nó lan truyền ngược giá trị lỗi và điều chỉnh giá trị trọng số để cải thiện giá trị đầu ra lần sau và cứ như thế qua nhiều lần học ví dụ học, tất cả giá trị trọng số giữa các kết nối sẽ được điều chỉnh.

1.3.2. Ứng dụng của mạng nơ ron trong xử lý ngôn ngữ tự nhiên

Ứng dụng của mạng nơ ron nhân tạo là rất lớn trong nhiều lĩnh vực, trong đó có lĩnh vực xử lý ngôn ngữ tự nhiên, nó đã được áp dụng trong nhiều bài toán xử lý ngôn ngữ tự nhiên và đạt được thành công như bài toán rút trích thông tin chủ quan (sentiment analysis) của người dùng từ một câu (Socher et al., 2011) hay từ một văn bản (Glorot et al., 2011), language modeling (Mnih and Hinton, 2007; Mikolov and Zweig, 2012), phát hiện cụm từ (Socher et al., 2011a) và những bài toán liên quan đến mối quan hệ ngữ nghĩa và nghĩa thành phần của các cụm từ (Socher et al., 2012) [2]. Điểm chung của các bài toán này là sử dụng biểu diễn phân phối từ (distributed word representation) như một đơn vị đầu vào cơ bản. Biểu diễn từ như

các vector liên tục có một lịch sử dài[1]. Một mô hình kiến trúc rất phổ biến để ước lượng mô hình ngôn ngữ mạng nơ ron (neural network language model – NNLM) được đề xuất trong [3], trong đó một mạng nơ ron lan truyền tiến với lớp chiếu tuyến tính và một lớp ẩn không tuyến tính (none-linear) được sử dụng để học cùng với sự biểu diễn vector từ và một mô hình ngôn ngữ thống kê. Công việc này đã được tiếp tục nghiên cứu bởi các nhà nghiên cứu sau này. Một mô hình kiến trúc thú vị khác của mô hình ngôn ngữ mạng nơ ron được trình bày trong [4], [5], trong đó các vector từ ban đầu được học sử dụng mạng nơ ron với một tầng ẩn, sau đó được dùng để huấn luyện mô hình ngôn ngữ mạng nơ ron, vì vậy các vector từ được học ngay cả khi không xây dựng mô hình ngôn ngữ mạng nơ ron đầy đủ. Trong [1] Tomas Mikolov và các cộng sự đã mở rộng kiến trúc này và chỉ tập trung vào bước đầu tiên xây dựng các vector từ từ một mô hình đơn giản. Có nhiều mô hình đã được đề xuất để ước tính biểu diễn liên tục của các từ gồm Latent Semantic Analysis (LSA) và Latent Dirichlet Allocation (LDA). Trong [1], Tomas Mikolov tập trung vào biểu diễn phân phối của các từ học bởi mạng nơ ron nhân tạo vì nó được cho thấy thực hiện tốt hơn đáng kể so với LSA về duy trì tính bình thường tuyến tính giữa các từ [6],[7], hơn nữa LDA thì tính toán rất tốn kém trên bộ dữ liệu lớn.

Biểu diễn phân phối của các từ trong không gian vector giúp các thuật toán học đạt được hiệu suất tốt hơn trong các bài toán xử lý ngôn ngữ tự nhiên bằng cách nhóm các từ tương tự nhau về ngữ nghĩa hay cú pháp. Biểu diễn từ được tính toán bởi mạng nơ ron là rất thú vị vì nó biểu diễn từ thành các vector nhưng vẫn giữ được tính ngữ nghĩa và cú pháp giữa các từ, các từ tương tự nhau về cú pháp và ngữ nghĩa được biểu diễn bởi các vector gần nhau, các vector được học xong có mối quan hệ tuyến tính đáng chú ý ví dụ như kết quả của phép tính $\text{vec}(\text{"Madrid"}) - \text{vec}(\text{"Spain"}) + \text{vec}(\text{"France"})$ là một vector gần với $\text{vec}(\text{"Paris"})$ hơn là các vector khác.

Nhiều thuật toán học máy (machine learning) cần dữ liệu đầu vào được biểu diễn như một vector đặc trưng với chiều dài cố định. Khi đầu vào là một câu, đoạn văn hay văn bản, bag-of-words được sử dụng rất phổ biến để biến đầu vào đó thành vector vì sự đơn giản, hiệu quả và thường khá chính xác. Nhưng bag-of-words có hai điểm yếu, một là mất trật tự của từ trong câu ví dụ hai câu mặc dù mang ý nghĩa khác nhau nhưng chỉ cần có cùng các từ thì chúng vẫn được biểu diễn giống nhau, hai là bỏ qua ngữ nghĩa của các từ ví dụ “powerful”, “strong”, “Paris” có cùng khoảng cách xa nhau, “powerful” nên gần “strong” hơn là “Paris”. Một cách tiếp cận đơn giản sử dụng word2vec để xây dựng vector cho câu, đoạn hay văn bản đó là tính trung bình tất cả các vector của các từ trong câu, đoạn hay văn bản nhưng cách này gặp một điểm yếu giống như mô hình bag-of-words đó là mất trật tự của từ trong câu, mặc dù như vậy nhưng nó cũng đạt được kết quả khá khả quan trong nhiều bài toán không quan tâm đến trật tự từ trong câu. Một cách tiếp cận phức tạp hơn là kết hợp các vector từ theo thứ tự được đưa ra bởi phân tích cú pháp của một câu, sử dụng các phép toán ma trận vector. Cách tiếp cận này chỉ hiệu quả với câu vì nó phụ thuộc vào việc phân tích cú pháp (Socher et al., 2011b). Trong [8], Quoc Le và Tomas Mikolov đề xuất Paragraph Vector, một thuật toán không giám

sát để học ra vector biểu diễn phân phối liên tục cho câu, đoạn hay văn bản. Ý tưởng xây dựng Paragraph Vector giống với word2vec (Mikolov et al., 2013). Trong mô hình Paragraph vector, vector biểu diễn được huấn luyện để dự đoán các từ trong một đoạn văn, cụ thể hơn họ nối các paragraph vector với nhiều word vectors từ một đoạn văn và dự đoán từ theo sau trong phạm vi context cho trước. Cả word vectors và paragraph vectors đều được huấn luyện bằng việc sử dụng stochastic gradient descent và kỹ thuật lan truyền ngược lỗi (backpropagation) (Rumelhart et al, 1986). Paragraph vector là riêng biệt giữa các đoạn văn nhưng words vector được chia sẻ giữa các đoạn văn, ví dụ vector của từ “powerful” giống nhau giữa các đoạn văn. Trong quá trình huấn luyện, dần dần các paragraph vector được suy ra bằng việc cố định các word vectors và đào tạo các paragraph vector mới cho đến khi hội tụ (lỗi trong quá trình học chấp nhận được). Kết quả kiểm nghiệm chỉ ra rằng Paragraph Vector khắc phục được các yếu điểm, thậm chí thực hiện tốt hơn với mô hình bag-of-words cũng như các kỹ thuật biểu diễn cho câu, đoạn, hay văn bản khác. Đạt kết quả ấn tượng trong các bài toán phân loại text và rút trích thông tin chủ quan (sentiment analysis) của người dùng từ câu, đoạn hay văn bản.

II. Cơ sở lý thuyết

2.1. Công cụ Word2vec

Word2vec là một công cụ để học ra các vector cho các từ. Các mô hình trong Word2vec sử dụng một mạng nơ ron đơn giản với duy nhất một tầng ẩn (hidden layer) để học ra các vector biểu diễn cho các từ trong bộ từ vựng. Các vector này thực ra là các cột trong ma trận trọng số giữa tầng ẩn và tầng đầu ra của mạng nơ ron. Word2vec có hai kiểu mô hình chính là CBOW (Continuous Bag Of Words) và Skip-gram. CBOW được huấn luyện để dự đoán một từ từ một context (gồm các từ xung quanh từ đó). Còn Skip-gram thì ngược lại được huấn luyện để dự đoán context (gồm các từ) từ một từ (các từ trong context nằm xung quanh từ này) [1],[9]. Mỗi mô hình có hai phương thức học (training method) khác nhau là negative sampling và hierarchical softmax.

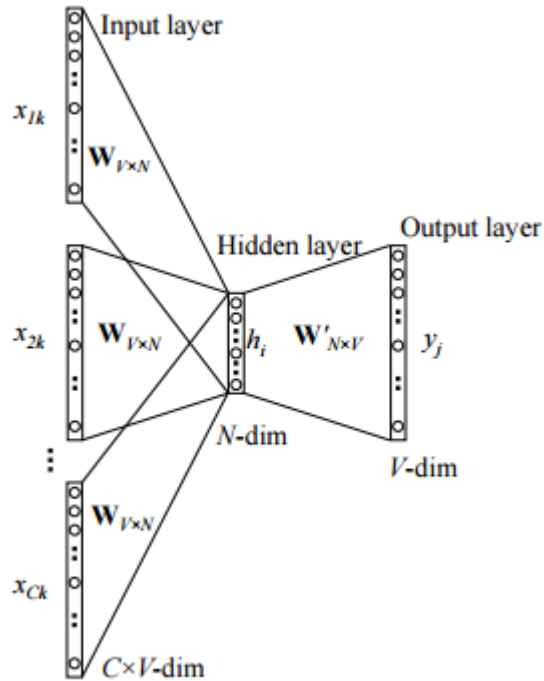
2.1.1. Mô hình CBOW (Continuous Bag Of Words)

Mô hình CBOW sử dụng mạng nơ ron được biểu diễn như hình 2 với context gồm C từ (các từ này nằm xung quanh từ được dự đoán theo kích thước của sổ, ví dụ cửa sổ = 2 thì context sẽ có 4 từ, 2 từ bên trái và 2 từ bên phải từ được dự đoán), mỗi từ được biểu diễn bằng one hot vector \mathbf{x} có số chiều là V (kích thước bộ từ vựng được đưa vào huấn luyện), nghĩa là trong vector chỉ có duy nhất một phần tử có giá trị 1 đó là vị trí của từ trong bộ từ vựng, còn lại có giá trị 0. Ví dụ bộ từ vựng cho huấn luyện có 10 từ được sắp xếp theo alphabet trong đó có từ “cat” ở vị trí thứ 3, thì từ “cat” sẽ có one hot vector là $(0,0,1,0,0,0,0,0,0,0)$. Tầng ẩn gồm N nơ ron, có thể biểu diễn là một vector \mathbf{h} có N chiều, mỗi nút có một giá trị đầu ra h_i . Các trọng số giữa tầng đầu vào và tầng ẩn được biểu diễn bởi ma trận $W_{V \times N}$. Mỗi dòng của ma trận W là vector v_w có N chiều biểu diễn từ tương ứng tại tầng đầu vào. Để

tính giá trị đầu ra cho tầng ẩn, CBOW lấy trung bình các vector của các từ trong context nhân với ma trận trọng số \mathbf{W} .

$$h = \frac{1}{C} W^T (x_1 + x_2 + \dots + x_C) \quad (1)$$

$$= \frac{1}{C} (v_{w_1} + v_{w_2} + \dots + v_{w_C})^T \quad (2)$$



Hình 2. Mô hình Continuous Bag of Words [10]

Các trọng số từ tầng ẩn và tầng đầu ra được biểu diễn bởi ma trận $W'_{N \times V}$. Đầu vào cho mỗi nút j (tương ứng với mỗi từ trong bộ từ vựng) tại tầng đầu ra (output layer) được tính như sau:

$$u_j = v'_{w_j}{}^T h \quad (3)$$

Trong đó v'_{w_j} là cột thứ j của ma trận W' (gọi là vector đầu ra của từ w_j), h vector tầng ẩn, được tính bởi công thức (2).

Lưu ý rằng v_w là vector đầu vào của từ w (là dòng của ma trận W), v'_w là vector đầu ra của từ w (là cột của ma trận W').

Vậy xác suất có điều kiện để từ được dự đoán đúng xuất hiện giữa các từ trong context là:

$$p(w_o | w_{l,1}, \dots, w_{l,C}) = y_{j^*} = \frac{\exp(u_{j^*})}{\sum_j \exp(u_j)} \quad (4)$$

Trong đó y_{j^*} là giá trị đầu ra của nút thứ j^* mà tại đó từ được dự đoán đúng, là từ đầu ra thật sự.

Mục tiêu huấn luyện là làm cho y_{j^*} hay $p(w_o | w_{l,1}, \dots, w_{l,C})$ đạt giá trị lớn nhất. Sử dụng log vào $p(w_o | w_{l,1}, \dots, w_{l,C})$ ta được hàm mất mát, mục đích là phải

làm hàm mất mát này nhỏ nhất:

$$E = -p \cdot w_O \cdot w_{I,1}, \dots, w_{I,C} = -u_{j^*} + \log \prod_{j'=1}^V \exp(u_{j'}) \quad (5)$$

Sử dụng gradient descent và kỹ thuật lan truyền ngược để cập nhật ma trận trọng số W' giữa tầng ẩn và tầng đầu ra [10].

$$v'_{w_j}{}^{(new)} = v'_{w_j}{}^{(old)} - \eta \cdot e_j \cdot h \quad \text{for } j = 1, 2, \dots, V. \quad (6)$$

Trong đó $\eta > 0$ là tốc độ học, $e_j = y_j - t_j$ là lỗi dự đoán, y_j là giá trị đầu ra tại nút j ở tầng đầu ra, $t_j = 1$ nếu nút thứ j là từ đầu ra thật sự còn không thì $t_j = 0$.

Từ (6) có thể thấy, mô hình kiểm tra xác suất đầu ra y_j và giá trị thật sự t_j (0 hoặc 1) của tất cả các từ trong bộ từ vựng. Nếu từ w_j được đánh giá quá cao ($y_j > t_j$) thì vector của từ w_j sẽ bị trừ đi một phần của vector h , vì thế vector v'_{w_j} của từ w_j sẽ dịch chuyển cách xa hơn vector trung bình của các từ đầu vào h trong context. Còn nếu $y_j < t_j$ nghĩa là $t_j = 1$, từ vị trí thứ j trong bộ từ vựng chính là từ được dự đoán đúng, thì vector v'_{w_j} sẽ dịch chuyển gần hơn vector trung bình của các từ đầu vào h trong context. Và khi y_j có giá trị rất gần với t_j thì vector v'_{w_j} sẽ chỉ thay đổi rất nhỏ [10].

Cập nhật ma trận trọng số giữa tầng đầu vào và tầng ẩn.

Theo [10]:
$$v_{w_{I,c}}{}^{(new)} = v_{w_{I,c}}{}^{(old)} - \frac{1}{C} \cdot \eta \cdot EH^T \quad (7)$$

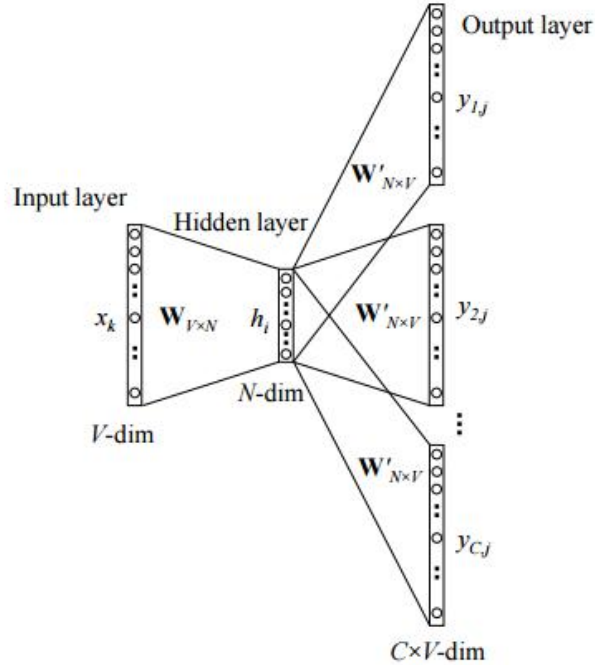
Trong đó EH là vector N chiều là tổng tất cả các tích của vector đầu ra (cột của ma trận W') của tất cả các từ trong bộ từ vựng với lỗi dự đoán của từng từ, theo [10]:

$$EH = \sum_j v'_{w_j} \cdot e_j \quad (8)$$

Vì các vector của các từ trong context C là one hot vector nên chỉ có các vector $v_{w_{I,c}}$ tương ứng với từ $w_{I,c}$ trong ma trận W này được cập nhật, còn lại giữ nguyên (I là vị trí tương ứng của từ trong ma trận W , c là vị trí từ trong context C).

2.1.2. Mô hình Skip-Gram

Mô hình Skip-Gram được giới thiệu ở trong [1][9]. Mô hình này ngược với mô hình CBOW được mô tả như hình 3.



Hình 3. Mô hình Skip-Gram [10]

Từ được dự đoán bây giờ là từ tại tầng đầu vào, còn các từ trong context ở tầng đầu ra, nghĩa là các từ được dự đoán là các từ trong context, còn từ đầu vào là từ w_I . Gọi v_{w_I} là vector đầu vào của từ w_I (cũng là dòng của ma trận trọng số $W_{V \times N}$ của tầng đầu vào và tầng ẩn). Giá trị đầu ra của tầng ẩn là

$$h = W^T x = v_{w_I}^T \quad (9)$$

Tương tự như ở CBOW vì x là one hot vector nên chỉ có một giá trị là 1 nên khi nhân với W^T sẽ được vector chuyển vị $v_{w_I}^T$ của vector đầu vào của từ w_I (dòng trong ma trận W tương ứng với vị trí từ đầu vào w_I trong bộ từ vựng).

Tại tầng đầu ra (output layer) thay vì tính một phân bố đa thức, ta tính C (C từ trong context) phân bố đa thức. Mỗi đầu ra được tính bằng cùng một ma trận W' giữa tầng ẩn và tầng output.

$$p(w_{c,j} | w_{O,c}, w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'} \exp(u_{j'})} \quad (10)$$

Trong đó $w_{c,j}$ là từ thứ j trong vector thứ c tại tầng đầu ra. $w_{O,c}$ là một từ dự đoán đúng thứ c trong các từ context, w_I là từ input, $y_{c,j}$ là đầu ra của của phần tử thứ j của vector thứ c trong tầng đầu ra, $u_{c,j}$ là giá trị đầu vào của phần tử thứ j của

vector thứ c trong tầng đầu ra, vì các vector tầng đầu ra dùng chung ma trận trọng số nên

$$u_{c,j} = u_j = v'_{w_j} \cdot h, \quad \text{với } c = 1, 2, \dots, C \quad (11)$$

Trong đó v'_{w_j} là vector đầu ra của từ thứ j trong bộ từ vựng V (là cột thứ j của ma trận W').

Hàm mất mát là [10]:

$$E = -\log p(w_{O,1}, w_{O,2}, \dots, w_{O,C} | w_I) \quad (12)$$

$$= -\log \prod_{c=1}^C \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^V \exp(u_{j'})} \quad (13)$$

$$= - \sum_{c=1}^C u_{j_c^*} + C \cdot \log \sum_{j'=1}^V \exp(u_{j'}) \quad (14)$$

Trong đó j_c^* là chỉ số của từ được dự đoán đúng thứ j trong vector thứ c của context C tại tầng đầu ra.

Thực hiện gradient descent và lan truyền ngược lỗi để cập nhật ma trận trọng số W' giữa tầng ẩn và tầng đầu ra, theo [10]:

$$v'_{w_j}^{(new)} = v'_{w_j}^{(old)} - \eta \cdot EI_j \cdot h \quad (15)$$

Trong đó h là giá trị đầu ra tại tầng ẩn được tính bởi công thức (9), EI_j là tổng lỗi dự đoán của các từ thứ j trong context C .

$$EI_j = \sum_{c=1}^C e_{c,j} \quad (16)$$

Trong đó $e_{c,j} = y_{c,j} - t_{c,j}$, với $y_{c,j}$ được tính ở công thức (10), $t_{c,j}$ có giá trị bằng 1 khi từ vị trí thứ j trong vector thứ c của context C tại tầng đầu ra là từ được dự đoán đúng, còn ngược lại thì bằng 0.

Cập nhật ma trận trọng số W giữa tầng đầu vào và tầng ẩn [10]:

$$v_{w_I}^{(new)} = v_{w_I}^{(old)} - \eta \cdot EH^T \quad (17)$$

Trong đó EH là vector N chiều, mỗi phần tử được tính như sau:

$$EH_i = \sum_{j=1}^V EI_j \cdot w'_{ij} \quad (18)$$

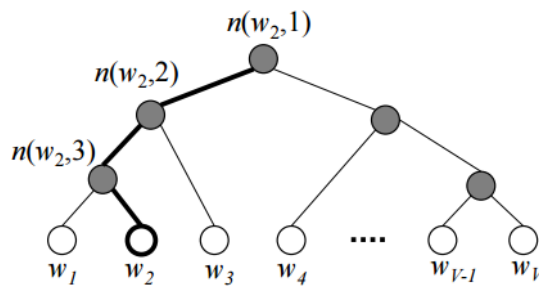
Trong đó EI_j được tính bởi công thức (16), w'_{ij} là thành phần trong ma trận trọng số W' .

Với hai mô hình trên CBOW và Skip-Gram, mỗi từ w trong bộ từ vựng tồn tại hai vector biểu diễn là vector đầu vào v_w (dòng của ma trận W giữa tầng đầu vào và tầng ẩn) và vector đầu ra v'_w (cột của ma trận W' giữa tầng ẩn và tầng đầu ra). Để cập nhật v_w thì đơn giản vì chỉ cần cập nhật một dòng của ma trận W tương ứng với

vị trí từ w trong bộ từ vựng với mỗi ví dụ học. Nhưng để cập nhật v'_w thì rất tốn kém, vì với mỗi ví dụ học, phải lặp mỗi từ w_j trong bộ từ vựng, tính net đầu vào của chúng u_j , xác suất dự đoán y_j (hoặc $y_{c,j}$ với Skip-gram), lỗi dự đoán e_j (hoặc El_j với Skip-Gram) và cuối cùng dùng lan truyền ngược lỗi để cập nhật các vector đầu ra v'_w cho chúng. Nên với những dữ liệu học cực lớn thì rất khó để thực hiện. Trong [9], có hai phương thức học hiệu quả hơn là **Hierarchical Softmax** và **Negative Sampling**.

2.1.3. Hierarchical Softmax

Hierarchical Softmax là một cải tiến của softmax. Model này sử dụng một cây nhị phân để biểu diễn tất cả các từ trong bộ từ vựng. Các từ sẽ nằm tại lá của cây (có V từ). Cây này có $V-1$ nút trong. Mỗi lá sẽ tồn tại một đường duy nhất đi từ nút gốc đến nút lá, và đường này được sử dụng để ước lượng xác suất của từ được biểu diễn bởi nút lá đó. Đây là ví dụ của một cây nhị phân, nút màu trắng là các từ trong bộ từ vựng V , nút màu đen là nút trong của cây. Một đường ví dụ từ nút gốc đến từ w_2 được in đậm, chiều dài của đường này $L_{w_2} = 4$ (số nút từ nút gốc đến nút lá biểu diễn từ w_2). $n(w, j)$ là nút thứ j trên đường đi từ root đến w .



Hình 4. Cây nhị phân [10]

Trong hierarchical softmax model, không có vector đầu ra biểu diễn cho từ. Thay vào đó là mỗi $V-1$ nút trong có một vector đầu ra $v'_{n(w,j)}$. Và xác suất của một từ là từ đầu ra được xác định như sau [9]:

$$p(w) = \prod_{j=1}^{L_w} \sigma(n(w, j+1) - ch(n(w, j)) \cdot v'_{n(w, j)}^T h) \quad (19)$$

Trong đó $\sigma(x) = 1/(1 + \exp(-x))$ [5], $ch(n)$ là con trái của nút n , $v'_{n(w,j)}$ là vector biểu diễn (output vector) của nút trong $n(w, j)$, h là giá trị đầu ra của tầng ẩn (trong skip-gram model $h = v_{w_i}^T$, trong CBOW $h = \frac{1}{C} \sum_{c=1}^C v_{w_c}^T$, x là một hàm được xác định như sau:

$$x = \begin{cases} 1 & \text{nếu } x \text{ đúng} \\ -1 & \text{ngược lại} \end{cases} \quad (20)$$

$$\text{Hiển nhiên có } \sum_{i=1}^V p(w_i) = 1 \quad (21)$$

Điều này làm cho hierarchical softmax tốt trong việc phân tán đa thức giữa tất cả các từ trong từ vựng.

Chú ý: Về mặt lý thuyết hierarchical softmax có thể sử dụng một trong các kiểu cây khác nhau, nhưng word2vec sử dụng cây nhị phân Huffman để học nhanh hơn. (Cây nhị phân Huffman, mỗi nút luôn có 2 con).[9]

Bây giờ ta sẽ cập nhật tham số cho vector biểu diễn của các nút trong. Để đơn giản ta sẽ sử dụng context chỉ chứa một từ trước. Sau đó việc mở rộng ra cho CBOW và Skip-gram sẽ dễ dàng hơn. Theo [10], để đơn giản về việc ký hiệu, ta ký hiệu như sau:

$$\cdot = n(w, j+1) = ch(n(w, j)) \quad (22)$$

Đối với mỗi ví dụ học, hàm lỗi được xác định như sau:

$$E = -\log p(w) = -\log \prod_{j=1}^L \sigma(v_j'^T h) \quad (23)$$

Sử dụng gradient descent và lan truyền ngược lỗi theo [10] ta có công thức cập nhật vector biểu diễn của nút trong là:

$$v_j'^{(new)} = v_j'^{(old)} - \eta(\sigma(v_j'^T h) - t_j) \cdot h \quad (24)$$

Trong đó $j = 1, 2, \dots, L(w) - 1$, $t_j = 1$ nếu \cdot đúng, $t_j = 0$ nếu ngược lại.

Có thể hiểu $\sigma(v_j'^T h) - t_j$ như là lỗi dự đoán cho nút $n(w, j)$. Nhiệm vụ của mỗi nút trong là dự đoán nên đi theo phía trái hay phía phải trong bước đi ngẫu nhiên. $t_j = 1$ nghĩa là nên đi theo con trái, $t_j = 0$ nghĩa là nên đi theo con phải. $\sigma(v_j'^T h)$ là kết quả dự đoán. Cho mỗi ví dụ học, nếu sự dự đoán của nút trong là rất gần với đường đi đúng thì vector biểu diễn v_j' sẽ chỉ di chuyển một chút ít, ngược lại v_j' sẽ di chuyển theo hướng thích hợp bằng cách di chuyển (có thể gần hoặc xa hơn so với h) vì vậy sẽ làm giảm lỗi dự đoán cho ví dụ học này. Công thức cập nhật này có thể được sử dụng cho cả CBOW và Skip-gram model. Khi sử dụng cho skip-gram model cần lặp lại cập nhật này cho mỗi C từ trong context đầu ra [10].

Để truyền ngược lỗi để học trọng số ma trận W giữa tầng đầu vào và tầng ẩn, ta sẽ lấy vi phân của E theo output của hidden layer h [10].

$$\frac{\partial E}{\partial h} = \sum_{j=1}^{L(w)-1} \frac{\partial E}{\partial v_j'^T h} \cdot \frac{\partial v_j'^T h}{\partial h} \quad (25)$$

$$= \sum_{j=1}^{L(w)-1} (\sigma(v_j'^T h) - t_j) \cdot v_j' \quad (26)$$

$$:= EH \quad (27)$$

Trong đó ký hiệu $:=$ là ký hiệu công thức (26) được đặt là EH .

Có thể thay thế trực tiếp EH vào (7) để được công thức cập nhật cho các vector đầu vào (dòng của ma trận W) của CBOW. Với skip-gram model ta cần tính một giá trị EH cho mỗi từ trong skip-gram context, thay tổng các giá trị EH vào (17) để được công thức cập nhật cho vector đầu vào của từ (dòng của ma trận W). Từ các công thức cập nhật ta có thể thấy rằng độ phức tạp tính toán cho mỗi ví dụ học trên mỗi context word sẽ giảm từ $O(V)$ sang $O(\log(V))$, đây là một cải thiện lớn trong tốc độ.

2.1.4. Negative Sampling

Ý tưởng của negative sampling là chỉ cập nhật một phần các trọng số trong hidden->output layer matrix. Các trọng số được cập nhật bao gồm của từ đầu ra và các từ được chọn là negative sampling (các từ không phải là từ được dự đoán đúng). Ta gọi xác suất phân bố được cần cho quá trình sampling này là sự phân bố nhiễu và ký hiệu nó là $P_n(w)$. Người ta có thể chọn một phân bố tốt theo kinh nghiệm. Ta thấy các từ có tần suất càng lớn trong bộ từ vựng thì khả năng chọn là negative sampling là cao hơn. Xác suất để chọn ra từ là negative sampling được tính như sau [9]:

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^n (f(w_j)^{3/4})} \quad (28)$$

Trong đó $f(w)$ là tần suất của từ trong bộ từ vựng. n là số từ trong bộ từ vựng.

Trong word2vec thay vì sử dụng cấu trúc của negative sampling để tạo phân bố đa thức sau thì tác giả [9] đưa ra một mục tiêu học đơn giản có khả năng tạo ra vector từ có chất lượng là:

$$E = -\log \sigma(v'_{w_o}{}^T h) - \sum_{w_j \in W_{neg}} \log \sigma(v'_{w_j}{}^T h) \quad (29)$$

Trong đó w_o là từ đầu ra (giả sử positive sample) và v'_{w_o} là vector biểu diễn đầu ra của nó, h là giá trị đầu ra của hidden layer: $h = \frac{1}{C} \sum_{c=1}^C v_{w_c}$ trong CBOW model và $h = v_{w_i}$ trong skip-gram model, $W_{neg} = \{w_j | j = 1, \dots, K\}$ là tập các từ được chọn dựa trên $P(w_i)$ (negative sample)

Theo [10], công thức cập nhật cho vector đầu ra của từ w_j (hay cột của ma trận W' giữa tầng ẩn và tầng đầu ra):

$$v'_{w_j}{}^{(new)} = v'_{w_j}{}^{(old)} - \eta(\sigma(v'_{w_j}{}^T h) - t_j) \cdot h \quad (30)$$

Trong đó $t_j = 1$ khi từ w_j là từ được dự đoán đúng (positive sample), $t_j = 0$ nếu ngược lại.

Điều này chỉ cần áp dụng cho những từ là từ được dự đoán đúng w_o (positive sample) và những từ thuộc W_{neg} thay vì tất cả từ trong bộ từ vựng. Điều này cho thấy ta có thể giảm khối lượng tính toán đi rất nhiều trên mỗi vòng lặp. Công thức này được áp dụng cho cả CBOW và Skip-gram. Với Skip-gram ta áp dụng công thức này cho mỗi từ trong context mỗi lần.

Để truyền ngược lỗi về hidden layer và cập nhật vector input của từ, ta cần lấy vi phân của E theo đầu ra của hidden layer h [10], ta được:

$$\frac{\partial E}{\partial h} = \sum_{w_j \in w_o \cup W_{neg}} \frac{\partial E}{\partial v'_{w_j}{}^T h} \cdot \frac{v'_{w_j}{}^T h}{\partial h} \quad (31)$$

$$= \sum_{w_j \in w_o \cup W_{neg}} (\sigma(v'_{w_j}{}^T h) - t_j) \cdot v'_{w_j} := EH \quad (32)$$

Bằng cách thay EH vào công thức (7) ta được công thức cập nhật cho vector đầu vào (dòng của ma trận W) của CBOW model. Và với Skip-gram model ta cần tính một giá trị EH cho mỗi từ trong context và thay tổng các giá trị EH này vào công thức (17) để được công thức cập nhật cho vector đầu vào của từ (dòng của ma trận W).

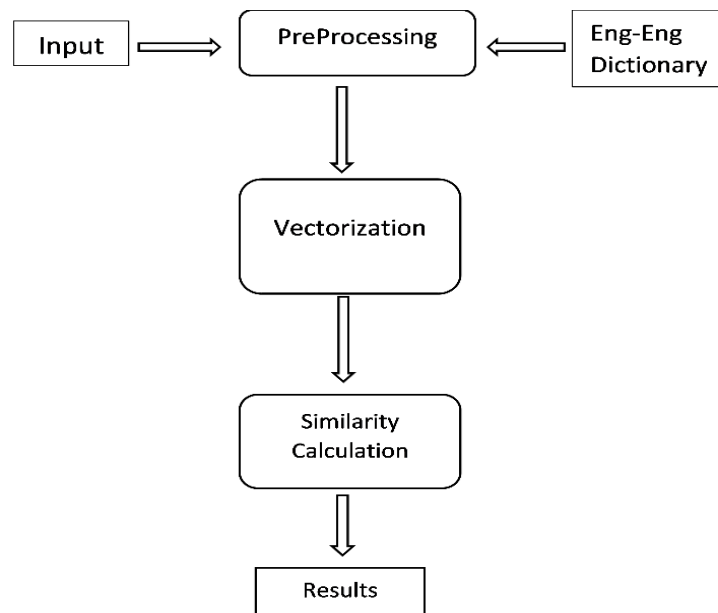
III. Phân tích và thiết kế hệ thống

3.1. Hệ thống đề xuất

Hệ thống được thực hiện như hình 6. Cụ thể đoạn mô tả đầu vào và nghĩa của từ trong từ điển Anh Anh đều được tiền xử lý (preprocessing) loại bỏ từ dừng và các ký tự đặc biệt. Sau đó được vector hóa bằng bộ dữ liệu word2vec training sẵn. Hệ thống sẽ tính toán và trả về kết quả.

Như đã đề cập ở các phần trước, thực chất của việc tìm ra tập từ tiếng Anh tốt nhất biểu diễn nghĩa của một đoạn mô tả bằng tiếng Anh chính là việc tính độ tương đồng giữa hai câu, đoạn mô tả và nghĩa của từ trong từ điển Anh Anh (cụ thể ở đây là từ điển Oxford) bằng độ tương đồng cosine. Sau đó chọn ra tập từ với nghĩa có độ tương đồng cao nhất với đoạn mô tả. Quá trình thực hiện xây dựng hệ thống như sau:

- Xây dựng dữ liệu từ điển Anh Anh.
- Thu thập các bộ dữ liệu word2vec đã được training sẵn.
- Xây dựng chức năng tính độ tương đồng của hai câu.
- Tìm ra tập từ tốt nhất biểu diễn nghĩa của đoạn mô tả.



Hình 5. Mô hình hệ thống

3.1.1. Xây dựng dữ liệu từ điển Anh Anh

Để xây dựng chức năng xây dựng dữ liệu từ điển Anh Anh, tôi chọn dữ liệu của từ điển Oxford, vì nghĩa của từ trong từ điển Oxford được biểu đạt dễ hiểu, cặn kẽ, bên cạnh đó Oxford có cung cấp một trang web từ điển online [ở đây](#). Nên tôi thực hiện viết chương trình crawl dữ liệu từ điển ở trang web này.

3.1.2. Thu thập các bộ dữ liệu word2vec đã được training sẵn

Với chức năng thu thập các bộ dữ liệu word2vec đã được training sẵn. Vì word2vec được sử dụng nhiều trong thời gian gần đây nên có nhiều bộ dữ liệu đã được training sẵn, điển hình bộ dữ liệu word2vec training sẵn của google được training trên 100 tỷ từ từ dữ liệu của Google News. Nhưng do hạn chế của máy tính làm đồ án nên tôi chỉ dùng lại dùng các bộ dữ liệu có kích thước nhỏ hơn đã được train sẵn ở “Word Embeddings Released with the ADCS 2015 paper” [11].

3.1.3. Chức năng tính độ tương đồng giữa hai câu

Như đã đề cập ở các phần trước, để tính độ tương đồng của hai câu, tôi đề xuất hai phương pháp kết hợp với dữ liệu word2vec training sẵn là:

- Lấy trung bình các vector biểu diễn các từ trong câu để được vector cho câu, sau đó sử dụng độ tương đồng cosine để tính độ tương đồng giữa hai câu.
- Tính độ tương đồng cosin của mỗi cặp từ, một từ của câu này với các từ trong câu kia, sau đó lấy giá trị max, nghĩa là lấy từ nào trong câu này gần nghĩa nhất với từ trong câu kia, sau đó trung bình cộng các giá trị max này lại, ta được độ tương đồng của 2 câu.

a. Phương pháp 1 lấy trung bình các vector biểu diễn từ trong câu

Với phương pháp này việc thực hiện rất nhanh và khá hiệu quả nhưng cách này gặp một điểm yếu là mất trật tự của từ trong câu nhưng vì bài toán của chúng tôi là tìm ra tập từ, gồm nhiều từ nên việc tìm ra các từ không liên quan, trái nghĩa cũng không ảnh hưởng quá nhiều, vì chỉ cần trong tập từ có từ thỏa mãn nội dung của đoạn mô tả là đạt yêu cầu của bài toán.

Ban đầu nghĩa của các từ trong từ điển Anh Anh (đã được thu thập) sẽ được loại bỏ từ dùng bằng bộ thư viện NLTK. Sau đó mỗi từ trong câu nghĩa sẽ được chuyển thành vector bằng bộ dữ liệu word2vec đã được học sẵn [11]. Mỗi từ sẽ tương ứng với một vector riêng biệt có dạng $v_w = (0.102, -4.31, -0.003, \dots)$ với số chiều tùy thuộc vào bộ dữ liệu học sẵn word2vec 100, 200 đến 1000.

Công thức để tính vector biểu diễn cho câu là:

$$v_{sentence} = \frac{1}{N} \sum_{j=1}^N v_{w_j} \quad (33)$$

Trong đó N là số từ trong câu, v_{w_j} là vector biểu diễn của từ w_j trong câu.

Khi đã có được vector câu từ (33) thì công thức tính độ tương đồng cosine giữa hai câu x, y:

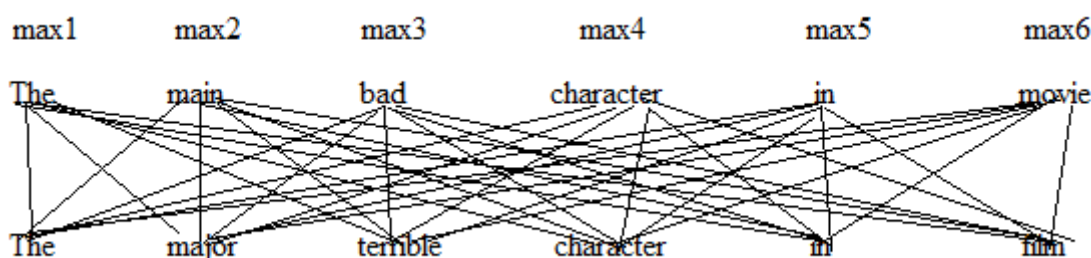
$$Similarity_{x,y} = \frac{\sum_{i=1}^n v_{x_i} \cdot v_{y_i}}{\sqrt{\sum_{i=1}^n v_{x_i}^2} \cdot \sqrt{\sum_{i=1}^n v_{y_i}^2}} \quad (34)$$

Trong đó v_x là vector biểu diễn của câu x , v_y là vector biểu diễn của câu y , n là số chiều của vector biểu diễn câu.

b. Phương pháp 2 lấy trung bình max của độ tương đồng các cặp từ

Tính độ tương đồng cosine của mỗi cặp từ, một từ của câu này với các từ trong câu kia, sau đó lấy giá trị max, nghĩa là lấy từ nào trong câu này gần nghĩa nhất với từ trong câu kia, sau đó trung bình cộng các giá trị max này lại, ta được độ tương đồng của 2 câu.

Ví dụ:



Hình 6. Biểu diễn cách tính độ tương đồng 2 câu theo phương pháp 2

Công thức tính độ tương đồng cosine giữa hai từ w_i, w_j là:

$$Similarity_{w_i, w_j} = \frac{\sum_{k=1}^n v_{w_{ik}} \cdot v_{w_{jk}}}{\sqrt{\sum_{k=1}^n v_{w_{ik}}^2} \cdot \sqrt{\sum_{k=1}^n v_{w_{jk}}^2}} \quad (35)$$

Trong đó n là số chiều của vector biểu diễn từ, $v_{w_{ik}}$ là thành phần trong vector v_{w_i} của từ w_i , $v_{w_{jk}}$ là thành phần trong vector v_{w_j} của từ w_j .

Công thức tính độ tương đồng giữa hai câu x, y của phương pháp này là:

$$Similarity_{x,y} = \frac{1}{N} \sum_{i=1}^N \max_{j \in \{1, \dots, M\}} Similarity_{w_i, w_j} \quad (36)$$

Trong đó N là số từ trong câu x , M là số từ trong câu y , w_i là từ trong câu x , w_j là từ trong câu y . $Similarity_{w_i, w_j}$ là độ tương đồng giữa hai từ được tính ở công thức (35).

Với tính độ tương đồng của hai câu của phương pháp 2 này, mỗi từ của câu này sẽ tìm từ nào có độ tương đồng lớn nhất trong câu còn lại, điều này sẽ cho thấy những cặp từ nào có độ tương đồng lớn sẽ có ảnh hưởng lớn đến tìm độ tương đồng của các câu.

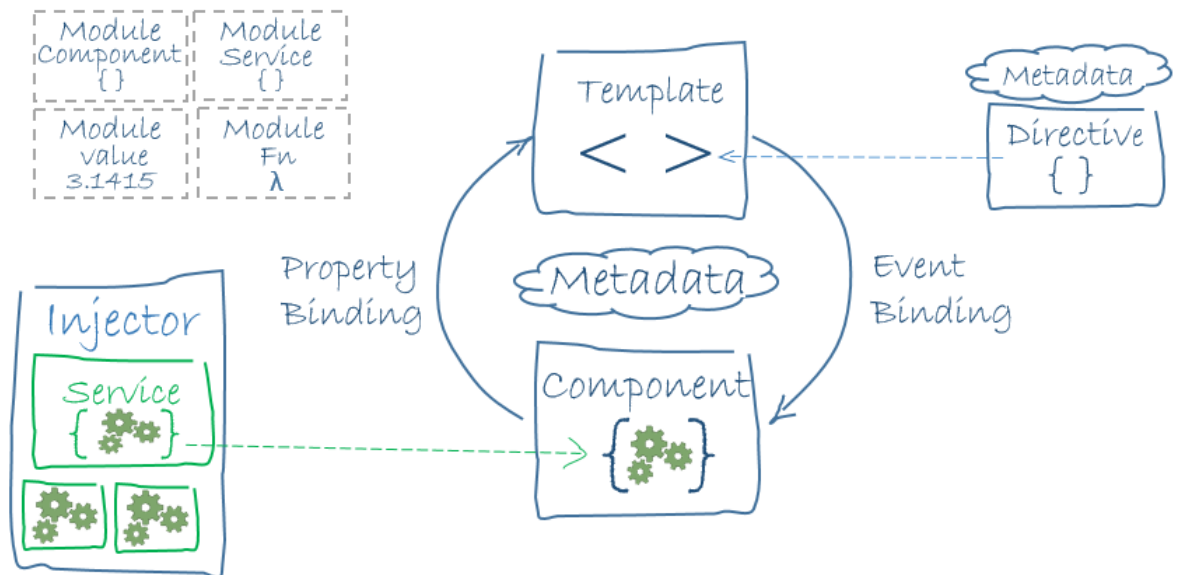
Khi đã biết được độ tương đồng giữa nghĩa của từ trong từ điển Anh Anh và đoạn mô tả. Hệ thống sẽ lấy tập những từ nào có nghĩa tương đồng nhất với đoạn mô tả dựa vào hai phương pháp tính độ tương đồng giữa hai câu trên.

3.2. Các công nghệ được sử dụng

3.2.1. Angular 2

Angular 2 là một framework của javascript, dùng để xây dựng ứng dụng phía client. Angular 2 chứa sẵn rất nhiều thư viện hỗ trợ vì vậy việc phát triển ứng dụng rất dễ dàng và nhanh chóng.

Kiến trúc của Angular 2 như sau:



Hình 7. Kiến trúc Angular 2 [13]

➤ Gồm các thành phần chính sau [13]:

- **Modules:** Các app trong Angular đều được module hóa và Angular có một module hệ thống gọi là Angular modules hoặc là NgModules.
- **Components:** Một component điều khiển một phần View. Vì vậy các View khác nhau sẽ tách biệt, nhưng chúng vẫn có thể tương tác với nhau.
- **Templates:** Template là phần HTML nói cho Angular cách tạo ra component.
- **Metadatas:** nói cho Angular biết cách xử lý một lớp (Class).
- **Data Binding:** Data Binding trong Angular 2 chính là sự tự động đồng bộ dữ liệu giữa Model và View. Data Binding cho phép tạo sự liên kết. Khi có bất kỳ sự thay đổi nào ở model, dữ liệu sẽ được thay đổi ở Template và ngược lại.
- **Directives:** Các template của Angular là kiểu động, khi Angular tạo ra chúng thì DOM bị biến đổi theo directives.
- **Services:** Service là một lớp mà có thể sử dụng ở các component. Nó có nhiều chức năng khác nhau như log thông báo, giao tiếp với server qua API,...
- **Dependency injection:** Dependency injection là một cách để cung cấp một class có sẵn với đầy đủ các dependencies mà nó cần. Hầu hết các dependency là services.

➤ **Ưu điểm của Angular 2:**

- Angular 2 là một framework đa nền tảng vì vậy có thể xây dựng ứng dụng đa nền tảng như web, di động, ...
- Hiệu năng của Angular 2 nhanh, tương thích với nhiều loại trình duyệt và hệ điều hành di động.
- Angular 2 sử dụng TypeScript nên nhận được sự hỗ trợ nhiều từ cộng đồng sử dụng .NET framework, việc sử dụng các thư viện TypeScript trong Angular được thực hiện dễ dàng.

➤ **Nhược điểm của Angular 2:**

- Vì là framework của javascript nên nó bị phụ thuộc bởi người dùng, khi người dùng vô hiệu hóa javascript trên trình duyệt thì ứng dụng sử dụng Angular sẽ không hoạt động.
- Vì là framework mới nên trao đổi trên các diễn đàn vẫn chưa nhiều nên khó khăn trong việc giải quyết vấn đề trong Angular của những người mới làm quen Angular 2.

3.2.2. Django Framework

Django là một web framework miễn phí mã nguồn mở được viết bằng Python. Django sử dụng mô hình model-view-template (MVT). Django được phát triển và bảo trì bởi *Django Software Foundation*(DSF) – một tổ chức phi lợi nhuận độc lập.

Mục tiêu chính của Django là đơn giản hóa việc tạo các website phức tạp có sử dụng cơ sở dữ liệu. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các component, tính năng phát triển nhanh, không làm lại những gì đã làm. Một số website nổi tiếng được biết tới xây dựng từ Django là Pinterest, Instagram, Mozilla, và Bitbucket, Disqus. [14]

3.3. Phân tích hệ thống

3.3.1. Mô tả hệ thống

a. Mục tiêu của hệ thống

Dựa vào vấn đề và giải pháp được đưa ra ở trên thì có thể thấy hệ thống sẽ mang lại các giá trị lớn như sau:

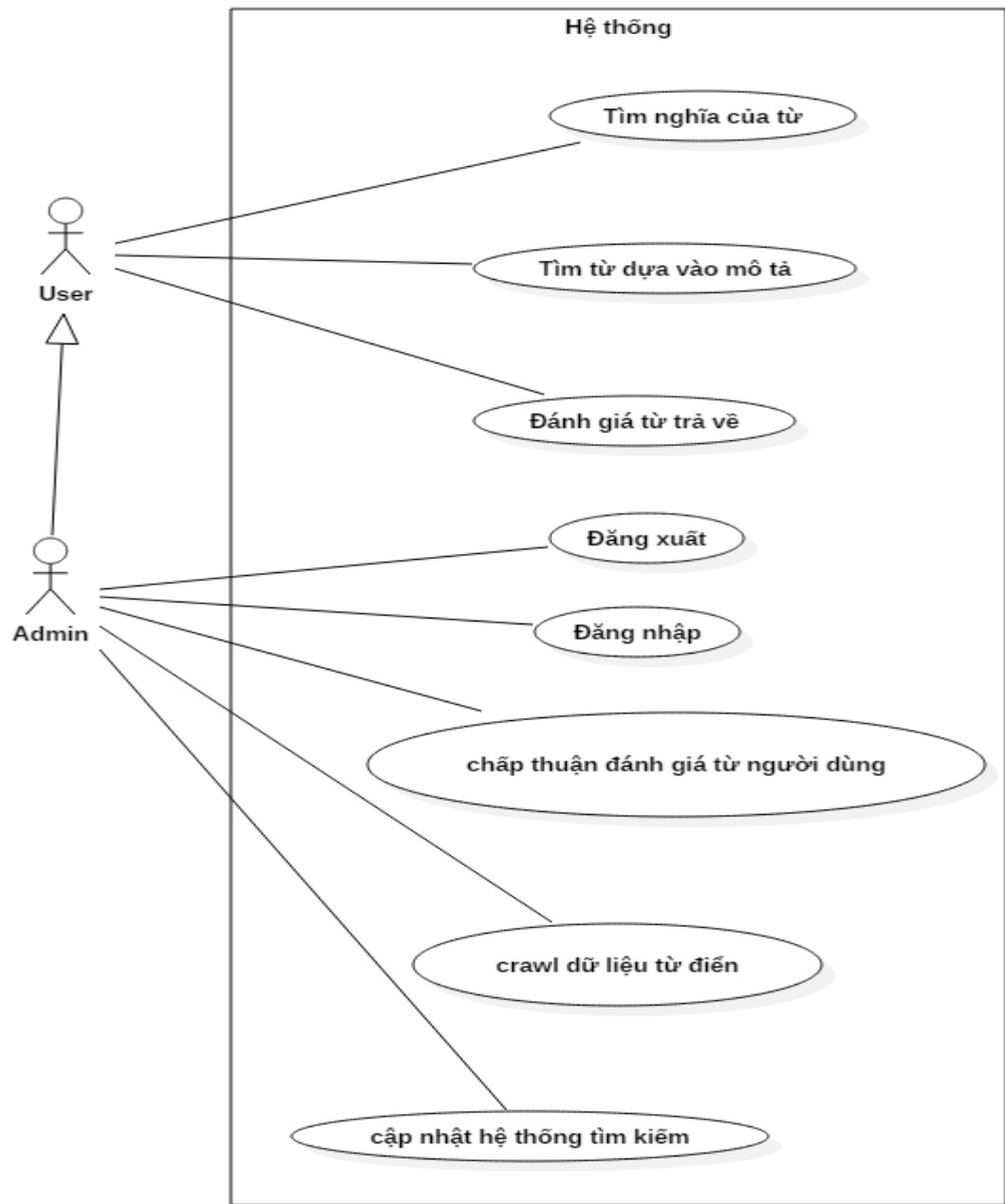
- ✓ Mang lại giá trị nghiệp vụ: Vì hệ thống được xây dựng trên nền tảng web nên người dùng có thể sử dụng tại bất cứ đâu chỉ cần có kết nối mạng trên điện thoại hoặc máy tính.
- ✓ Mang lại giá trị sử dụng : Người dùng có thể nhanh chóng tìm ra các từ tiếng Anh mà người dùng muốn chỉ bằng việc mô tả từ đó hoặc có thể tra từ để xem nghĩa, cách phát âm, các ví dụ của từ đó trong tiếng Anh. Người dùng không cần phải cài đặt hệ thống mà có thể sử dụng trực tiếp trên web, mang lại tính tiện lợi, dễ sử dụng.

b. Các yêu cầu của hệ thống:

- ✓ Hệ thống cung cấp giao diện dễ sử dụng, trả về kết quả nhanh chóng.
- ✓ Hệ thống cung cấp hai chức năng chính là tra từ để tìm nghĩa của từ và tìm từ dựa vào mô tả của từ.
- ✓ Hệ thống cung cấp chức năng đánh giá của người dùng cho từ trong tập kết quả từ trả về. Từ đó để cải thiện hệ thống tìm kiếm cho các lần tìm kiếm tiếp theo.
- ✓ Hệ thống cung cấp giao diện cho admin để quản lý các thông tin được đánh giá từ người dùng để từ đó cập nhật vào cơ sở dữ liệu từ điển.

3.3.2. Mô tả Use Case

a. Use case tổng quát



Hình 8. Biểu đồ usecase tổng quát

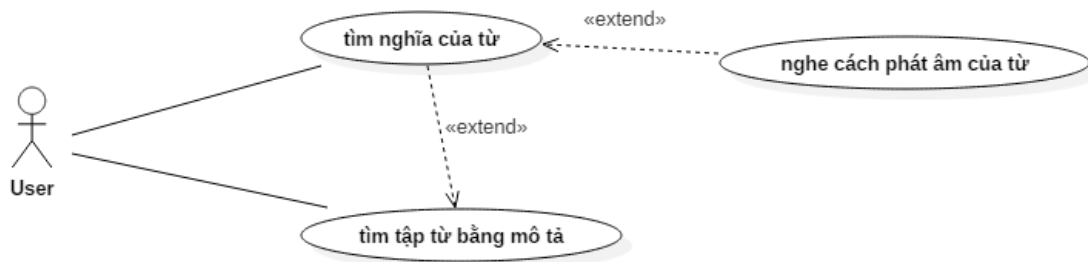
Các tác nhân tham gia vào hệ thống:

Hệ thống có hai tác nhân là tác nhân User và tác nhân Admin:

- ❖ Tác nhân User: là tác nhân không có tài khoản trong hệ thống có thể thực hiện các chức năng chính của hệ thống như tìm nghĩa của từ, tìm tập từ bằng mô tả từ đó
- ❖ Tác nhân Admin: là tác nhân kế thừa tác nhân User, có thể thực hiện được tất cả các chức năng của tác nhân User và thực hiện các chức năng của admin như

chấp nhận đánh giá của người dùng, cập nhật cơ sở dữ liệu từ điển, cập nhật hệ thống tìm kiếm.

b. Use Case User

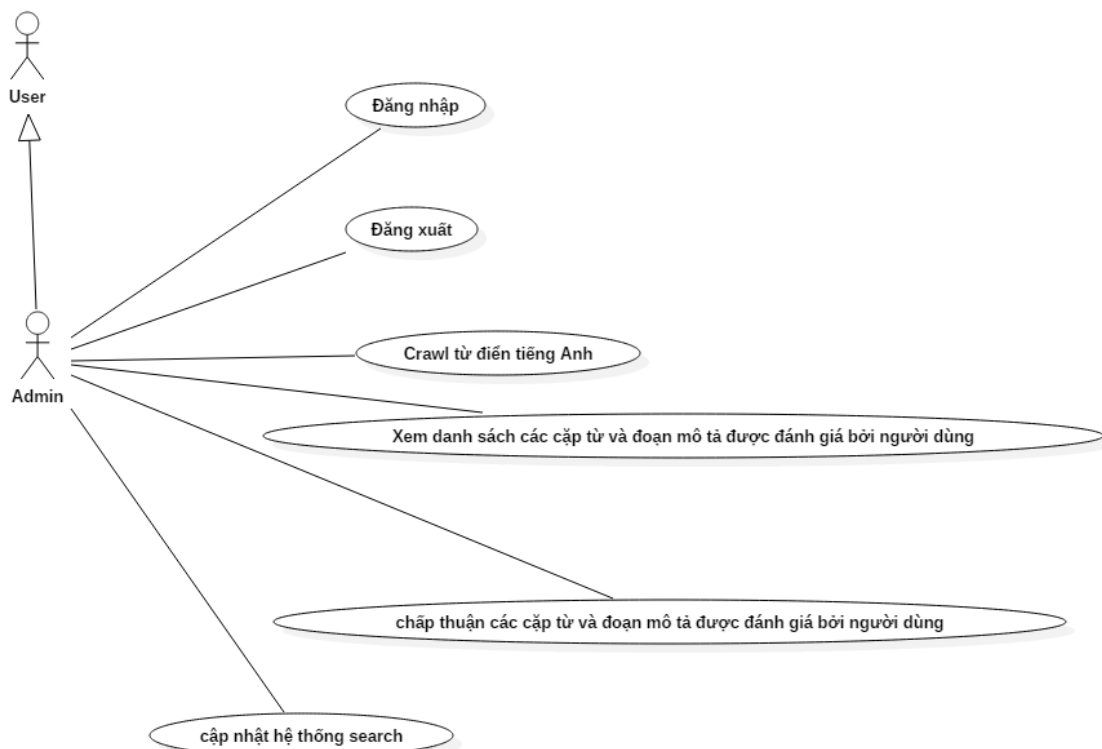


Hình 9. Biểu đồ use case tác nhân User

Đây là ca sử dụng của tác nhân User.

- User là tác nhân không có tài khoản để đăng nhập vào hệ thống
- Nhưng user có các chức năng chính của hệ thống như tra cứu nghĩa của từ, và có thể nghe cách phát âm của từ.
- Tìm tập từ bằng cách mô tả nghĩa của từ đó. Khi hệ thống trả về tập từ, user có thể xem nghĩa của mỗi từ bằng cách nhấp vào từ đó.

c. Use Case Admin



Hình 10. Biểu đồ use case Admin

Đây là ca sử dụng của Admin

- Admin là tác nhân kế thừa tác nhân User, có đầy đủ các chức năng của User.
- Admin là tác nhân có quyền cao nhất trong hệ thống.

- Admin có một tài khoản trong hệ thống để đăng nhập và thực hiện các chức năng lấy dữ liệu từ điển tiếng Anh trên trang từ điển Oxford để cập nhật thêm từ vào cơ sở dữ liệu tiếng Anh của hệ thống.
- Khi người dùng sử dụng hệ thống, người dùng có thể đánh giá từ trong tập từ trả về bởi hệ thống có phù hợp với đoạn mô tả mà người dùng cung cấp hay không từ đó admin có thể xem xét và chấp thuận có cập nhật vào cơ sở dữ liệu hay không, điều này sẽ cải thiện hệ thống tìm kiếm cho các lần tìm kiếm sau của người dùng. Chức năng này rất quan trọng vì bản chất của chức năng tìm tập từ của hệ thống dựa vào đoạn mô tả mà người dùng cung cấp là độ tương đồng của đoạn mô tả với nghĩa của các từ trong cơ sở dữ liệu mà mỗi người dùng sẽ có khả năng mô tả nghĩa của từ khác nhau. Vì vậy đoạn mô tả càng được mô tả gần với nghĩa của từ thì các từ trả về càng chính xác. Nghĩa của từ ở đây đã được cập nhật bởi sự đánh giá của người dùng. Càng nhiều người dùng phản hồi đánh giá, hệ thống hoạt động càng tốt.
- Các dữ liệu được tính toán được lưu vào file dưới dạng binary để thực hiện tính toán nhanh hơn khi tìm kiếm từ bằng đoạn mô tả. Vì vậy khi cơ sở dữ liệu từ điển Anh Anh của hệ thống được cập nhật thì admin có thể chạy chức năng cập nhật hệ thống tìm kiếm để tính toán và cập nhật dữ liệu vào file dưới dạng binary.

3.3.3. Đặc tả các chức năng của hệ thống

a. Đặc tả use case UC01 « Đăng nhập »

Mã use case	UC01	Tên use case	Đăng nhập
Tác nhân	Admin		
Mô tả	Use case mô tả quá trình đăng nhập của admin khi muốn tham gia vào các chức năng của hệ thống		
Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Admin	Chọn chức năng đăng nhập
	2	Hệ thống	Hiện thị form đăng nhập
	3	Admin	Nhập các thông tin username và password được mô tả ở bảng dưới.
	4	Admin	Yêu cầu đăng nhập
	5	Hệ thống	Kiểm tra các thông tin mà khách nhập vào
	6	Hệ thống	Gửi thông báo đồng thời hiện thị màn hình chức năng của admin
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	5a1	Hệ thống	Thông báo lỗi nếu admin nhập thiếu tên tài khoản hoặc mật khẩu
	5a2	Hệ thống	Thông báo lỗi nếu nhập tên

		tài khoản hoặc mật khẩu không tồn tại trong hệ thống
Hậu điều kiện	Không	

Bảng 1. Đặc tả use case UC01 « Đăng nhập »

Dữ liệu đầu vào khi đăng nhập

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Username		Có		cuong
2	Password		Có		cuong1234

Bảng 2. Mô tả dữ liệu đầu vào khi đăng nhập

b. Đặc tả use case UC02 « Tìm nghĩa của từ »

Mã use case	UC02		Tên use case	Tìm nghĩa của từ
Tác nhân	User			
Mô tả	Use case mô tả quá trình user tra từ, tìm nghĩa của từ khi tham gia vào hệ thống			
Tiền điều kiện	Không			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động	
	1	User	Chọn chức năng tìm kiếm	
	2	Hệ thống	Dẫn đến trang tìm kiếm	
	3	User	Chọn chức năng tìm kiếm bởi từ (search by word)	
	4	User	Nhập từ cần tìm kiếm	
	5	User	Gửi yêu cầu tìm kiếm	
	6	Hệ thống	Kiểm tra thông tin được nhập vào	
	7	Hệ thống	Hệ thống tìm kiếm từ trong cơ sở dữ liệu	
	8	Hệ thống	Hiển thị nghĩa, cách phát âm, ví dụ của từ được tìm kiếm	
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động	
	6a	Hệ thống	Thông báo cần nhập duy nhất một từ.	
	7a	Hệ thống	Hệ thống sẽ tự động crawl từ ở trang web trực tuyến Oxford nếu từ không có ở cơ sở dữ liệu của hệ thống	
	7b	Hệ thống	Hiển thị thông báo nếu từ không tồn tại hoặc quá trình crawl thất bại	
Hậu điều kiện	Không			

Bảng 3. Đặc tả use case UC02 « Tìm nghĩa của từ »

c. Đặc tả use case UC03 « Tìm tập từ bằng mô tả »

Mã use case	UC03	Tên use case	Tìm tập từ bằng mô tả
Tác nhân	User		
Mô tả	Use case mô tả quá trình user tìm tập từ bằng cách mô tả nghĩa của từ trong tập từ mà user muốn tìm		
Tiền điều kiện	Không		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	User	Chọn chức năng tìm kiếm
	2	Hệ thống	Dẫn đến trang tìm kiếm
	3	User	Chọn chức năng tìm kiếm bởi mô tả (search by description)
	4	User	Nhập đoạn mô tả cần tìm kiếm
	5	User	Chọn phương pháp tìm kiếm từ
	6	User	Chọn số lượng từ cần trả về
	7	User	Gửi yêu cầu tìm kiếm
	8	Hệ thống	Kiểm tra thông tin được nhập vào
	9	Hệ thống	Hiển thị tập từ
	10	User	Nhấp vào từ để gửi yêu cầu xem nghĩa của từ
	11	Hệ thống	Hiển thị nghĩa, cách phát âm, ví dụ của từ được tìm kiếm
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	8a	Hệ thống	Thông báo cần nhập lại nếu user nhập các ký tự đặc biệt hoặc các từ không có nghĩa.
	9a	Hệ thống	Hiển thị thông báo và trả về danh sách rỗng nếu không tìm thấy từ nào biểu diễn nghĩa của đoạn mô tả
Hậu điều kiện	Không		

Bảng 4. Đặc tả use case UC03 « Tìm tập từ bằng mô tả »

d. Đặc tả use case UC04 « Nghe phát âm của từ »

Mã use case	UC04	Tên use case	Nghe phát âm của từ
Tác nhân	User		
Mô tả	Use case mô tả quá trình user nghe cách phát âm của từ khi đã tra từ		
Tiền điều kiện	User tìm nghĩa của từ		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	User	Click chuột lên biểu tượng loa (Anh Anh hoặc Anh Mỹ) để gửi yêu cầu nghe phát âm

Luồng sự kiện thay thế	2	Hệ thống	Gửi yêu cầu phát âm đến trang từ điển online Oxford
	3	Hệ thống	Trả về âm của từ
	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Trả về thông báo lỗi nếu gửi yêu cầu đến trang Oxford thất bại.
Hậu điều kiện	Không		

Bảng 5. Đặc tả use case UC04 « Nghe phát âm của từ »

e. Đặc tả use case UC05 « Crawl từ điển Anh Anh »

Mã use case	UC05		Tên use case	Crawl từ điển Anh Anh
Tác nhân	Admin			
Mô tả	Use case mô tả quá trình crawl từ điển Anh Anh của Admin			
Tiền điều kiện	Đăng nhập			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động	
	1	Admin	Gửi yêu cầu crawl dữ liệu từ điển	
	2	Hệ thống	Chạy crawl đồng thời cập nhật vào cơ sở dữ liệu những từ crawl được.	
	3	Hệ thống	Trả về thông báo quá trình crawl thành công.	
Luồng sự kiện thay thế	Không có			
Hậu điều kiện	Không			

Bảng 6. Đặc tả use case UC05 « Crawl từ điển Anh Anh »

f. Đặc tả use case UC06 « Xem danh sách cặp từ và đoạn mô tả được đánh giá »

Mã use case	UC06		Tên use case	Xem danh sách cặp từ và đoạn mô tả được đánh giá
Tác nhân	Admin			
Mô tả	Use case mô tả quá trình xem danh sách cặp từ và đoạn mô tả được đánh giá bởi người dùng			
Tiền điều kiện	Đăng nhập			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động	
	1	Admin	Gửi yêu cầu xem danh sách	
	2	Hệ thống	Hiển thị danh sách các cặp từ và đoạn mô tả được đánh giá bởi người dùng	
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động	
	2a	Hệ thống	Hệ thống trả về danh sách trống nếu không có đánh giá	

			nào của người dùng
Hậu điều kiện	Không		

Bảng 7. Đặc tả use case UC06 « Xem danh sách cặp từ và đoạn mô tả được đánh giá »

g. Đặc tả use case UC07 « Chấp thuận các cặp từ và đoạn mô tả được đánh giá bởi người dùng »

Mã use case	UC07		Tên use case	Chấp thuận các cặp từ và đoạn mô tả được đánh giá bởi người dùng
Tác nhân	Admin			
Mô tả	Use case mô tả quá trình chấp thuận các cặp từ và đoạn mô tả được đánh giá bởi người dùng			
Tiền điều kiện	Đăng nhập, xem danh sách cặp từ và đoạn mô tả được đánh giá bởi người dùng			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động	
	1	Admin	Chấp thuận cặp từ và đoạn mô tả bằng cách tick vào biểu tượng is_approved tương ứng.	
	2	Hệ thống	Cập nhật trường is_approved = true của bảng definition (nghĩa của từ) trong cơ sở dữ liệu	
	3	Hệ thống	Hiển thị thông báo chấp thuận thành công, đồng thời cặp từ và đoạn mô tả sẽ biến mất trong danh sách cặp từ và đoạn mô tả.	
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động	
	2a	Hệ thống	Hệ thống hiển thị thông báo lỗi nếu xảy ra lỗi trong hệ thống	
Hậu điều kiện	Không			

Bảng 8. Đặc tả use case UC07 « Chấp thuận các cặp từ và đoạn mô tả của người dùng »

h. Đặc tả use case UC08 « Cập nhật hệ thống tìm kiếm »

Mã use case	UC08		Tên use case	Cập nhật hệ thống tìm kiếm
Tác nhân	Admin			
Mô tả	Use case mô tả quá trình cập nhật hệ thống tìm kiếm			
Tiền điều kiện	Đăng nhập			
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động	
	1	Admin	Click vào button Start ở giao	

			diện admin Update System
	2	Hệ thống	Thực hiện tính toán lại dữ liệu theo sự cập nhật của cơ sở dữ liệu từ điển và lưu vào file dưới dạng binary
	3	Hệ thống	Hiển thị thông báo cập nhật thành công
Luồng sự kiện thay thế	STT	Thực hiện bởi	Hành động
	2a	Hệ thống	Hệ thống hiển thị thông báo lỗi nếu xảy ra lỗi trong hệ thống
Hậu điều kiện	Không		

Bảng 9. Đặc tả use case UC08 « Cập nhật hệ thống tìm kiếm »

i. Đặc tả use case UC09 « Đăng xuất »

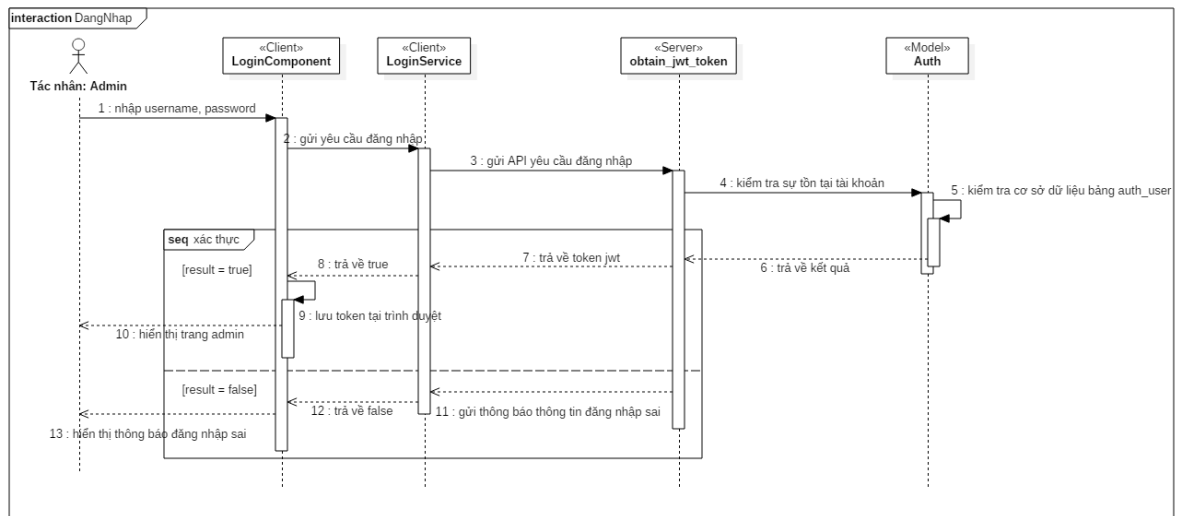
Mã use case	UC09	Tên use case	Đăng xuất
Tác nhân	Admin		
Mô tả	Use case mô tả quá trình đăng xuất của admin khi không muốn tham gia vào phiên làm việc của hệ thống.		
Tiền điều kiện	Admin đã đăng nhập vào hệ thống		
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Admin	Chọn chức năng đăng xuất
	2	Hệ thống	Kết thúc phiên làm việc của admin
	3	Hệ thống	Hiển thị trang chủ
Luồng sự kiện thay thế	Không có		
Hậu điều kiện	Không có		

Bảng 10. Đặc tả use case UC09 « Đăng xuất »

3.4. Thiết kế hệ thống

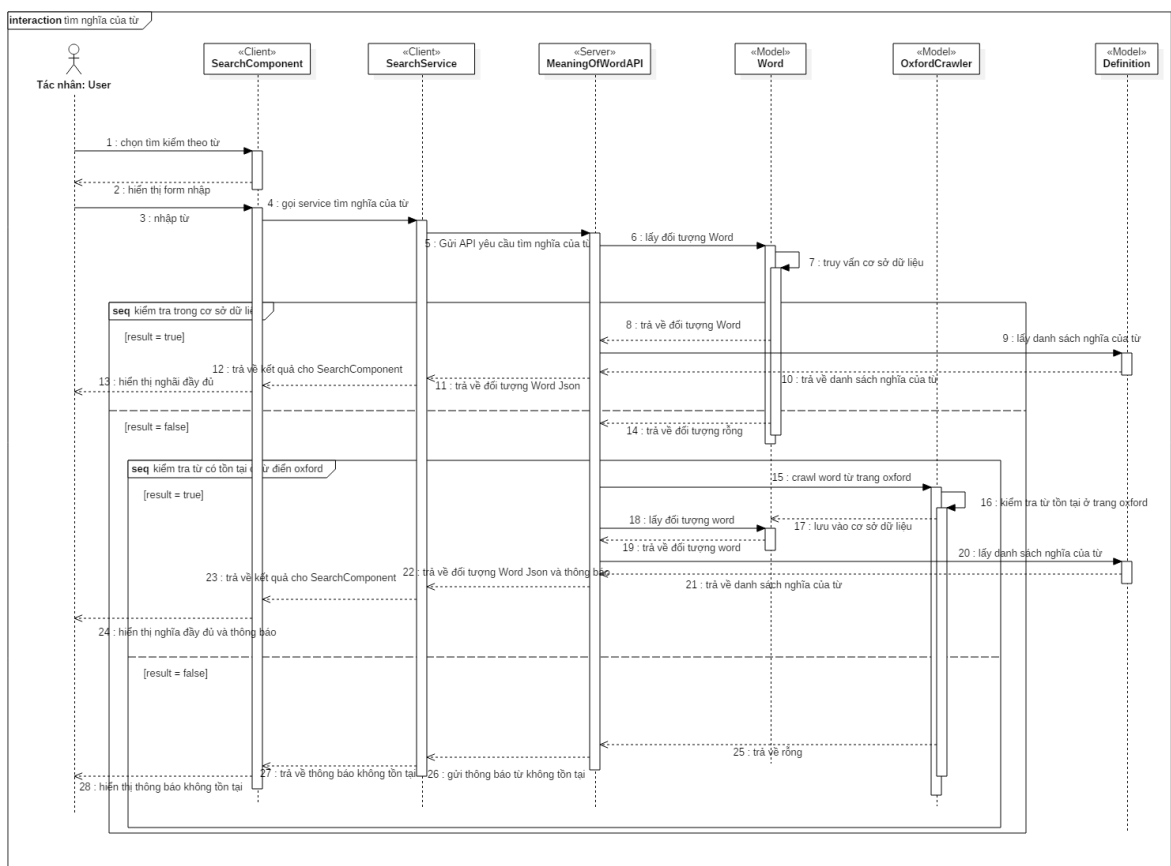
3.4.1. Biểu đồ trình tự

a. Biểu đồ trình tự đăng nhập



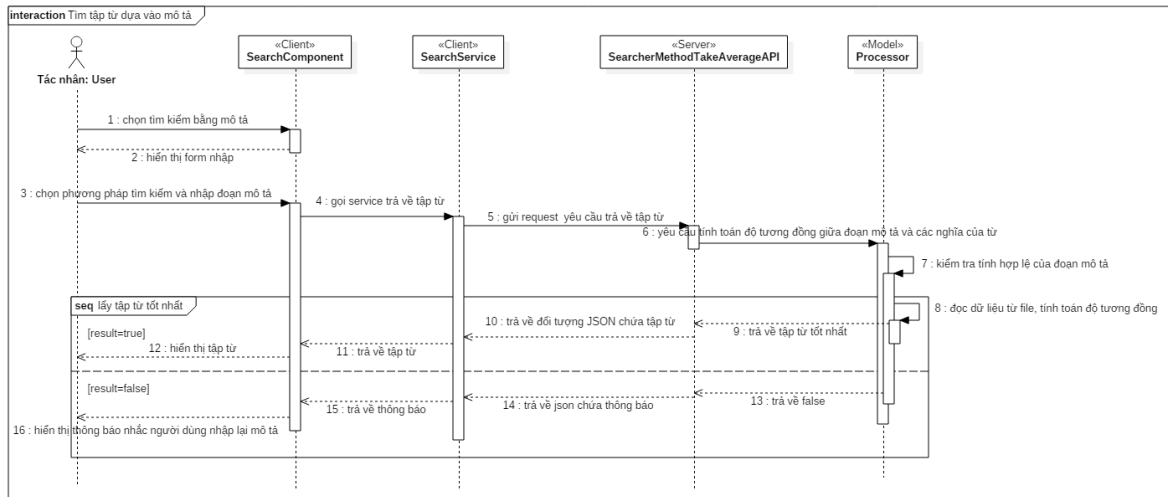
Hình 11. Biểu đồ trình tự đăng nhập

b. Biểu đồ trình tự tìm nghĩa của từ



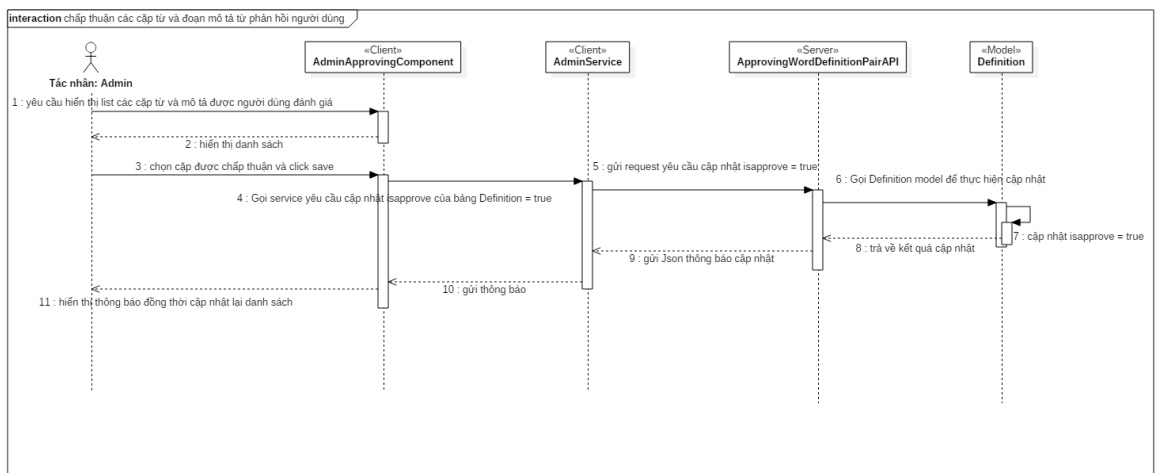
Hình 12. Biểu đồ trình tự tìm nghĩa của từ

c. Biểu đồ trình tự tìm tập từ dựa vào mô tả



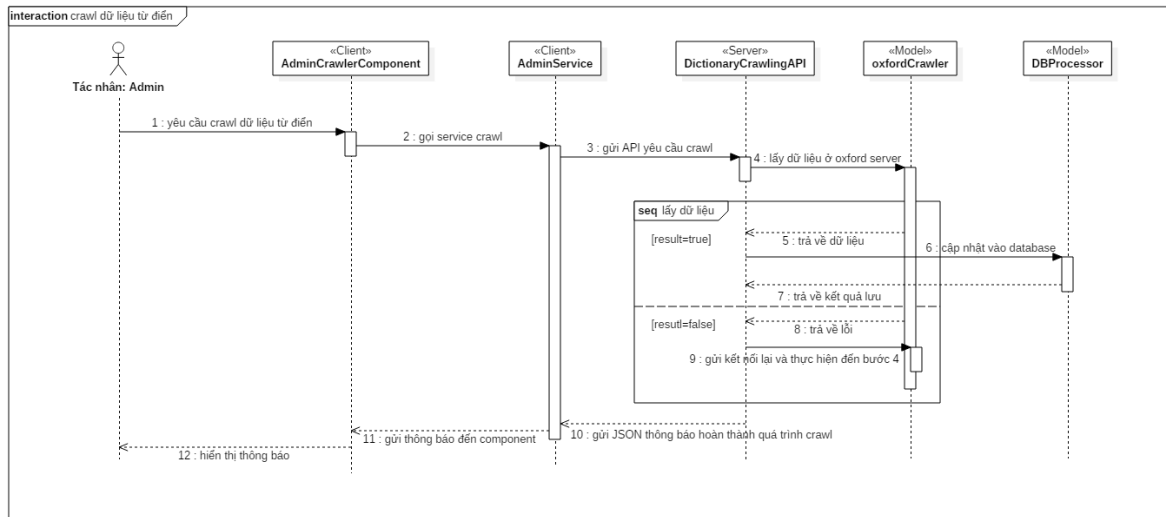
Hình 13. Biểu đồ trình tự tìm tập từ dựa vào mô tả

d. Biểu đồ trình tự chấp thuận các cặp từ và đoạn mô tả từ đánh giá của người dùng



Hình 14. Biểu đồ trình tự chấp thuận các cặp từ và đoạn mô tả từ đánh giá của người dùng

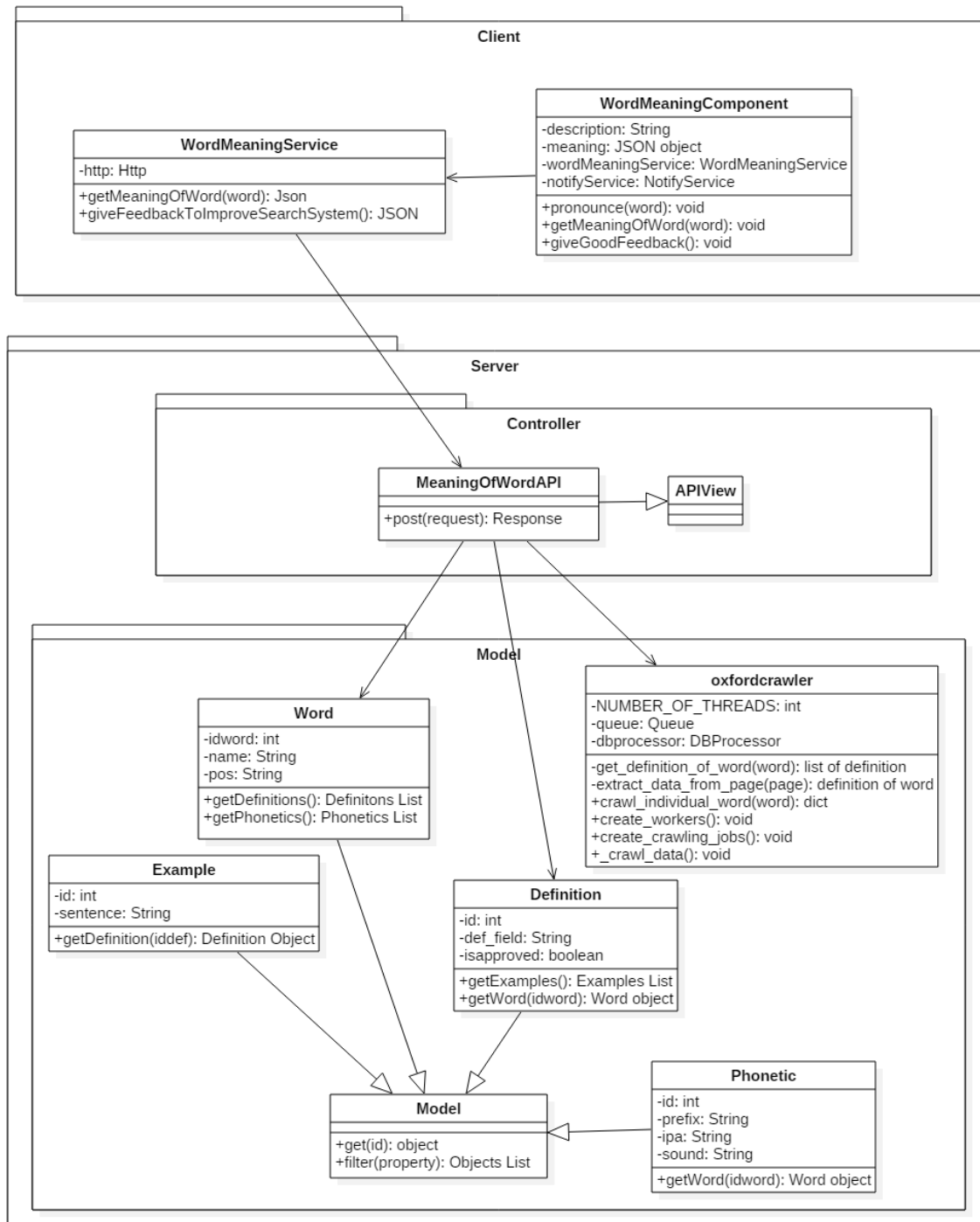
e. Biểu đồ trình tự crawl dữ liệu từ điển cho hệ thống



Hình 15. Biểu đồ trình tự crawl dữ liệu từ điển cho hệ thống

3.2.2. Thiết kế biểu đồ lớp

a. Biểu đồ lớp tìm nghĩa của từ



Hình 16. Biểu đồ lớp tìm nghĩa của từ

Mô tả biểu đồ lớp:

Hệ thống được thiết kế theo mô hình client server, client xử lý các nhiệm vụ ở phía client, client giao tiếp với server qua RESTFUL API.

Về phía Client sử dụng Angular 2, nên mỗi chức năng của hệ thống là một component. Ở chức năng tìm nghĩa của từ, Client gồm lớp WordMeaningComponent và lớp WordMeaningService.

- Lớp **WordMeaningService** có các thuộc tính và phương thức sau:
 - Thuộc tính *http*: đối tượng thuộc lớp Http, đây là lớp thực hiện gửi các Restful request như GET, POST, PUT, DELETE,...
 - Phương thức *getMeaningOfWord()*: có đầu vào là word, phương thức này sẽ gửi Restful request POST đến server yêu cầu nghĩa, ví dụ, cách phát âm cho từ.
 - Phương thức *giveFeedbackToImproveSearchSystem()*: gửi request tới Server để cập nhật vào cơ sở dữ liệu.
- Lớp **WordMeaningComponent** có các thuộc tính và phương thức sau:
 - Thuộc tính *word*: kiểu String, để hiển thị ra giao diện từ được tìm kiếm
 - Thuộc tính *description*: kiểu String, để hiển thị ra giao diện và làm đầu vào cho phương thức *giveGoodFeedback()*
 - Thuộc tính *wordMeaningService*: là đối tượng thuộc lớp *WordMeaningService*, dùng để thực hiện các request giao tiếp với Server qua Restful API.
 - Thuộc tính *notifyService*: là đối tượng lớp *NotifyService*, thực hiện chức năng hiển thị các thông báo cho người dùng.
 - Phương thức *getMeaningOfWord()*: đầu vào là thuộc tính word, có chức năng gọi WordMeaningService để lấy nghĩa, ví dụ, phát âm của từ.
 - Phương thức *pronounce()*: dùng để thực hiện phát ra âm của từ khi người dùng click vào.
 - Phương thức *giveGoodFeedback()*: phương thức này có chức năng là khi người dùng đánh giá từ trong tập từ trả về có nghĩa tương tự đoạn mô tả mà người dùng nhập vào thì nó sẽ gọi phương thức *giveFeedbackToImproveSearchSystem()* của lớp *WordMeaningService* để gửi request tới Server cập nhật vào cơ sở dữ liệu. Phương thức này gọi *notifyService* để hiển thị thông báo khi thực hiện xong.

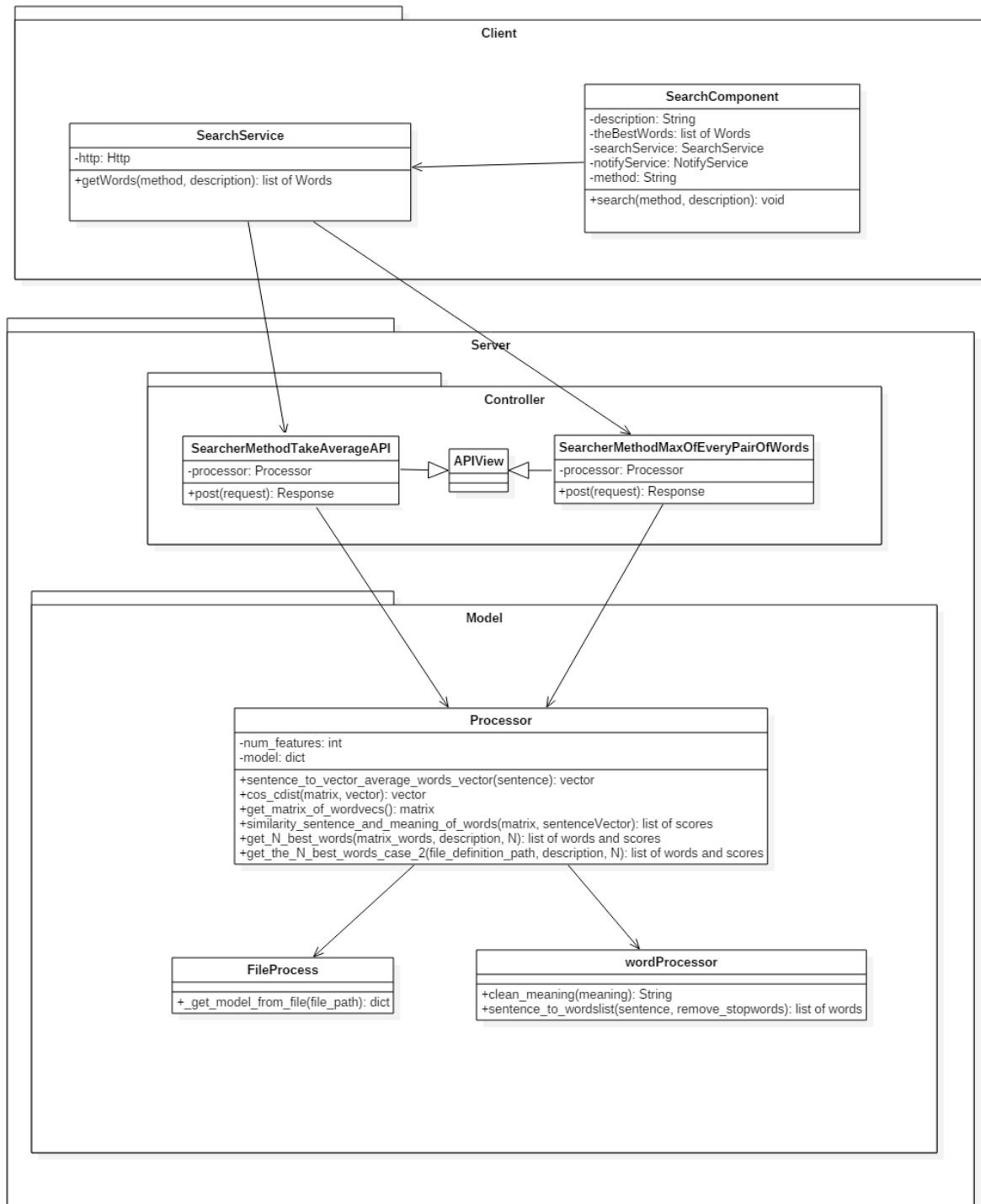
Về phía **Server** chia thành hai phần là Controller và Model. Controller gồm các lớp gọi các lớp trong Model, xử lý logic, trả về đối tượng Response đến Client. Model gồm các lớp giao tiếp với cơ sở dữ liệu, hoặc thực hiện tính toán. Ở chức năng tìm nghĩa của từ, Server có các lớp chính sau:

- Lớp **MeaningOfWordAPI**: thuộc thành phần Controller, là lớp kế thừa lớp APIView thuộc rest framework trong framework Django, có phương thức sau:
 - Phương thức *post(request)*: đây là phương thức kế thừa ở lớp APIView, thực hiện trả lời các request POST ở phía client. Phương thức này thực hiện gọi các lớp **Word**, **Definition**, **Example** ở thành phần Model để lấy dữ liệu ở cơ sở dữ liệu. Nếu trong cơ sở dữ liệu không tồn tại từ thì phương thức *post* sẽ thực hiện gọi đối tượng *oxfordCrawler* để crawl dữ liệu từ từ điển online Oxford. Sau đó nó sẽ trả về kết quả đối tượng Response chứa nghĩa của từ.

- **Lớp OxfordCrawler:** đây là lớp thực hiện chức năng crawl dữ liệu từ điển Oxford. Có các phương thức, thuộc tính chính sau:
 - o Thuộc tính *NUMBER_OF_THREADS*: kiểu int, số lượng Threads được sử dụng để thực hiện crawl dữ liệu song song, nhằm giảm thời gian crawling.
 - o Thuộc tính *Queue*: là đối tượng lớp Queue, được sử dụng để lưu trữ từ cần được crawl, theo cơ chế vào trước được crawl trước.
 - o Thuộc tính *Dbprocessor*: là đối tượng lớp DBProcessor, thực hiện các chức năng kết nối, xử lý với cơ sở dữ liệu.
 - o Phương thức *get_definitions_of_word()*: kiểu trả về danh sách các nghĩa của từ.
 - o Phương thức *extract_data_from_page()*: có chức năng trích rút các nghĩa, ví dụ, cách phát âm trong trang html.
 - o Phương thức *crawl_individual_word()*: crawl nghĩa cho một từ
 - o Phương thức *create_workers()*: kiểu void, khởi tạo các threads
 - o Phương thức *create_crawling_jobs()*: Để đưa các danh sách từ cần crawl vào queue.
 - o Phương thức *_crawl_data()*: kiểu void, sẽ chạy một vòng lặp for cho đến khi crawl xong tất cả các từ trong queue, nếu từ nào crawl lỗi do lỗi mạng thì từ đó sẽ tự động được thêm vào queue và được crawl lần sau.
- **Lớp Word:** thuộc thành phần Model, là lớp kế thừa lớp Model thuộc framework Django, lớp Word này sẽ map với bảng Word trong cơ sở dữ liệu, có các thuộc tính và phương thức chính sau:
 - o Thuộc tính *idword*: kiểu int, là duy nhất, thể hiện là id của đối tượng word.
 - o Thuộc tính *name*: kiểu String là tên của từ.
 - o Thuộc tính *pos*: kiểu String, viết tắt của Part of speech, là loại từ như danh từ, động từ.
 - o Phương thức *getDefinitions()*: kiểu trả về là danh sách các đối tượng Definition.
 - o Phương thức *getPhonetics()*: kiểu trả về là danh sách các đối tượng Phonetic.
- **Lớp Definition:** thuộc thành phần Model, là lớp kế thừa lớp Model, lớp **Defintion** map với bảng **Definition** trong cơ sở dữ liệu, có các thuộc tính và phương thức chính sau:
 - o Thuộc tính *id*: kiểu int, là duy nhất, thể hiện là id của đối tượng defintion.
 - o Thuộc tính *def_field*: kiểu String là nghĩa của từ.
 - o Thuộc tính *isapproved*: kiểu boolean, là true nếu là nghĩa đúng của từ, còn false nghĩa là chưa được admin chấp thuận (thuộc tính này cho chức năng đánh giá của người dùng về cặp từ và đoạn mô tả mà người dùng khi tìm kiếm).
 - o Phương thức *getExamples()*: kiểu trả về là danh sách các đối tượng Example.

- Lớp **Example**: thuộc thành phần Model, là lớp kế thừa lớp Model, lớp **Example** map với bảng **Example** trong cơ sở dữ liệu, có các thuộc tính và phương thức chính sau:
 - Thuộc tính *id* : kiểu int, là duy nhất, thể hiện là id của đối tượng example.
 - Thuộc tính *sentence* : kiểu String là câu ví dụ.
 - Phương thức *getDefinition()* : kiểu trả về là đối tượng Definition mà đối tượng Example thuộc.
- Lớp **Phonetic**: thuộc thành phần Model, là lớp kế thừa lớp Model, lớp **Phonetic** map với bảng **Phonetic** trong cơ sở dữ liệu, có các thuộc tính và phương thức chính sau:
 - Thuộc tính *id* : kiểu int, là duy nhất, thể hiện là id của đối tượng example.
 - Thuộc tính *prefix* : kiểu String, mô tả kiểu phát âm Anh Anh hay Anh Mỹ.
 - Thuộc tính *ipa* : kiểu String mô tả kiểu ipa phát âm Anh Anh hay Anh Mỹ.
 - Thuộc tính *sound* : kiểu String, là đường dẫn đến file mp3 để phát âm của từ.
 - Phương thức *getWord()* : kiểu trả về là đối tượng Word mà đối tượng Phonetic thuộc.

b. Biểu đồ lớp tìm tập từ dựa vào đoạn mô tả



Hình 17. Biểu đồ lớp tìm tập từ dựa vào mô tả

Mô tả biểu đồ lớp:

Về phía Client gồm các lớp chính sau:

- Lớp **SearchService** có các thuộc tính và phương thức sau:
 - Thuộc tính *http*: đối tượng thuộc lớp *Http*, đây là lớp thực hiện gửi các Restful request như GET, POST, PUT, DELETE,...

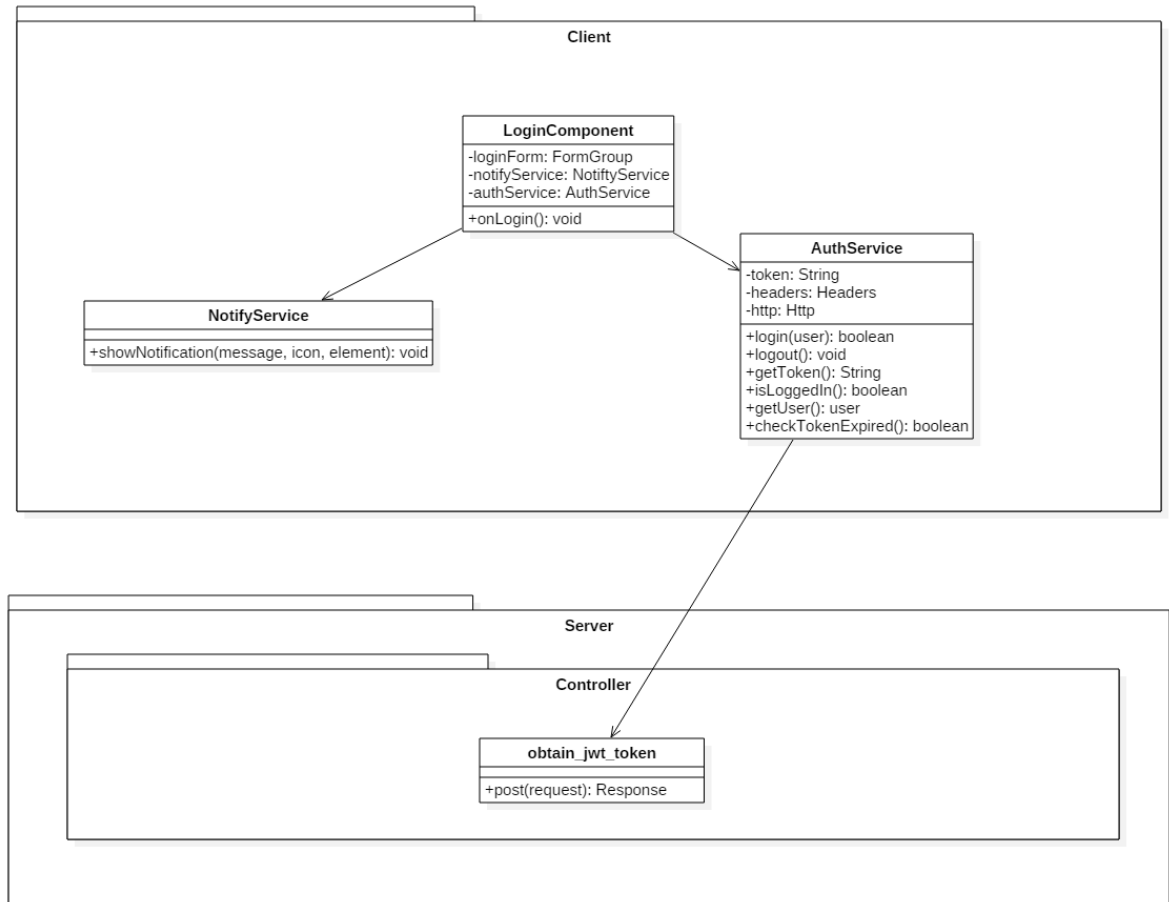
- Phương thức *getWords(method, description)*: là phương pháp tìm kiếm tập từ, *description* là đoạn mô tả, phương thức này sẽ gửi Restful request POST đến server yêu cầu tìm tập từ biểu diễn nghĩa tốt nhất cho đoạn mô tả.
- Lớp **SearchComponent** có các thuộc tính và phương thức sau :
 - Thuộc tính *description*: kiểu String, để lưu đoạn mô tả khi người dùng nhập vào đoạn mô tả.
 - Thuộc tính *theBestWords* : kiểu mảng String, để lưu trữ các từ tốt nhất mà phương thức *search()* trả về.
 - Thuộc tính *searchService*: là đối tượng thuộc lớp *SearchService*, dùng để thực hiện các request giao tiếp với Server qua Restful API.
 - Thuộc tính *notifyService*: được mô tả ở biểu đồ lớp tìm nghĩa của từ trên.
 - Phương thức *search(method, description)* : đầu vào là thuộc tính *method*, và *description*, có chức năng gọi *searchService* để gửi request đến server tính toán lấy ra tập từ biểu diễn nghĩa tốt nhất cho đoạn mô tả.

Về phía **Server** có các lớp chính sau:

- Lớp **SearcherMethodTakeAverageAPI**: thuộc thành phần Controller, là lớp kế thừa lớp *APIView* thuộc rest framework trong framework Django, có phương thức sau:
 - Phương thức *post(request)*: đây là phương thức kế thừa ở lớp *APIView*, thực hiện trả lời các request POST ở phía client. Phương thức này thực hiện gọi phương thức *get_N_best_words()* của đối tượng *processor* thuộc lớp *Processor* để lấy ra tập N từ tốt nhất biểu diễn nghĩa cho đoạn mô tả. Phương thức này trả về đối tượng *Reponse* chứa tập N từ đến client.
- Lớp **SearcherMethodMaxOfEveryPairOfWords**: là lớp thực hiện chức năng tương tự lớp **SearcherMethodTakeAverageAPI** chỉ khác cách tính toán ra tập từ. Lớp này có phương thức sau:
 - Phương thức *post(request)* : Phương thức này thực hiện gọi phương thức *get_the_N_best_words_case_2()* của đối tượng *processor* thuộc lớp *Processor* để lấy ra tập N từ tốt nhất biểu diễn nghĩa cho đoạn mô tả. Tương tự như phương thức *post* của lớp **SearcherMethodTakeAverageAPI**.
- Lớp **Processor**: đây là lớp thuộc thành phần Model, thực hiện các chức năng tính toán. Có các phương thức, thuộc tính chính sau:
 - Thuộc tính *Num_features*: kiểu int, số chiều của vector biểu diễn cho từ trong **word2vec**.
 - Thuộc tính *Model*: kiểu dictionary, có key là từ, còn value là vector biểu diễn của từ đó (được lấy từ dữ liệu **word2vec** training sẵn).
 - Phương thức *sentence_to_vector_average_words_vector(sentence)*: đầu vào là một câu, câu sẽ được xử lý thành tập các vector biểu diễn các từ trong câu, phương thức này thực hiện tính trung bình tập các vector đó để thành vector cho câu được đưa vào.
 - Phương thức *cos_cdist(matrix, vector)* : đầu vào là một matrix có các dòng là các vector biểu diễn nghĩa của từ trong từ điển, vector là vector biểu diễn đoạn mô tả. Phương thức này sử dụng thư viện *spicy* trong python để tính

- độ tương đồng cosine giữa matrix vector và vector, trả về danh sách khoảng cách giữa vector và các vector trong matrix.
- o Phương thức *get_matrix_of_wordvecs()* : đầu vào là tập các nghĩa của từ trong từ điển, chuyển đổi thành matrix tập các vector biểu diễn.
 - o Phương thức *calculate_similarity_sentence_and_meaning_of_words()* : tính toán độ tương đồng của đoạn mô tả và nghĩa của các từ trong từ điển để đưa ra một vector chứa điểm độ tương đồng.
 - o Phương thức *get_N_best_words(matrix, description, N=10)*: matrix là tập các vector nghĩa của từ trong từ điển, description là đoạn mô tả, N là số từ cần trả về, mặc định N = 10. Phương thức này trả về tập N từ tốt nhất biểu diễn nghĩa của đoạn mô tả.
 - o Phương thức *get_the_N_best_words_case_2(file_path, description, N=10)*: Phương thức này có chức năng tương tự phương thức *get_N_best_words()* trên, với *file_path* là đường dẫn lưu một dictionary có key là từ ký hiệu là X trong từ điển, value là tập các vector biểu diễn các từ trong nghĩa của từ X (mục đích lưu vào file là để giảm thời gian tính toán), *description* là đoạn mô tả mà người dùng nhập, N là số từ cần trả về, mặc định N = 10.
 - **Lớp FileProcessor**: thuộc thành phần Model, là lớp xử lý với file
 - o Phương thức *get_model_from_file(file_path)*: đầu vào là file_path đường dẫn đến nơi lưu dữ liệu word2vec training sẵn, kiểu trả về một dictionary có key là từ, value là vector biểu diễn từ đó.
 - **Lớp WordProcessor**: thuộc thành phần Model
 - o Phương thức *clean_meaning(meaning)* : đầu vào là một câu, có chức năng loại bỏ các ký tự đặc biệt, ký tự html, ...
 - o Phương thức *sentence_to_wordlist(sentence)* : đầu vào là một câu, có chức năng tách câu và loại bỏ từ dừng trong câu, mục đích cho việc tính độ tương đồng giữa hai câu.

c. Biểu đồ lớp cho chức năng đăng nhập của admin



Hình 18. Biểu đồ lớp cho chức năng đăng nhập

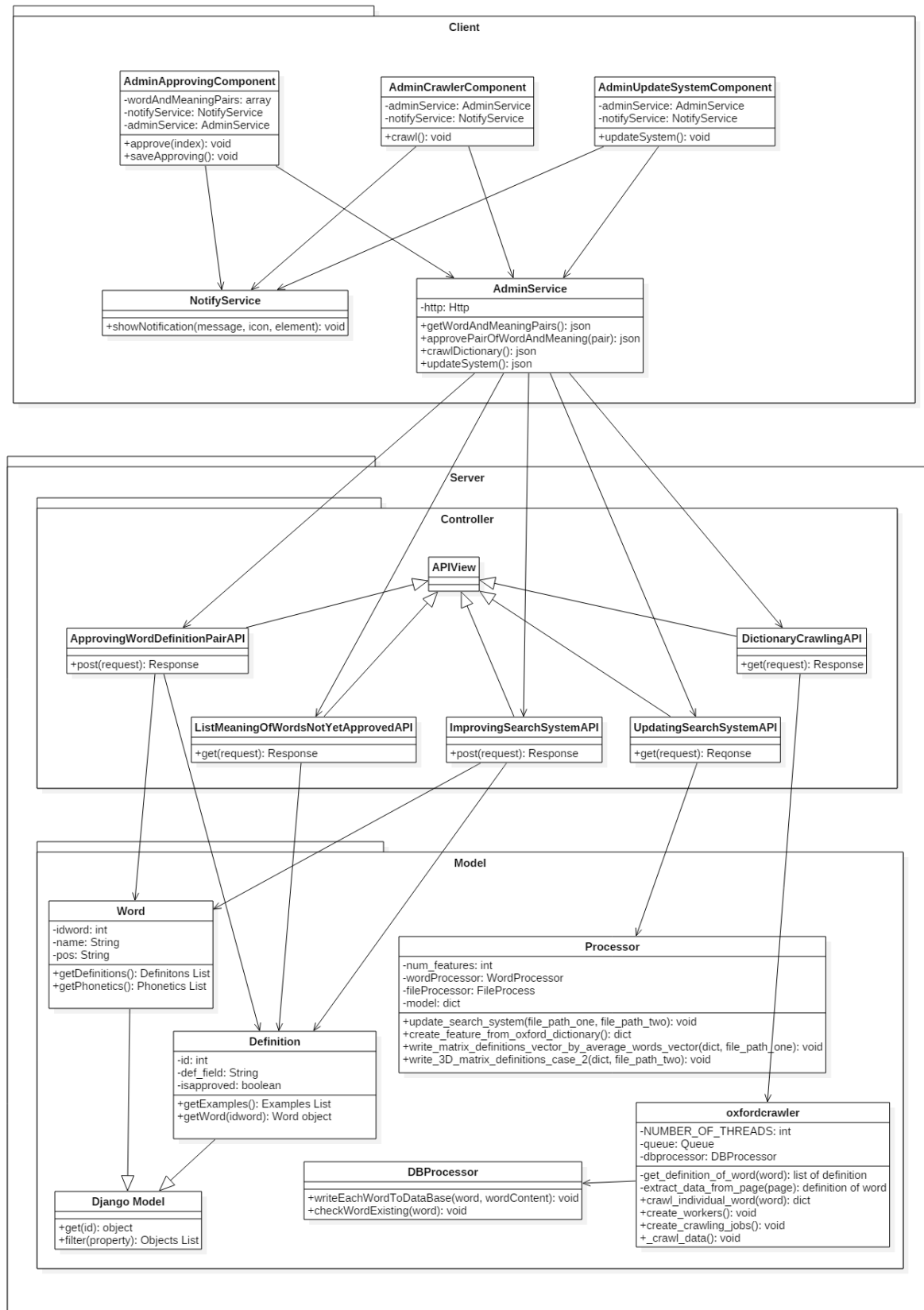
Về phía client gồm có lớp **LoginComponent**, **AuthService** và lớp **NotifyService**.

- Lớp **LoginComponent** gồm có các thuộc tính và phương thức chính sau:
 - Thuộc tính *loginForm*: là đối tượng của lớp **FormGroup** thuộc `angular.form` để lưu thông tin đăng nhập username và password.
 - Thuộc tính *notifyService*: là đối tượng thuộc lớp **NotifyService** để gọi chức năng hiển thị thông báo.
 - Thuộc tính *authService*: là đối tượng thuộc lớp **AuthService**
 - Phương thức *onLogin()*: kiểu `void`, là hàm gọi hàm `login` của đối tượng `authService` để xử lý đăng nhập.
- Lớp **AuthService** gồm có các thuộc tính và phương thức chính sau:
 - Thuộc tính *Token*: kiểu `String`, để lưu token khi xác thực bằng JSON Web Tokens
 - Thuộc tính *headers*: kiểu `Headers` để lưu thông tin headers của request.
 - Thuộc tính *http*: kiểu `http` để thực hiện các phương thức `http` `get`, `post`, `delete`, `put`, ... đến server.
 - Phương thức *login(user)*: kiểu `boolean`, đầu vào là thông tin đăng nhập của user thực hiện gửi request đến **obtain_jwt_token** tại server để xác minh tài khoản người dùng.
 - Phương thức *logout()*: kiểu `void` thực hiện đăng xuất, xóa session user.

- Phương thức *isLoggedIn()*: kiểu boolean kiểm tra session user còn tồn tại không.
- Phương thức *checkTokenExpired()* : kiểu boolean kiểm tra xem token còn hiệu lực không.

Về phía server có lớp **obtain_jwt_token** đây là lớp thuộc framework Django để thực hiện xác minh tài khoản người dùng bằng JSON Web Tokens.

d. Biểu đồ lớp cho các chức năng của Admin



Hình 19. Biểu đồ lớp cho các chức năng của Admin

Về phía client gồm có các lớp chính sau:

- Lớp **AdminApprovingComponent** gồm có các thuộc tính và phương thức chính sau:
 - Thuộc tính *wordAndMeaningPairs*: kiểu mảng các json chứa từ và nghĩa của nó.
 - Thuộc tính *notifyService*: đối tượng thuộc lớp *NotifyService* để gọi chức năng hiển thị thông báo.
 - Thuộc tính *adminService*: đối tượng thuộc lớp *AdminService*
 - Phương thức *saveApproving()*: kiểu void, là hàm gọi hàm *approvePairOfWordAndMeaning()* của đối tượng *adminService* cập nhật cặp từ và nghĩa trong cơ sở dữ liệu.
- Lớp **AdminService** gồm có các thuộc tính và phương thức chính sau:
 - Thuộc tính *http*: đối tượng lớp *Http* để thực hiện các phương thức http get, post, delete, put, ... đến server.
 - Phương thức *getWordAndMeaningPairs()*: kiểu json để gửi request đến server để lấy ra các cặp từ và đoạn mô tả tương ứng mà được người dùng đánh giá.
 - Phương thức *approvePairOfWordAndMeaning ()*: kiểu json, gửi request đến server để thực hiện cập nhật cặp từ và đoạn mô tả trong cơ sở dữ liệu.
 - Phương thức *crawlDictionary()*: kiểu json để gửi request thực hiện crawl dữ liệu từ điển.
 - Phương thức *updateSystem()* : kiểu json để gửi request đến server thực hiện cập nhật lại hệ thống tìm kiếm khi cơ sở dữ liệu anh anh thay đổi.

Về phía server có các lớp chính sau:

- Lớp **ApprovingWordDefinitionPairAPI**: thuộc thành phần Controller, là lớp kế thừa lớp *APIView* thuộc rest framework trong framework Django, có phương thức sau:
 - Phương thức *post(request)* : đây là phương thức kế thừa ở lớp *APIView*, thực hiện trả lời các request POST ở phía client. Phương thức này thực hiện cập nhật thuộc tính *isapprove* của đối tượng lớp *Definition* thành true để chấp thuận cặp từ và đoạn mô tả của người dùng đã đánh giá.
- Lớp **ListMeaningOfWordsNotYetApprovedAPI**: thuộc thành phần Controller, là lớp kế thừa lớp *APIView*. Lớp này có phương thức sau:
 - Phương thức *get(request)* : Phương thức này thực hiện ra tất cả các đối tượng *definition* mà có thuộc tính *isapprove=false*.
- Lớp **ImprovingSearchSystemAPI**: thuộc thành phần Controller, là lớp kế thừa lớp *APIView*. Lớp này có phương thức sau:
 - Phương thức *post(request)* : Phương thức này thực hiện lưu các cặp từ và đoạn mô tả được người dùng đánh giá.
- Lớp **UpdatingSearchSystemAPI**: thuộc thành phần Controller, là lớp kế thừa lớp *APIView*. Lớp này có phương thức sau:
 - Phương thức *get(request)* : Phương thức này thực hiện gọi hàm *update_search_system()* của đối tượng lớp *Processor* để cập nhật lại dữ liệu

các file cần thiết cho hệ thống search tính toán khi cơ sở dữ liệu từ điển thay đổi.

- Lớp **DictionaryCrawlingAPI**: thuộc thành phần Controller, là lớp kế thừa lớp APIView. Lớp này có phương thức sau:
 - o Phương thức *get(request)*: Phương thức này thực hiện gọi đối tượng oxfordCrawler để crawl dữ liệu từ điển oxford.
- Lớp **Processor**: đây là lớp thuộc thành phần Model, được mô tả ở phần b biểu đồ lớp tìm tập từ dựa vào mô tả, nhưng có các phương thức để thực hiện các chức năng của admin là:
 - o Phương thức *create_feature_from_oxford_dictionary()*: phương thức này kiểu dictionary để tạo mỗi từ tương ứng với một nghĩa.
 - o Phương thức *write_matrix_definitions_vector_by_average_words_vector(dict, file_path_one)*: đầu vào *dict* là dữ liệu trả về của hàm *create_feature_from_oxford_dictionary()*, *file_path_one* là đường dẫn file để lưu dữ liệu tính toán. Phương thức này sẽ thực hiện tính toán lại dữ liệu để lưu vào file dưới dạng binary cho phương pháp tìm tập từ thứ nhất (biểu diễn câu bằng vector trung bình của các vector của các từ trong câu).
 - o Phương thức *write_3D_matrix_definitions_case_2(dict, file_path_two)*: đầu vào *dict* là dữ liệu trả về của hàm *create_feature_from_oxford_dictionary()*, *file_path_two* là đường dẫn file để lưu dữ liệu tính toán. Phương thức này sẽ thực hiện tính toán lại dữ liệu để lưu vào file dưới dạng binary cho phương pháp tìm tập từ thứ hai (tính độ tương đồng giữa hai câu dựa vào tính độ tương đồng giữa các cặp từ).
 - o Phương thức *update_search_system(file_path_one, file_path_two)*: đầu vào *file_path_one*, *file_path_two* lần lượt là đường dẫn của file lưu các dữ liệu tính toán của các phương pháp tìm tập từ, phương thức này sẽ thực hiện gọi hàm *write_matrix_definitions_vector_by_average_words_vector()* và hàm *write_3D_matrix_definitions_case_2()* để lưu vào file dưới dạng binary.
- Lớp **OxfordCrawler**: đây là lớp thuộc thành phần Model, được mô tả ở phần a biểu đồ lớp tìm nghĩa của từ.
- Lớp **DBProcessor**: đây là lớp thuộc thành phần Model thực hiện chức năng xử lý cơ sở dữ liệu. Có các phương thức chính sau:
 - o Phương thức *writeEachWordToDataBase(word, wordContent)*: kiểu void, đầu vào gồm word và wordContent chứa nội dung của từ như nghĩa, cách phát âm, ví dụ. Phương thức này sẽ thực hiện lưu vào cơ sở dữ liệu thông tin của từ được crawl thành công.

- o Phương thức *checkWordExisting(word)* : kiểu boolean, đầu vào là word, phương thức này sẽ thực hiện kiểm tra xem từ đã tồn tại trong cơ sở dữ liệu chưa.

3.2.3. Thiết kế cơ sở dữ liệu

a. Các bảng trong cơ sở dữ liệu quan hệ

Các bảng lưu trữ cơ sở dữ liệu từ điển tiếng Anh

Bảng Word: lưu trữ dữ liệu các từ.

idword	Int	Khóa chính, tự động tăng, not null	Mã từ
name	Varchar(50)	Not null.	Tên của từ
pos	Varchar(50)		Part of speech. Ví dụ noun, adverb

Bảng 11. Mô tả bảng Word trong cơ sở dữ liệu

Bảng Definition: lưu trữ nghĩa của từ

id	Int	Khóa chính, tự động tăng, not null	Mã nghĩa
idword	int	khóa ngoài tham chiếu đến bảng word, not null	Mã từ mà nghĩa thuộc
def	Varchar(1000)		Nghĩa của từ
isapproved	Bit(1)		Được chấp thuận là nghĩa của từ hay không

Bảng 12. Mô tả bảng Definition trong cơ sở dữ liệu

Bảng Example: lưu trữ ví dụ của từ

id	Int	Khóa chính, tự động tăng, not null	Mã ví dụ
iddef	int	khóa ngoài tham chiếu đến bảng definition	Mã nghĩa mà ví dụ thuộc
sentence	Varchar(500)		Ví dụ của từ với mỗi nghĩa

Bảng 13. Mô tả bảng Example trong cơ sở dữ liệu

Bảng Phonetic: lưu trữ toàn bộ cách phát âm của từ

id	Int	Khóa chính, tự động tăng, not null	Mã phonetic
----	-----	------------------------------------	-------------

idword	int	khóa ngoài tham chiếu đến bảng word	Mã từ mà phonetic thuộc
prefix	Varchar(45)		Kiểu phát âm, anh anh hoặc anh mỹ
ipa	Varchar(45)		Kiểu đọc, anh anh hoặc anh mỹ.
Sound	Varchar(500)		Đường dẫn đến file mp3 phát âm.

Bảng 14. Mô tả bảng Phonetic trong cơ sở dữ liệu

Các bảng chính được sinh ra bởi framework Django

Bảng auth_user: lưu trữ thông tin người dùng trong hệ thống

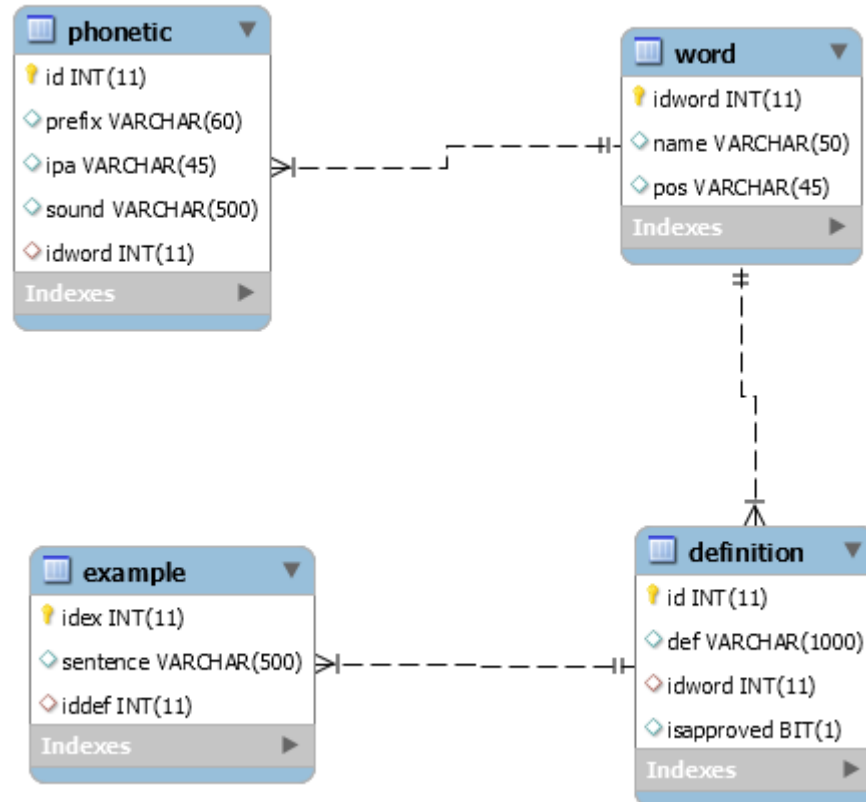
id	Int	Khóa chính, tự động tăng, not null	Mã người dùng
Password	Varchar(45)	not null	Mật khẩu của user được mã hóa trên hệ thống web
Last_login	DateTime		Thời điểm đăng nhập cuối
Is_superuser	TINYINT(1)	Not null	Biểu thị là admin hay không
Username	Varchar(30)	Not null	Tên tài khoản đăng nhập
Firstname	Varchar(30)		Tên của người dùng
Lastname	Varchar(30)		Tên họ của người dùng
Email	Varchar(254)	Not null	Email của người dùng
Is_staff	Tinyint(1)	Not null	Thể hiện người dùng có thể truy cập trang admin hay không
Is_active	Tinyint(1)	Not null	Thể hiện user có đang hoạt động hay không
Date_joined	DateTime		Ngày tham gia hệ thống

Bảng 15. Mô tả bảng auth_user trong cơ sở dữ liệu

b. Các liên kết giữa các bảng

- Liên kết giữa bảng **Word** với bảng **Definition** là liên kết 1-n vì 1 từ có nhiều nghĩa

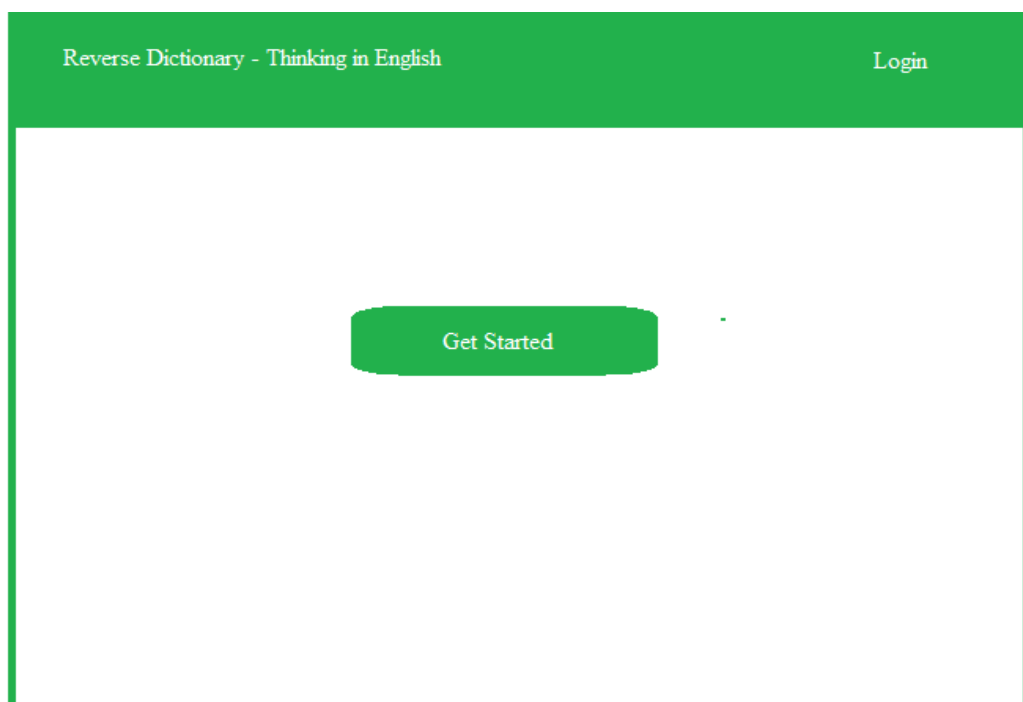
Các bảng lưu trữ dữ liệu từ điển Anh Anh



Hình 21. Mô hình liên hệ giữa các bảng lưu trữ dữ liệu từ điển Anh Anh của hệ thống

3.2.4. Thiết kế giao diện

a. Giao diện trang home



Hình 22. Thiết kế giao diện trang chủ

b. Giao diện trang tìm kiếm bằng mô tả

Reverse Dictionary - Thinking in English Login

dropdown

search by method Type thing to search Search

word basic

description advanced

The best words

word_1 word_2 word_3 word_4

word_5 word_6 word_7 word_8

Hình 23. Thiết kế giao diện trang tìm kiếm bằng mô tả

c. Giao diện trang tìm kiếm bằng từ

Reverse Dictionary - Thinking in English Login

dropdown

search by method Type thing to search Search

Word

BRE /word/ NAME /word/

1. The meaning 1
Example 1
Example 2

2. The Meaning 2

3. The Meaning 3
Example 1
Example 2

4. The meaning 4

Hình 24. Thiết kế giao diện trang tìm kiếm bằng từ

- d. Giao diện trang Admin chấp thuận đánh giá của người dùng

Approving	word	description	isapproved
Crawler	word_1	description_1	<input type="checkbox"/>
	word_2	description_2	<input type="checkbox"/>
Update System	<input type="checkbox"/>
	<input type="checkbox"/>
	<input type="checkbox"/>

Save

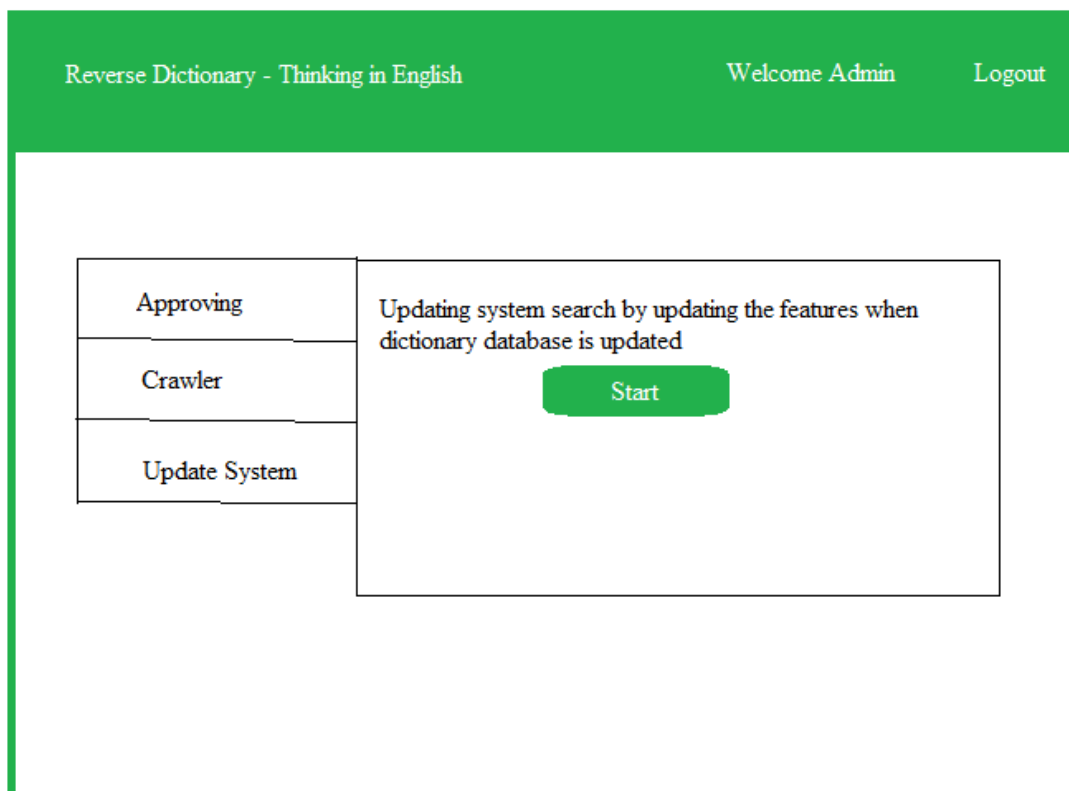
Hình 25. Thiết kế giao diện trang admin chấp thuận đánh giá của người dùng

- e. Giao diện trang admin chức năng Crawling

Approving	
Crawler	Crawl oxford dictionary to update our dictionary database Start
Update System	

Hình 26. Thiết kế giao diện admin chức năng crawling

f. Giao diện trang admin chức năng cập nhật hệ thống



Hình 27. Thiết kế giao diện trang admin cập nhật hệ thống

IV. Cài đặt và đánh giá hệ thống

4.1. Cài đặt

Môi trường phát triển ứng dụng:

- Hệ điều hành Ubuntu 14.04
- Ngôn ngữ lập trình: Python, Typescript, HTML, CSS
- Framework: Angular 2, Django, Bootstrap

4.2. Đánh giá hệ thống

Trong phần thực nghiệm, tôi sử dụng 5 bộ dữ liệu word2vec đã được training sẵn với hai mô hình CBOW và Skip-Gram khác nhau, với kích thước context $C=10$ với tất cả các mô hình và số chiều của vector biểu diễn từ khác nhau 100, 200, 300 được download tại trang zucon.net, các bộ dữ liệu word embedding này được sử dụng trong bài báo ADCS 2015 [11] có cấu trúc như hình 6 sau:

```
character -0.715851 -0.156755 0.093060 0.284060 -0.301335 -0.657832 -0.182967  
-0.330230 -0.140031 -0.097877 0.098202 -0.287551 -0.000513 0.027513 -0.433179
```

Hình 28. Ví dụ của một từ trong bộ dữ liệu word2vec training sẵn

Bộ dữ liệu để thực nghiệm với 150 cặp từ và đoạn mô tả nghĩa của từ đó được xây dựng bởi nhiều người thật có trình độ tiếng Anh khác nhau. Ví dụ:

a part of body	leg
an argument between two country	dispute
very attractive easy to love	adorable
main road between towns or city	highway
a large road which designed for a lot of vehicles with high speed	motorway
main road between towns or city	motorway
a thing that carry people or products between floors in the building	elevator
a machine that carry people or products between floors in the building	elevator
moving stairs that carry people between different floors	escalator
to force someone having sex even they do not want to	rape
to plan secretly together especially to do harm somebody	conspire
to ask organization or company for money or support	solicit
an amount of money given to somebody by an organization to study abroad	scholarship
a group members that was elected to govern a city or a country	council
a group members that was elected to make a law	congress
a group people that was elected to make a law	congress
a group people that was elected to make decision	committee
the possibility that something will happen	prospect
the possibility that something will happen	chance
very comfortable and rich	luxurious
a place to live	accommodation
a place to give people a protection	shelter
a person who receive the property from someone who died	beneficiary
a person who receive the property from someone who died	inheritor
a person who receive the property from someone who died	heir

Hình 29. Bộ dữ liệu để thực nghiệm

Bộ dữ liệu từ điển Anh Anh được sử dụng trong hệ thống là từ điển Oxford với gần 50.000 từ.

4.2.1. Môi trường thực nghiệm

Hệ thống được xây dựng bằng ngôn ngữ lập trình Python trên máy tính với cấu hình sau: CPU Core i5 3337U, tốc độ xử lý 1.80 GHz, bộ nhớ RAM 6Gb, hệ điều hành Ubuntu 14.04.

4.2.2. Kết quả đánh giá

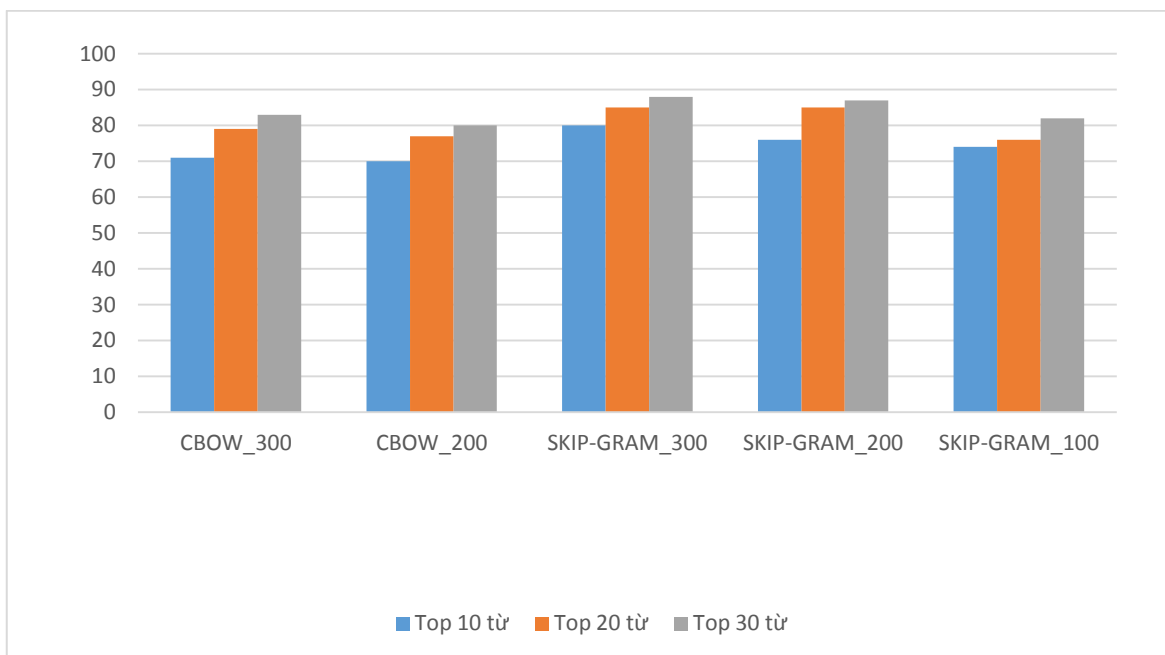
Hệ thống được đánh giá như sau:

- Sử dụng bộ dữ liệu thực nghiệm với 150 cặp từ và đoạn mô tả nghĩa của từ.
- Hệ thống sử dụng lần lượt các bộ dữ liệu word2vec training sẵn với kích thước context $C = 10$ (trong mô hình CBOW và Skip-Gram)
- Hệ thống trả về tập từ kết quả tương ứng với mỗi đoạn mô tả, nếu từ tương ứng với đoạn mô tả trong dữ liệu thử nghiệm thuộc tập từ trả về sẽ được tính là đạt yêu cầu. Số phần trăm đúng bằng số lần đạt yêu cầu chia cho số cặp từ và đoạn mô tả trong dữ liệu thử nghiệm.
- Số tập từ trả về lần lượt là 10 từ, 20 từ và 30 từ.

a. Đánh giá với phương pháp 1 lấy trung bình các vector biểu diễn từ trong câu

	Top 10 từ	Top 20 từ	Top 30 từ
CBOW_300	70.77%	79.22%	83.11%
CBOW_200	70.12%	76.62%	79.87%
SKIPGRAM_300	80.52%	85.06%	87.66%
SKIPGRAM_200	75.97%	85.06%	87.01%
SKIPGRAM_100	74.03%	78.57%	81.82%

Bảng 16. Kết quả đánh giá từng bộ dữ liệu word2vec training sẵn, kích thước context bằng 10 với phương pháp 1



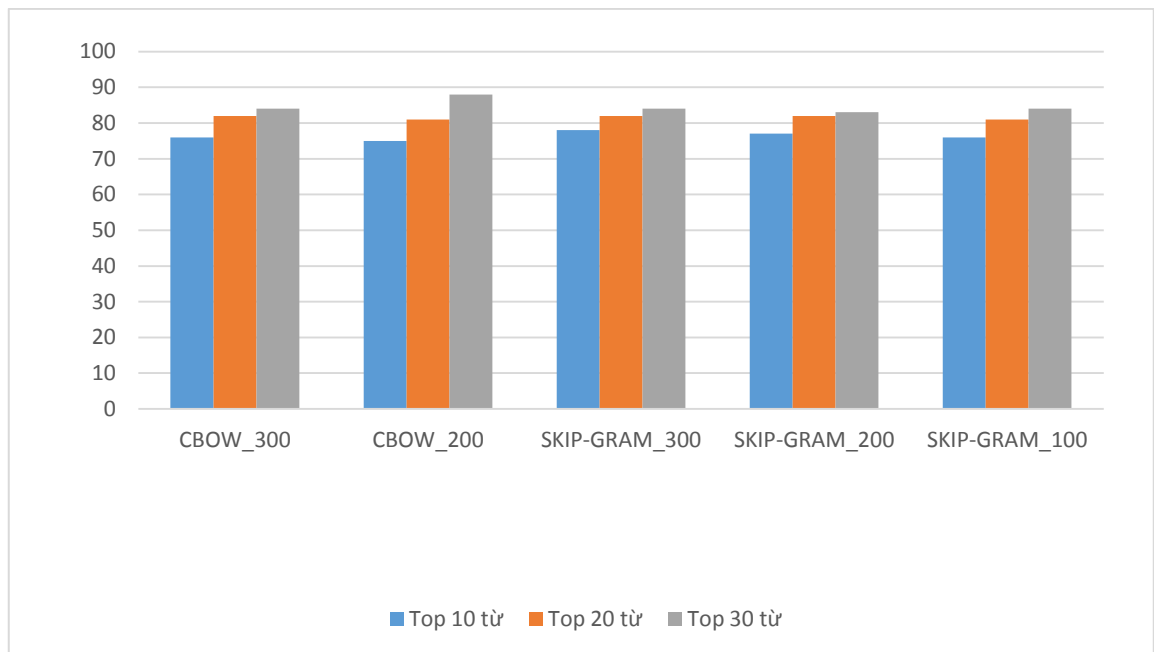
Biểu đồ 1: Đánh giá với từng bộ dữ liệu word2vec training sẵn với phương pháp 1.

Dựa vào bảng 1 và biểu đồ 1, hệ thống với phương pháp 1 sử dụng dữ liệu word2vec được training sẵn với mô hình Skip-Gram thì có kết quả cao hơn với mô hình CBOW, số chiều của vector biểu diễn từ càng lớn thì kết quả của hệ thống càng cao và số từ đưa ra càng nhiều thì tỷ lệ từ cần được hệ thống trả về xuất hiện trong tập từ càng cao.

b. Đánh giá với phương pháp 2 lấy trung bình max của độ tương đồng các cặp từ

	Top 10 từ	Top 20 từ	Top 30 từ
CBOW_300	75.97%	82.47%	84.41%
CBOW_200	74.67%	81.17%	87.66%
SKIPGRAM_300	77.92%	81.82%	83.77%
SKIPGRAM_200	76.62%	82.46%	83.11%
SKIPGRAM_100	75.97%	81.17%	84.41%

Bảng 17. Kết quả đánh giá từng bộ dữ liệu word2vec training sẵn, kích thước context bằng 10 với phương pháp 2



Biểu đồ 2: Đánh giá với từng bộ dữ liệu word2vec training sẵn với phương pháp 2.

Dựa vào biểu đồ 2 và bảng 2, với phương pháp 2 hệ thống đưa ra kết quả cao hơn với dữ liệu word2vec sử dụng mô hình CBOW.

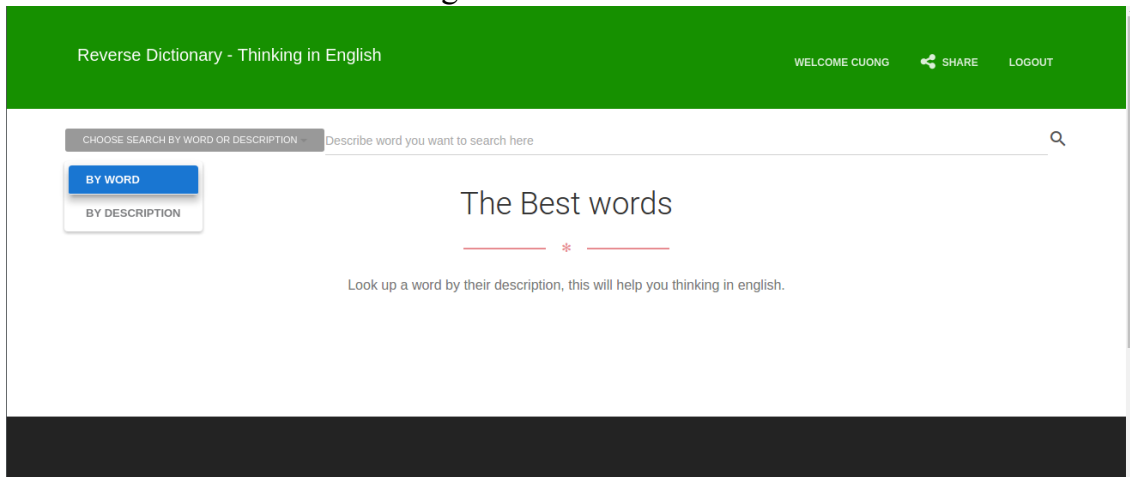
4.3. Kết quả đạt được

Giao diện trang chủ của hệ thống



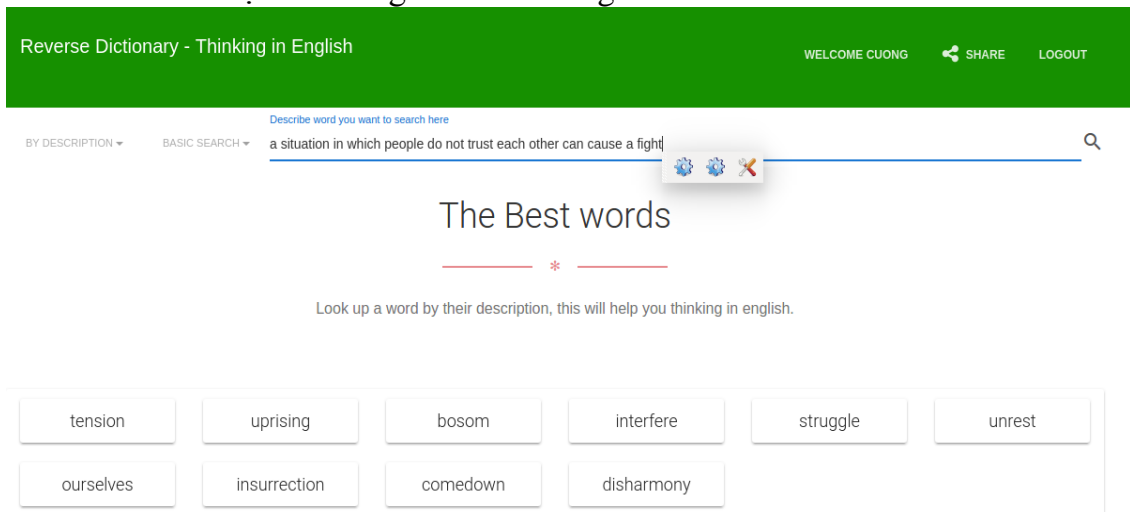
Hình 30. Giao diện trang chủ hệ thống

Màn hình bắt đầu của chức năng tìm kiếm



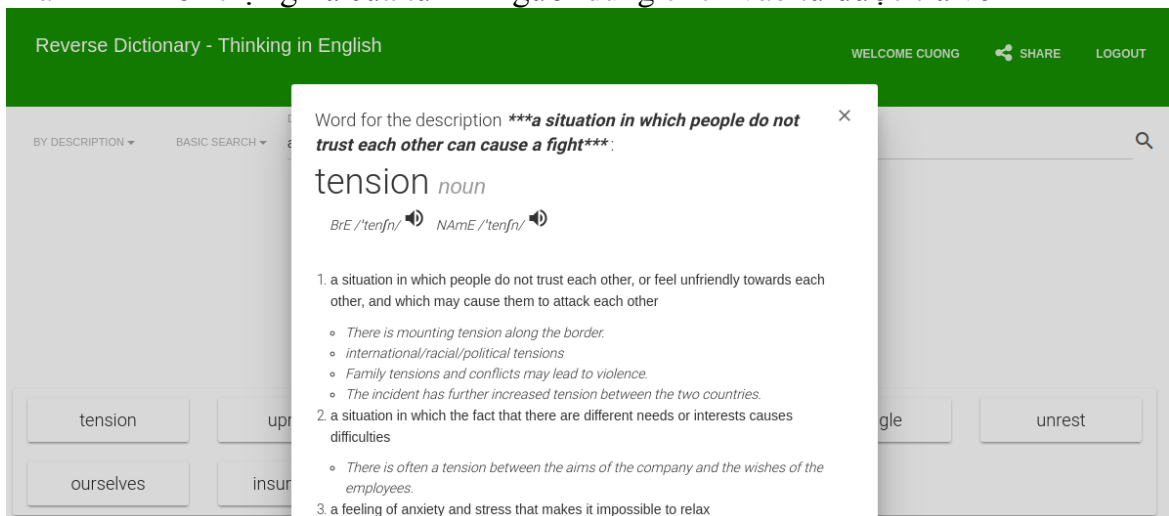
Hình 31. Giao diện trang tìm kiếm

Màn hình hiển thị chức năng tìm kiếm bằng mô tả



Hình 32. Giao diện chức năng tìm kiếm bằng mô tả

Màn hình hiển thị nghĩa của từ khi người dùng click vào từ được trả về



Hình 33. Giao diện chức năng hiển thị nghĩa của từ khi người dùng click vào từ được trả về

Màn hình hiển thị chức năng tìm kiếm nghĩa của từ

Look up a word
BY WORD ▾ hello 🔍

The Best words

Look up a word by their description, this will help you thinking in english.

hello *exclamation, noun*

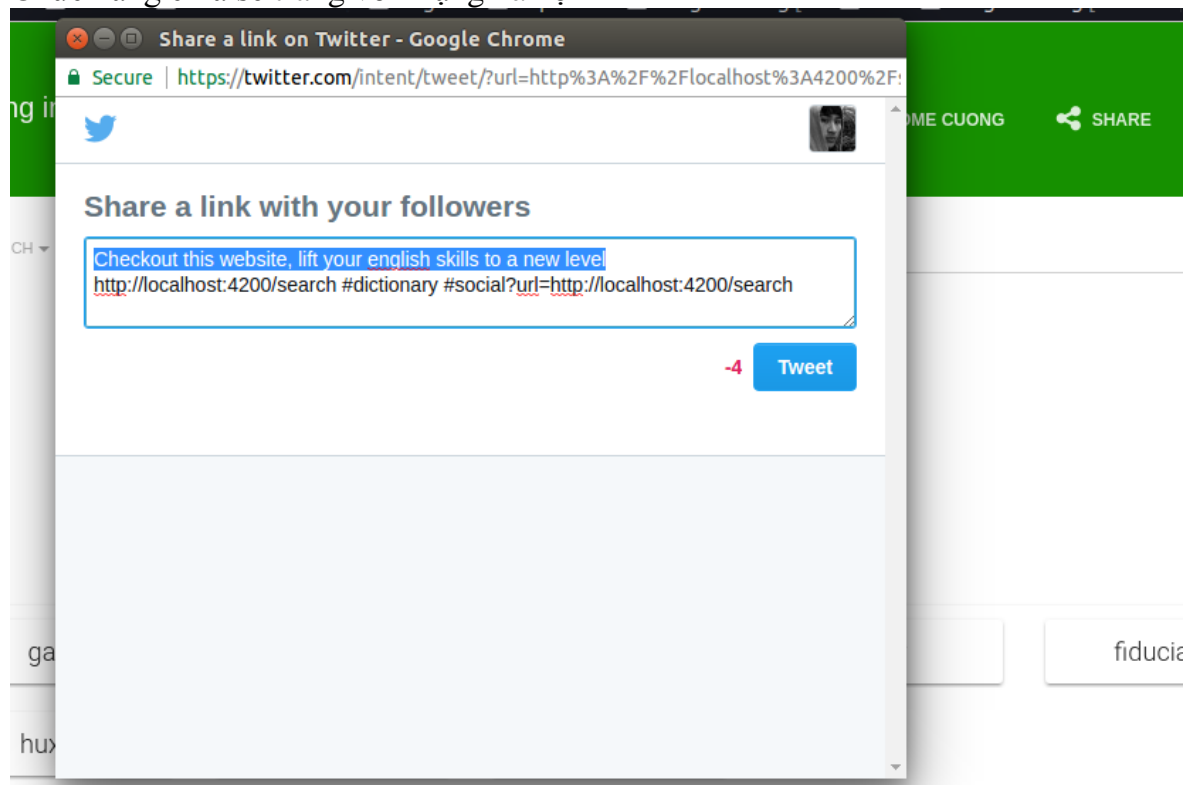
BrE /hə'ləʊ/ 🗣️ NAmE /hə'loʊ/ 🗣️

1. used as a greeting when you meet somebody, when you answer the telephone or when you want to attract somebody's attention

- Hello John, how are you?
- Hello, is there anybody there?
- Say hello to Liz for me.
- They exchanged hellos (= said hello to each other) and forced smiles.

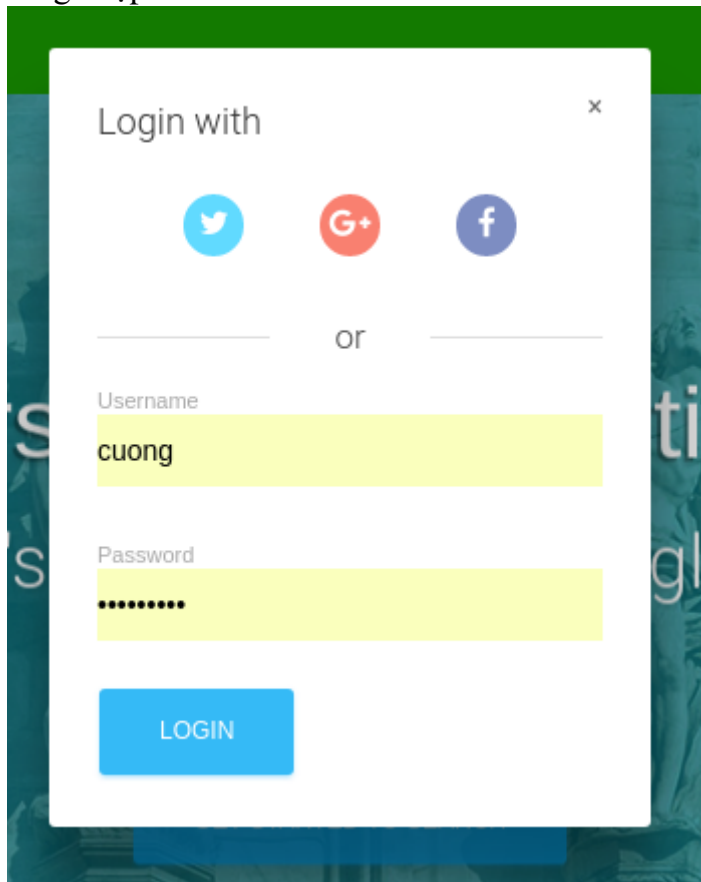
Hình 34. Màn hình hiển thị chức năng tìm kiếm nghĩa của từ

Chức năng chia sẻ trang với mạng xã hội



Hình 35. Giao diện chức năng chia sẻ mạng xã hội

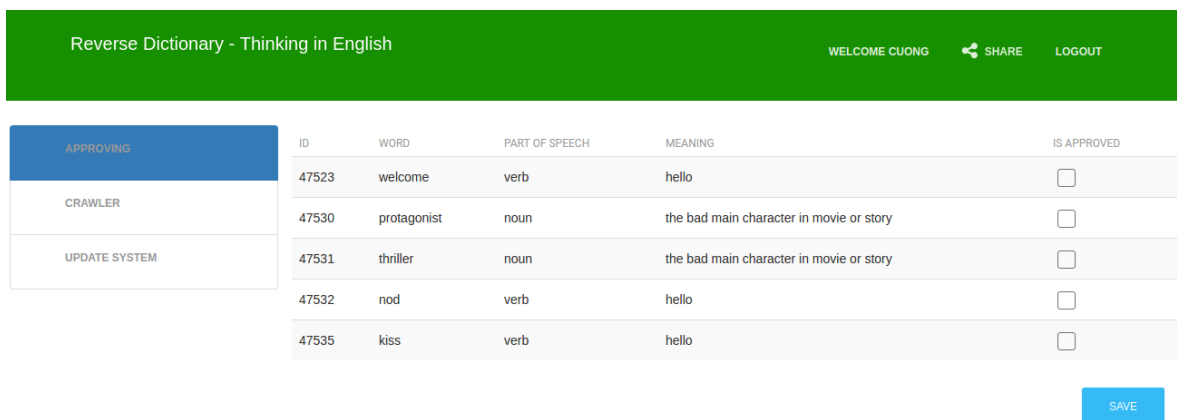
Giao diện chức năng đăng nhập



The login form is titled "Login with" and includes a close button (x). It offers three social media login options: Twitter, Google+, and Facebook. Below these, there is a separator line with the word "or". The form contains two input fields: "Username" with the value "cuong" and "Password" with masked characters. A blue "LOGIN" button is positioned at the bottom.

Hình 36. Giao diện chức năng đăng nhập

Giao diện trang Admin có các chức năng như chấp thuận đánh giá của người dùng để cập nhật vào cơ sở dữ liệu, chức năng crawl dữ liệu từ điển và chức năng cập nhật hệ thống tìm kiếm



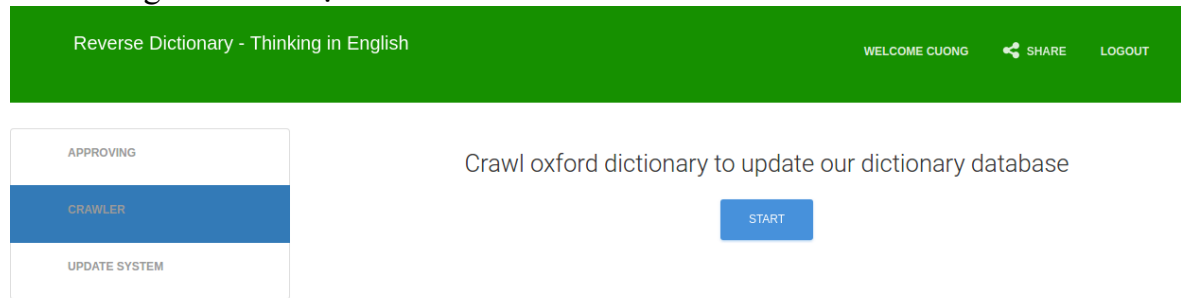
The admin interface is titled "Reverse Dictionary - Thinking in English". It includes a header with "WELCOME CUONG", "SHARE", and "LOGOUT". On the left, there is a sidebar with three buttons: "APPROVING" (highlighted), "CRAWLER", and "UPDATE SYSTEM". The main area displays a table of user reviews.

ID	WORD	PART OF SPEECH	MEANING	IS APPROVED
47523	welcome	verb	hello	<input type="checkbox"/>
47530	protagonist	noun	the bad main character in movie or story	<input type="checkbox"/>
47531	thriller	noun	the bad main character in movie or story	<input type="checkbox"/>
47532	nod	verb	hello	<input type="checkbox"/>
47535	kiss	verb	hello	<input type="checkbox"/>

A blue "SAVE" button is located at the bottom right of the table.

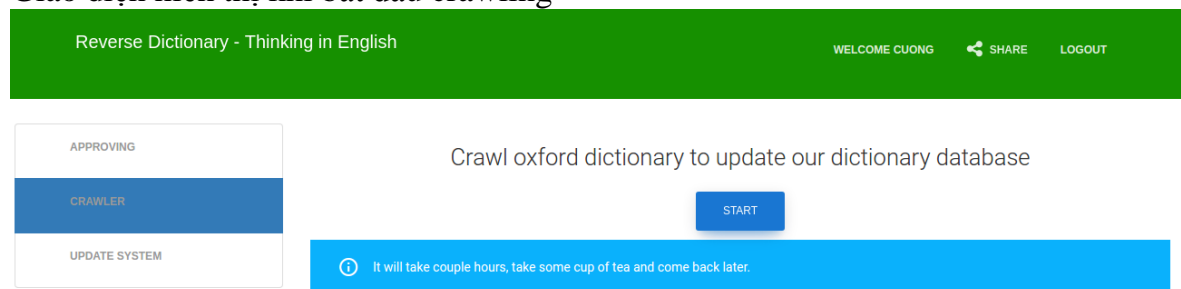
Hình 37. Giao diện admin chức năng chấp thuận đánh giá của người dùng

Chức năng crawl dữ liệu từ điển



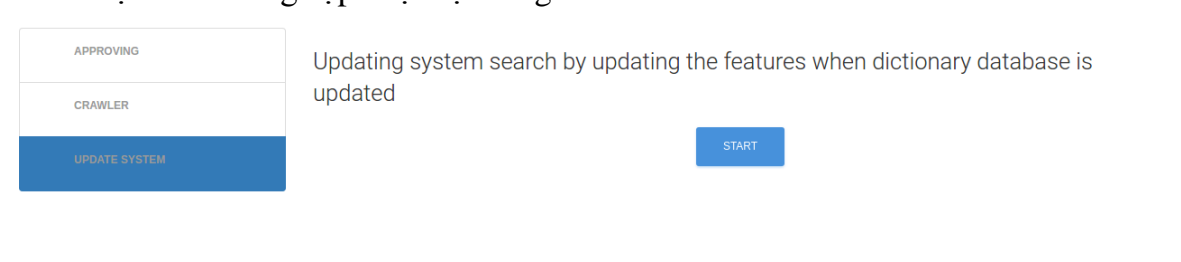
Hình 38. Giao diện chức năng crawling của admin

Giao diện hiển thị khi bắt đầu crawling



Hình 39. Giao diện chức năng hiển thị quá trình crawling

Giao diện chức năng cập nhật hệ thống



Hình 40. Giao diện hiển thị chức năng cập nhật hệ thống

V. Kết luận

Đồ án “Xây dựng hệ thống tìm tập từ tiếng Anh tốt nhất biểu diễn nghĩa của đoạn mô tả tiếng Anh” đã hoàn thành đầy đủ các nhiệm vụ được đặt ra. Hệ thống đã sử dụng các bộ dữ liệu word2vec đã được training sẵn và đề xuất hai phương pháp tính độ tương đồng giữa hai câu. Qua thực nghiệm, với cả hai phương pháp 1 và 2 thì hệ thống trả về kết quả khá tốt từ 70% đến 89% tùy vào bộ dữ liệu word2vec được sử dụng. Vì vậy có thể thấy hệ thống rất có tiềm năng phát triển. Bên cạnh nghiên cứu và thử nghiệm thì tôi cũng đã áp dụng để xây dựng hệ thống trên nền tảng Web với các công nghệ nổi bật như Angular 2, Django framework, RESTful API.

Nhận thấy việc trình bày mô tả trong tiếng Anh phụ thuộc nhiều vào trình độ mỗi người học nên tôi cũng đã xây dựng chức năng cho phép người dùng đánh giá những từ nào mà hệ thống trả về phù hợp đoạn mô tả của người dùng để từ đó người quản trị xem xét và cập nhật vào bộ dữ liệu từ điển. Từ đó nâng cao chất lượng trả về kết quả của hệ thống.

Hệ thống mới chỉ sử dụng các bộ dữ liệu word2vec training sẵn với số từ được train ít do hạn chế cấu hình máy. Vì vậy trong tương lai tôi muốn sử dụng với bộ dữ liệu word2vec của Google đã được train sẵn với 100 tỷ từ từ dữ liệu của Google News, để hệ thống trả về kết quả chính xác hơn.

Vì hệ thống mới chỉ được xây dựng trên nền tảng web nên trong tương lai tôi muốn xây dựng trên nền tảng di động để phục vụ nhiều người dùng hơn, tiện sử dụng hơn.

TÀI LIỆU THAM KHẢO

- [1] Tomas Mikolov, Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space.
- [2] Minh-Thang Luong, Richard Socher, and Christopher D. Manning. Better Word Representations with Recursive Neural Networks for Morphology
- [3] Y. Bengio, R. Ducharme, P. Vincent. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137-1155, 2003.
- [4] T. Mikolov. Language Modeling for Speech Recognition in Czech, Masters thesis, Brno University of Technology, 2007.
- [5] T. Mikolov, J. Kopecky, L. Burget, O. Glembek and J. Cernocky. Neural network based language models for highly inflective languages, In: Proc. ICASSP 2009.
- [6] T. Mikolov, W.T. Yih, G. Zweig. Linguistic Regularities in Continuous Space Word Representations. NAACL HLT 2013.
- [7] A. Zhila, W.T. Yih, C. Meek, G. Zweig, T. Mikolov. Combining Heterogeneous Models for Measuring Relational Similarity. NAACL HLT 2013.
- [8] Quoc Le, Tomas Mikolov. Distributed Representations of Sentences and Documents.
- [9] Tomas Mikolov, Ilya Sutskever, Kai Chen. Distributed Representations of Words and Phrases and their Compositionality.
- [10] Xin Rong ronxin@umich.edu - Word2vec Parameter Learning Explained
- [11] Word Embeddings Released with the ADCS 2015 paper – website <http://www.zucco.net/ntlm.html>, last visited May 2017
- [12] Multilayer Perceptron (MLP)
<https://www.analyticsvidhya.com/blog/2016/08/evolution-core-concepts-deep-learning-neural-networks/>, last visited May 2017
- [13] Angular architecture overview,
<https://angular.io/docs/ts/latest/guide/architecture.html>, last visited May 2017
- [14] Django web framework – Wikipedia
[https://en.wikipedia.org/wiki/Django_\(web_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)), last visited May 2017
- [15] RESTful web service là gì,
<http://www.codehub.vn/RESTful-Web-Services-La-Gi>, last visited May 2017