

ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

Tel. (84-511) 3736949, Fax. (84-511) 3842771
Website: itf.dut.udn.vn, E-mail: cntt@dut.udn.vn



BÁO CÁO DỰ ÁN CHUYÊN NGÀNH 2
ĐỀ TÀI
HỆ THỐNG DỰ BÁO XU HƯỚNG CỦA
CHỦ ĐỀ BÀI BÁO KHOA HỌC

SINH VIÊN THỰC HIỆN:

Nguyễn Văn Mạnh	LỚP: 20T1	NHÓM: 20.10
Nguyễn Văn Hoàng Phúc	LỚP: 20T1	NHÓM: 20.10
Nguyễn Công Cường	LỚP: 20T1	NHÓM: 20.10

GIẢNG VIÊN HƯỚNG DẪN: TS. Huỳnh Hữu Hưng

Đà Nẵng 06-2024

TÓM TẮT ĐỒ ÁN

Trong thời đại công nghệ số hiện nay, đề tài “dự báo xu hướng của chủ đề bài báo khoa học” đã trở thành một chủ đề quan trọng, giúp các nhà nghiên cứu, học giả và các tổ chức khoa học định hướng và phát triển nghiên cứu một cách hiệu quả hơn. Mục tiêu của đề tài là trích xuất từ khoá từ một bài báo khoa học nói chung và một đoạn văn bản nói riêng; cung cấp các sơ đồ, biểu đồ thể hiện các từ khoá (chủ đề) đang thịnh hành một cách trực quan; cho phép nghiên cứu sinh tìm kiếm thông qua mô tả về công việc muốn thực hiện, hiển thị danh sách các bài báo khoa học liên quan nhất; gợi ý các chủ đề đang thịnh hành và bài báo khoa học về chủ đề đó.

Đây là đề tài phục vụ cho môn học “PBL7: Dự án chuyên ngành 2”, nên tập trung tái xây dựng và cải thiện hướng kết hợp ưu điểm của các mô hình dựa trên ngôn ngữ và các mô hình dựa trên đồ thị để trích xuất từ khoá cho các abstract của bài báo khoa học. Chúng tôi dựa trên phương pháp đạt độ chính xác cao nhất trong chủ đề này là Phraseformer để tái xây dựng và đề xuất phương pháp BERTGraph với việc kết hợp giữa BERT Transformer và ExEM tự xử lý trên tập abstract. Kết quả thu được độ chính xác là 33.41% trên độ đo F1-score, kết quả cải thiện 3% so với việc chỉ sử dụng BERT thông thường và rút ra các kết luận về các bộ tham số huấn luyện và xây dựng mô hình.

Lời cuối, chúng em xin cảm ơn thầy Huỳnh Hữu Hưng đã có những góp ý, đánh giá, kiểm tra trong quá trình chúng em nghiên cứu và thực hiện đề tài, giúp cho đề tài ngày càng khắc phục được các nhược điểm và ngày càng cải thiện hơn. Một lần nữa, xin cảm ơn thầy.

Bảng 1-1 Bảng phân công nhiệm vụ

Sinh viên	Nhiệm vụ	Hoàn thành
Nguyễn Văn Mạnh	Phân công công việc, đảm bảo tiến độ đồ án	x
	Tìm kiếm và chọn bộ dữ liệu	x
	Đề xuất các mô hình KP + KW extraction	x
	Tìm hiểu và code kiến trúc BERT	x
	Kiểm thử model sau khi huấn luyện	x
	Hỗ trợ xây dựng Server (FE)	x
	Ghép nối và thử nghiệm sản phẩm	x
	Viết báo cáo	x
	Làm slide	x
Nguyễn Văn Hoàng Phúc	Tìm kiếm và chọn bộ dữ liệu	x
	Đề xuất các mô hình KP + KW extraction	x
	Tìm hiểu và code kiến trúc BERT	x
	Tìm hiểu và code kiến trúc Graph Embedding	x
	Kết hợp 2 kiến trúc BERT + Graph + FF	x
	Hỗ trợ xây dựng Server (BE)	x
	Ghép nối và thử nghiệm sản phẩm	x
	Viết báo cáo	x
	Làm slide	x
Nguyễn Công Cường	Tìm kiếm và chọn bộ dữ liệu	x
	Đề xuất các mô hình KP + KW extraction	x
	Tìm hiểu và code kiến trúc Graph Embedding	x
	Code inference demo cho model	x
	Kiểm thử model sau khi huấn luyện	x
	Hỗ trợ xây dựng Server (BE)	x
	Ghép nối và thử nghiệm sản phẩm	x
	Viết báo cáo	x
	Làm slide	x

MỤC LỤC

Chương 1. Giới thiệu	10
1.1. Thực trạng sản phẩm.....	10
1.2. Các công việc liên quan	11
1.3. Các vấn đề cần giải quyết	12
1.4. Đề xuất giải pháp tổng quan	12
Chương 2. Giải pháp trích xuất từ khoá	13
2.1. Học thông tin văn bản (Text learning):.....	14
2.2. Học cấu trúc (Structure learning).....	15
2.2.1. Xây dựng cặp từ và mối liên hệ	16
2.2.2. Xây dựng Co-occurrence network	17
2.2.3. Tìm tập chủ đạo (dominating set)	18
2.2.4. Xây dựng random walks	19
2.2.5. Huấn luyện model Skipgram + CBOW:	21
2.3. Đại diện từ cuối cùng (Final word representation)	22
2.4. Gắn thẻ trình tự và phân loại (Sequence labeling and classification).....	23
2.5. Tham số của mô hình	23
Chương 3. Phân tích và thiết kế hệ thống	24
3.1. Phân tích thiết kế hệ thống.....	24
3.1.1. Kiến trúc hệ thống	24
3.1.2. Mô tả các module trong hệ thống.....	24
3.2. Phân tích yêu cầu	25
3.3. Phân tích chức năng	26
3.3.1. Đối tượng sử dụng.....	26
3.3.2. Yêu cầu chức năng	26
3.4. Thiết kế cơ sở dữ liệu.....	27
3.4.1. Lược đồ cơ sở dữ liệu.....	27
3.4.2. Mô tả bảng người dùng “users”	28

3.4.3.	Mô tả bảng tracking “trackings”	29
3.4.4.	Mô tả bảng đặt lại mật khẩu “password_resets”	29
3.5.	Xây dựng hệ thống.....	29
3.5.1.	Tổng quan về hệ thống	29
3.5.2.	Quy trình tracking và huấn luyện trong hệ thống	30
Chương 4.	Kết quả.....	31
4.1.	Tập dữ liệu	31
4.2.	Các tham số trong quá trình huấn luyện	32
4.3.	Baseline model:.....	33
4.4.	Quá trình huấn luyện.....	33
4.5.	Kết quả	33
4.6.	Kết quả ứng dụng.....	35
4.6.1.	Chức năng đăng nhập, đăng kí	35
4.6.2.	Chức năng quên mật khẩu	35
4.6.3.	Chức năng chỉnh sửa thông tin cá nhân	36
4.6.4.	Chức năng gợi ý bài báo.....	36
4.6.5.	Chức năng tìm kiếm	37
4.6.6.	Chức năng thống kê top 10 từ khoá xu hướng	39
4.6.7.	Chức năng xem bảng xếp hạng xu hướng các từ khoá	40
4.6.8.	Chức năng tracking lịch sử người dùng	41
4.6.9.	Chức năng tracking để crawl dữ liệu huấn luyện.....	42
4.6.10.	Chức năng quản lý người dùng	43
Chương 5.	Kết luận và hướng phát triển.....	44
5.1.	Kết quả đạt được:	44
5.2.	Hướng phát triển	44

MỤC LỤC HÌNH ẢNH

Hình 2.1 Kiến trúc tổng quan của mô hình BERTGraph	13
Hình 2.2 The Transformer model architecture	14
Hình 2.3 The BERT Transformer architecture	15
Hình 2.4 Sơ đồ khối tổng quan mô quá trình học cấu trúc	16
Hình 2.5 Hình vẽ đồ thị graph cho 1/12 tập dữ liệu huấn luyện	18
Hình 2.6 Một ví dụ của một đồ thị có 6 nodes, 8 edges và tập chủ đạo của nó.	19
Hình 2.7 Ví dụ cho random walks	20
Hình 2.8 Kiến trúc Skipgram + CBOW	21
Hình 2.9 Ví dụ cho các cặp đầu vào và đầu ra của mô hình Skipgram + CBOW	22
Hình 3.1 Lược đồ cơ sở dữ liệu	28
Hình 3.2 Tổng quan về hệ thống	30
Hình 3.3 Tổng quan quy trình tracking và huấn luyện	30
Hình 4.1 Hình ảnh một mẫu dữ liệu Hình ảnh một mẫu dữ liệu	32
Hình 4.2 Hình ảnh huấn luyện 10 epochs trên tập dữ liệu	33
Hình 4.3 Chức năng đăng nhập	35
Hình 4.4 Chức năng quên mật khẩu	36
Hình 4.5 Chức năng chỉnh sửa thông tin cá nhân	36
Hình 4.6 Chức năng gợi ý bài báo	37
Hình 4.7 Giao diện ban đầu của trang tìm kiếm	37
Hình 4.8 Tìm kiếm theo đoạn mô tả	38
Hình 4.9 Model sẽ trích xuất các keyword trong đoạn mô tả	38
Hình 4.10 Search bởi một keyword duy nhất	39
Hình 4.11 Tìm kiếm theo thông tin khác	39

Hình 4.12 Thống kê top 10 từ khoá xu hướng	40
Hình 4.13 Bảng xếp hạng xu hướng từ khoá và wordcloud.....	40
Hình 4.14 Biểu đồ xu hướng top 10 từ khoá	41
Hình 4.15 Khi người dùng tìm kiếm và chọn vào xem bài báo	41
Hình 4.16 Lịch sử tìm kiếm các từ khoá của người dùng	42
Hình 4.17 Dữ liệu tracking thu được để huấn luyện	42
Hình 4.18 Chức năng quản lý người dùng	43

MỤC LỤC BẢNG

Bảng 1-1 Bảng phân công nhiệm vụ	3
Bảng 3-1 Kiến trúc hệ thống	24
Bảng 3-2 Mô tả bảng người dùng “users”	28
Bảng 3-3 Mô tả bảng tracking “trackings”	29
Bảng 3-4 Mô tả bảng đặt lại mật khẩu “password_resets”	29
Bảng 4-1 Số liệu thống kê của ba tập dữ liệu.....	31
Bảng 4-2 Kết quả BERT và BERTGraph (F1-score).....	34

DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
KP	Keyparse
KW	Keyword
BERT	Bidirectional Encoder Representations from Transformers
BE	Backend
FE	Frondend
WV	Word2Vec
FT	FastText

Chương 1. Giới thiệu

Phần này sẽ tập trung vào việc phân tích thực trạng, vấn đề liên quan và cần giải quyết của các phương pháp trích xuất keyword áp dụng trong dự báo xu hướng, đồng thời đề xuất các giải pháp tổng quan được thực hiện trong bài báo.

1.1. Thực trạng sản phẩm

Trong thời đại công nghệ số hiện nay, việc dự báo xu hướng của chủ đề bài báo khoa học đã trở thành một lĩnh vực quan trọng, giúp các nhà nghiên cứu, học giả và các tổ chức khoa học định hướng và phát triển nghiên cứu một cách hiệu quả hơn. Hệ thống dự báo xu hướng chủ đề bài báo khoa học (gọi tắt là Hệ thống) là công cụ tiên tiến giúp phân tích, tổng hợp và dự đoán các xu hướng nghiên cứu khoa học trong tương lai dựa trên dữ liệu hiện có. Để dự báo xu hướng, tức là các từ khóa được nhắc đến nhiều nhất thì công việc cụ thể là tìm ra phương pháp trích xuất từ khóa hoặc khóa ngữ hiệu quả.

Hiện nay số lượng các tài liệu văn bản ngày càng lớn trên các phương tiện truyền thông xã hội. Do đó, việc tìm kiếm và tóm tắt tinh gọn và hiệu quả đối với tất cả các tài liệu này là vô cùng quan trọng [1]. Phương pháp trích xuất từ khóa hoặc khóa ngữ là một giải pháp để xác định một tập hợp các thuật ngữ có thể kết luận ý chính của một tài liệu [2]. Quy trình này giúp người đọc nhanh chóng nắm bắt nội dung của một tài liệu. Phương pháp trích xuất từ khóa có thể được sử dụng hiệu quả bởi nhiều ứng dụng khai thác dữ liệu văn bản như phân loại, tóm tắt, phát hiện chủ đề và theo dõi chủ đề, cụm từ và phân loại [3,4].

Nhiều nghiên cứu đã được thực hiện để giải quyết vấn đề trích xuất từ khóa. Một số phương pháp tập trung vào nội dung của tài liệu để thu được từ khóa. Một số sử dụng phân tích ngữ pháp và ngữ nghĩa. Các phương pháp khác sử dụng thống kê số liệu như tần suất từ (TF) hoặc tần suất từ-đảo ngược tần suất tài liệu (TF-IDF) [5]. Mặt khác, các phương pháp dựa trên đồ thị tạo thành một nhóm phương pháp bằng cách xây dựng đồ thị các từ. Trong lớp này, các nút trung tâm nhất chỉ ra các từ khóa [1]. Ngoài ra, còn có các mô hình kết hợp sử dụng kết hợp các phương pháp dựa trên nội dung và đồ thị để chọn từ khóa. Tóm lại, cách kết hợp phương pháp nào là phù hợp và mạnh mẽ nhất để trích xuất từ khóa vẫn còn là chủ đề đang được thảo luận.

1.2. Các công việc liên quan

Các kỹ thuật xuất sắc hiện tại để trích xuất từ khóa có thể được chia thành ba nhóm: dựa trên văn bản, dựa trên đồ thị và mô hình kết hợp. Các phương pháp dựa trên văn bản tạo ra các từ khóa trực tiếp từ văn bản gốc bằng cách áp dụng các kỹ thuật xử lý ngôn ngữ tự nhiên. Trong khi đó, các phương pháp dựa trên đồ thị chuyển đổi tài liệu thành một đồ thị xuất hiện, trong đó các nút đại diện cho các từ và các cạnh cho thấy mối quan hệ giữa hai từ trong ngữ cảnh cửa sổ. Mặt khác, các mô hình kết hợp tận dụng cả đại diện văn bản và đồ thị của tài liệu để phát hiện các từ khóa. Trong những đoạn tiếp theo, chúng tôi sẽ nghiên cứu kỹ hơn ba phân loại này.

Trong mô hình dựa trên văn bản, mục tiêu là tạo ra các từ khóa trực tiếp từ văn bản gốc [5]. Mô hình đơn giản nhất trong phân loại này sử dụng kỹ thuật TF-IDF để trích xuất từ khóa. Sau đó, các nghiên cứu tập trung vào các phương pháp học máy để huấn luyện một bộ phân loại nhằm bắt các từ khóa. Với sự ra đời của các phương pháp học sâu như Mạng nơ-ron tích chập, RNN và Transformers đã trở thành giải pháp phổ biến cho bài toán này. Có một số phương pháp dựa trên văn bản bao gồm KEA [6], KP-Miner [7], WINGNUS [8], RAKE [9], YAKE [10], TNT-KID [11], [12, 13, 14, 15, 16].

Ý tưởng chính của các phương pháp dựa trên đồ thị là xây dựng một đồ thị xuất hiện từ các tài liệu. Đồ thị này minh họa tương tác của các từ trong tổng hợp. Trong đồ thị này, các từ đại diện cho các nút và có một cạnh giữa hai từ nếu chúng xuất hiện cùng nhau trong một cửa sổ. Sau khi xây dựng đồ thị xuất hiện, một số phép đo trung tâm như độ, độ gần, trung gian và vector riêng được áp dụng để tìm chuỗi từ khóa. Trong các phương pháp này, các từ khóa được xác định bởi các nút trung tâm nhất. Một số phương pháp bao gồm TextRank [17], CollabRank [18], DegExt[19], NE-Rank [20], TopicRank [21], Positionrank [22], M-GCKE [5], [23, 24, 25, 26, 27, 1] sử dụng lý thuyết đồ thị để chọn từ khóa.

Các mô hình kết hợp cố gắng kết hợp hai phân loại trước đó. Những mô hình này tính toán điểm cho từ từ cả đồ thị xuất hiện và nội dung tài liệu. Các phương pháp khác nhau sử dụng các cách khác nhau để kết hợp những điểm này. Tác giả trong các nghiên cứu [28, 29, 30, 31] đã đề xuất các phương pháp kết hợp. Các nghiên cứu [32, 33, 34, 35, 36] đã tiến hành rà soát các kỹ thuật trích xuất từ khóa và từ khóa chính yếu.

Trong nghiên cứu của chúng tôi, bằng cách kết hợp mô hình dựa trên đồ thị và văn bản, sử dụng gán thể trình tự và phân loại, chúng tôi cố gắng phát triển một cách tiếp cận hiệu quả trích xuất từ khóa có thể loại bỏ nhược điểm của các nghiên cứu trước.

1.3. Các vấn đề cần giải quyết

Các vấn đề chính của đề tài này bao gồm:

- Cách thức để rút gọn và trích xuất các từ khóa chính từ văn bản một cách hiệu quả và chính xác. Đây là bài toán mà các nhà nghiên cứu đang phải đối mặt.
- Các phương pháp trích xuất từ khóa hiện tại chỉ sử dụng thông tin ngôn ngữ (nhóm textual) hoặc cấu trúc đồ thị (nhóm graph-based), chưa kết hợp hai thông tin này một cách phù hợp.
- Đề xuất phương pháp BERTGraph kết hợp học tập đại diện văn bản bằng Transformer và học tập đại diện cấu trúc bằng kỹ thuật nhúng đồ thị.
- Xây dựng mô hình kết hợp các đại diện về ngôn ngữ và cấu trúc thông tin hiệu quả (hybrid models).
- Đánh giá hiệu quả của BERTGraph so với các phương pháp dựa trên một nhóm duy nhất.

Giải quyết các vấn đề này sẽ giúp cải thiện đáng kể chất lượng và hiệu quả của các hệ thống trích xuất từ khóa và dự đoán xu hướng bài báo khoa học, đáp ứng tốt hơn nhu cầu của người dùng và góp phần thúc đẩy sự phát triển của công nghệ xử lý vấn đề ngôn ngữ tự nhiên trong tương lai.

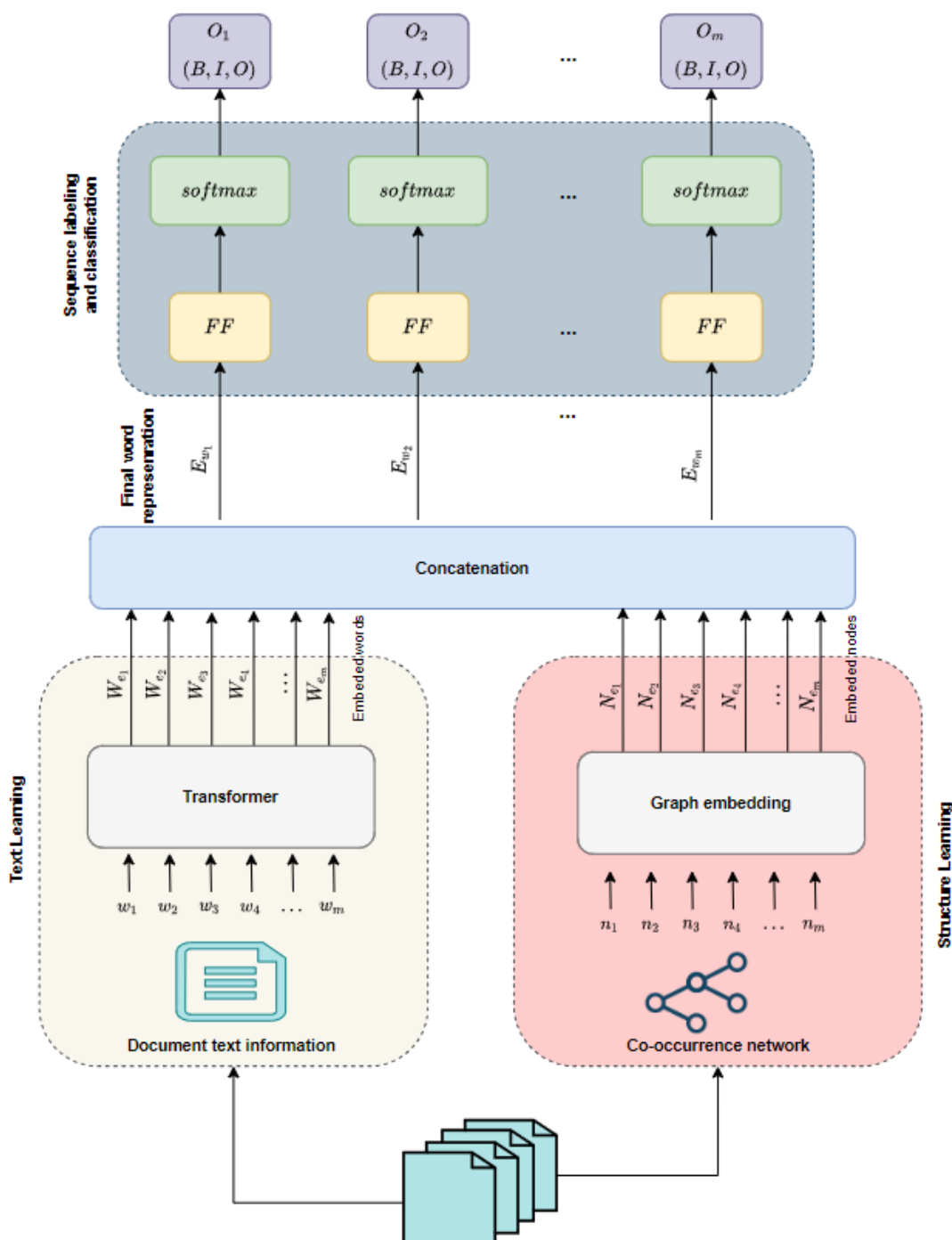
1.4. Đề xuất giải pháp tổng quan

Giải pháp tổng quan của bài báo được đề xuất như sau:

- Đề xuất một phương pháp đa phương thức gọi là BERTGraph kết hợp học tập ngữ cảnh từ văn bản bằng Transformer và học tập đại diện ngữ cảnh từ đồ thị tương tác bằng kỹ thuật nhúng đồ thị.
- Kết hợp hai đại diện về ngôn ngữ và cấu trúc thông tin thành một vectơ đại diện duy nhất cho mỗi từ.
- Giải quyết bài toán trích xuất từ khóa như một nhiệm vụ phân đoạn phụ đề, sử dụng encoder BIO.
- Sử dụng mô hình phân loại để dự đoán nhãn BIO cho mỗi từ trong văn bản.
- Đánh giá hiệu quả của BERTGraph trên kết hợp 3 tập dữ liệu và so sánh với các phương pháp dựa trên ngôn ngữ hoặc đồ thị duy nhất.
- Kết quả cho thấy BERTGraph vượt trội hơn đáng kể so với các phương pháp dựa trên một phương thức.
- Như vậy, giải pháp tổng quan là kết hợp học tập ngữ cảnh đa phương thức, sử dụng phân đoạn và phân loại để giải quyết bài toán trích xuất từ khóa.

Chương 2. Giải pháp trích xuất từ khoá

Phần này trình bày một phương pháp đa phương thức gọi là *BERTGraph* kết hợp học tập ngữ cảnh từ văn bản bằng *Transformer* và học tập đại diện ngữ cảnh từ đồ thị tương tác bằng kỹ thuật nhúng đồ thị.



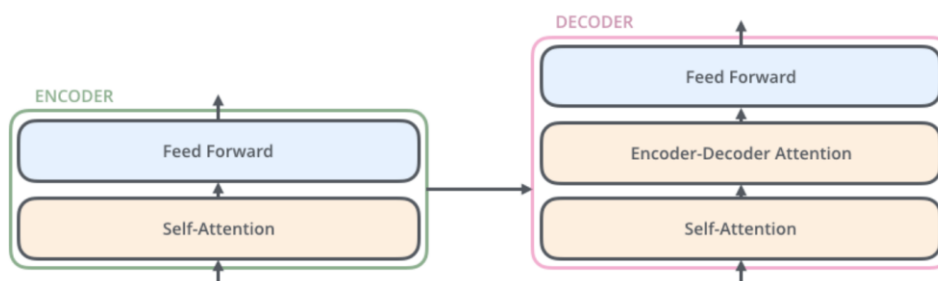
Hình 2.1 Kiến trúc tổng quan của mô hình BERTGraph

Dòng khung làm việc bao gồm bốn bước chính như sau:

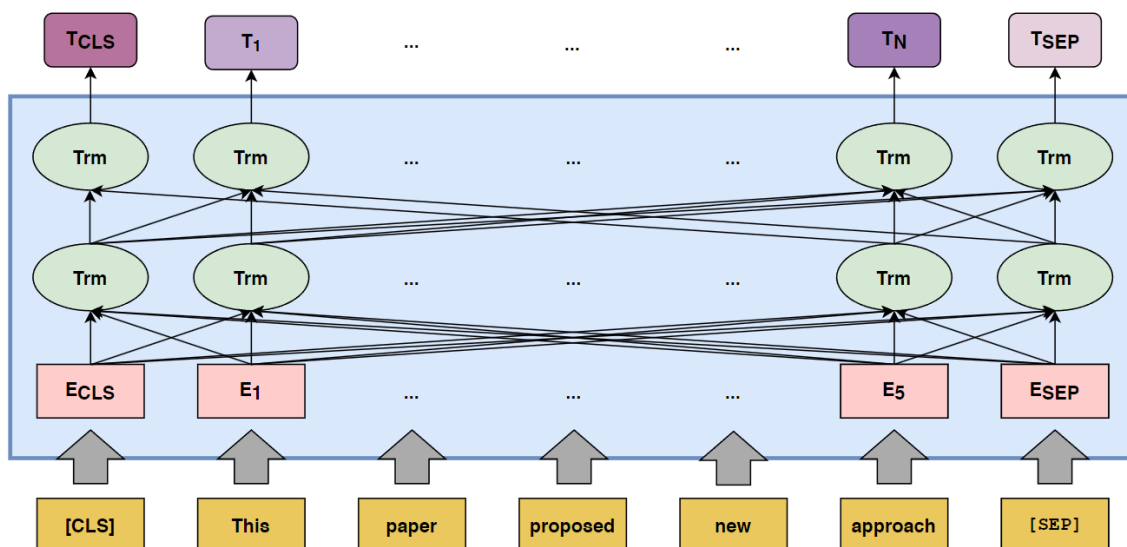
- **Học thông tin văn bản:** Tìm đại diện văn bản cho tất cả các từ bằng mạng biến đổi BERT. Phần này của BERTGraph cung cấp sự hiểu biết sâu hơn để đánh giá tính tương đồng ngữ nghĩa giữa các từ.
- **Học cấu trúc:** Tạo ngữ cảnh cho mỗi từ từ đồ thị xuất hiện bằng kỹ thuật nhúng đồ thị để học đại diện cấu trúc.
- **Đại diện từ cuối cùng:** Nối đại diện văn bản và cấu trúc thành một đại diện duy nhất cho mỗi từ.
- **Gắn thẻ trình tự và phân loại:** Xây dựng trích xuất chuỗi từ khóa dưới dạng nhiệm vụ gắn thẻ trình tự và gắn nhãn cho mỗi từ dựa trên phương pháp BIO qua một lớp hoàn toàn kết nối để phân loại từ.

2.1. Học thông tin văn bản (Text learning):

Bước đầu tiên của BERTGraph là tạo ra vector văn bản cho mỗi từ. Với sự ra đời của Transformer, cách làm việc với dữ liệu văn bản thực sự đã thay đổi. Transformer loại bỏ nhược điểm của kiến trúc RNN và CNN. Áp dụng cơ chế tự chú ý cho phép Transformer song song hóa nhiều hơn các kiến trúc khác [37]. Một Transformer bao gồm các thành phần encoder và decoder như thể hiện tại Hình 2.2. Một thành phần encoder bao gồm một số khối encoder có hai lớp: một lớp Multi-Head Attention và một lớp Feed Forward. Mặt khác, các khối có một lớp Multi-Head Attention bị che trước lớp truyền thông tính tạo thành thành phần decoder [39]. Ngoài ra, cả hai thành phần chứa cùng số lượng khối. Có các mô hình dựa trên cấu trúc Transformer như BERT [40], OpenGPT [41,42], XLNet [43] và ELMo [44]. Trong nghiên cứu này, kỹ thuật học đại diện văn bản là Transformer BERT, cấu trúc được trình bày ở Hình 2.3. Một trong những lợi thế quan trọng của BERT so với các mô hình như Word2Vec là BERT tạo ra đại diện vector từ dựa trên mỗi câu hoặc mỗi tài liệu chứa từ đó. Điều này có nghĩa là BERT có khả năng nắm bắt ngữ cảnh của từ trong tài liệu. Thông tin văn bản của tài liệu bao gồm tiêu đề và tóm tắt của tài liệu. Khối bên trái tại Hình 2.1 cho thấy quá trình học vector từ dựa trên văn bản.



Hình 2.2 The Transformer model architecture



Hình 2.3 The BERT Transformer architecture

Ở bước này, chúng tôi sử dụng một pretrain model của BERT được đánh giá cao trong các nhiệm vụ xử lý ngôn ngữ tự nhiên trên tập dữ liệu tiếng anh là google-bert/bert-base-uncased. Các thử nghiệm xây dựng kiến trúc cũng như huấn luyện trên tập dữ liệu khác không mang lại kết quả tương tự như pretrained model đã giới thiệu. Để đảm bảo khả năng hiểu ngữ nghĩa của từ (phần quan trọng trong kiến trúc) thì kết quả thực hiện xây dựng và huấn luyện của BERT tự xây dựng sẽ bỏ qua và sử dụng các kết quả của mô hình BERT pretrained.

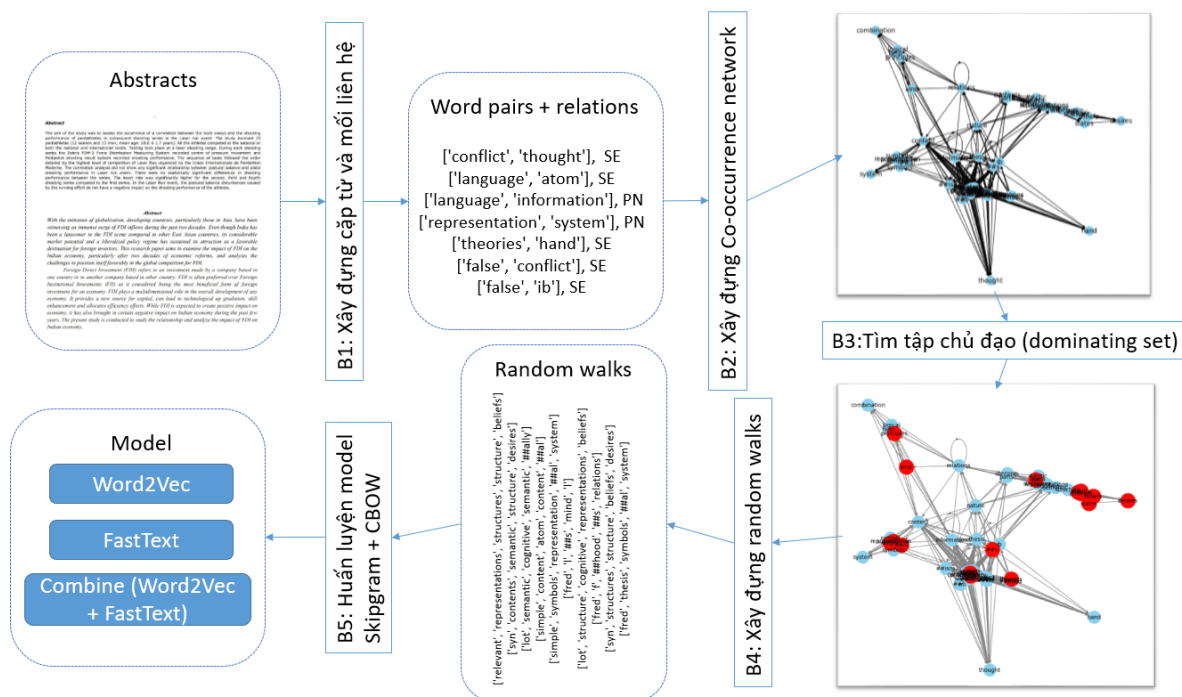
2.2. Học cấu trúc (Structure learning)

Bước thứ hai là học vector cấu trúc cho mỗi từ dựa trên thuật toán ExEm mà chúng tôi tự xây dựng lại.

Nó được giới thiệu nằm trong ba thuật toán nhúng đồ thị bao gồm ExEm [47], Node2vec [48] và DeepWalk [49] trong mạng xuất hiện. Một mạng xuất hiện được định nghĩa là đồ thị $G = (V, E)$ trong đó các nút biểu diễn các từ và mỗi cạnh $e \in E$ cho thấy mối quan hệ xuất hiện giữa từ vi và từ vj xuất hiện trong ngữ cảnh của số. Bởi vì thông tin có thể thu được từ một tài liệu đơn lẻ là hữu hạn, chúng tôi xây dựng đồ thị xuất hiện trên tất cả các tài liệu thay vì một tài liệu đơn lẻ, như khối bên phải ở Hình 2.1. Lưu ý rằng trước khi xây dựng đồ thị này, chúng tôi loại bỏ các stopwords và dấu câu. Sau đó, nhiệm vụ của chúng tôi là học các biểu diễn ẩn chiều của các từ từ đồ thị này bằng kỹ thuật nhúng đồ thị. Có các thuật toán học đại diện nút khác nhau. Theo kết quả được báo cáo bởi các nghiên cứu khác nhau, ba phương pháp nhúng đồ thị dựa trên random walks có hiệu suất tốt hơn. Trong DeepWalk, một tập các random walks được tạo bởi việc bắt đầu từ mỗi nút trong đồ thị. Trong khi đó, Node2vec đề xuất phương pháp random walks đã được chỉnh sửa từ DeepWalk. Phương pháp này sử dụng hai tham số để kiểm soát

không gian tìm kiếm. Ngoài ra, ExEm là một kỹ thuật sử dụng lý thuyết tập lớn thống trị để tạo random walks. ExEm mô tả các lân cận cục bộ bằng cách bắt đầu mỗi đường thăm dò với một nút thống trị. Cấu trúc toàn cục cũng được bắt rơi bằng cách chọn một nút thống trị khác trong random walks. Cả ba phương pháp đều nhúng các random walks này vào mô hình mạng thần kinh dạng vực để học các đại diện nút.

Trong bài báo này chúng tôi tiếp cận theo hướng ExEm với quá trình thực thi khác biệt và giải pháp đề xuất khác với ExEm được giới thiệu trong bài báo. Cụ thể kiến trúc ExEm được mô tả qua hình sau:



Hình 2.4 Sơ đồ khối tổng quan mô quá trình học cấu trúc

2.2.1. Xây dựng cặp từ và mối liên hệ

Từ đầu vào là các đoạn abstract từ tất cả các bài báo trong tập dữ liệu, chúng tôi thực hiện tiền xử lý bao gồm loại bỏ các từ stopword, viết thường. Vì để thống nhất với các token được embedding bên BERT ở mục 2.2, chúng tôi thực hiện phân tách token của các câu, đoạn trong abstract sử dụng tokenizer của model BERT. Nhận thấy các token không phải là cụm danh từ thì sẽ không được xét đến trong keyword nên chúng tôi chỉ giữ lại những token mà nằm trong cụm danh từ thường được làm chủ ngữ, tân ngữ trong câu.

Một ví dụ đơn giản chứng minh cho khẳng định trên, chúng tôi chọn ngẫu nhiên các keyword trong toàn bộ keyword thu được kết quả sau: “bluetooth; bit error rate; modulation; white noise; telepresence; animation; avatars; self-organizing map; sigma delta modulators; social brain”. Có thể dễ dàng thấy được 10/10 keywords được chọn

ngẫu nhiên đề là danh từ, cụm danh từ. Tất nhiên, chúng tôi sẽ kiểm tra trên bộ keyword nhiều hơn nên đảm bảo được tính đúng đắn của khẳng định này.

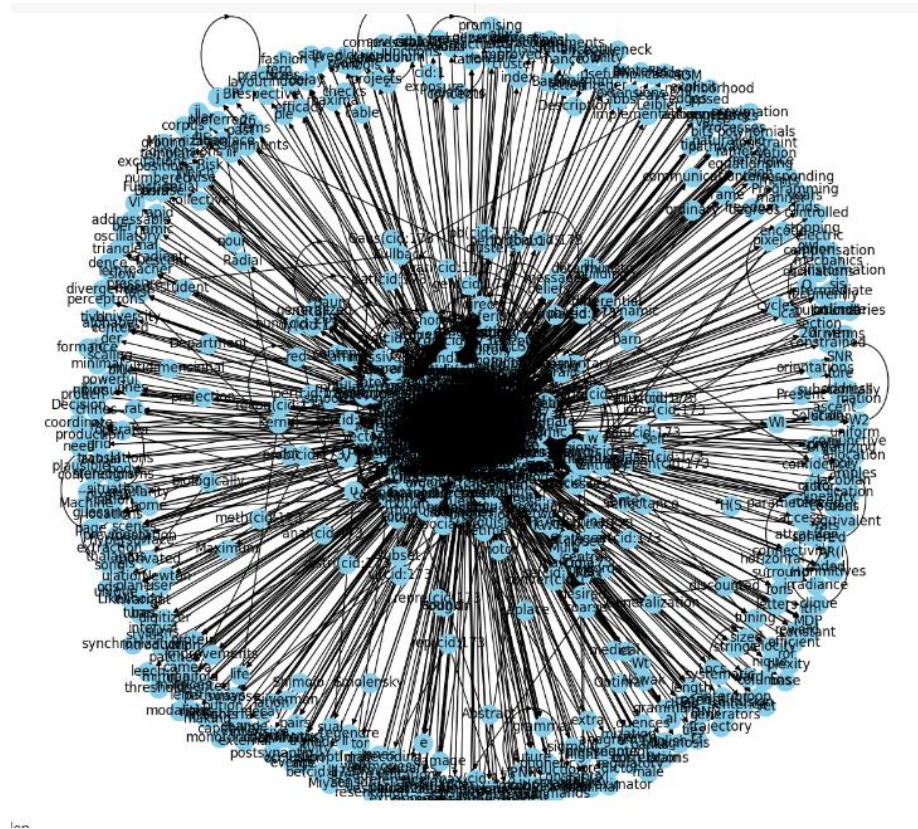
Chúng tôi xây dựng 2 quan hệ được xem là liên quan trực tiếp đến vấn đề trích xuất keyword và hiểu ngữ cảnh của từ trong câu. Một là mối quan hệ của các cụm danh từ (PN), hai là mối quan hệ giữa các cụm danh từ trong câu (SE). Đối với quan hệ đầu tiên, giữa các cụm danh từ sẽ có các danh từ bổ nghĩa cho danh từ, tính từ bổ nghĩa cho danh từ, trạng từ bổ nghĩa cho tính từ, nhiều tính từ bổ nghĩa cho danh từ,... tất cả các mối liên hệ trong một cụm danh từ đó chúng tôi xét chung vào một loại quan hệ là PN. Tiếp đến quan hệ còn lại, trong một câu sẽ có nhiều cụm danh từ do đó để biểu diễn sự liên kết nằm trong cùng một câu có sự liên quan với nhau nhưng ở mức độ nhẹ hơn chúng tôi đặt quan hệ đó là SE.

Ví dụ: “Việt Nam là một quốc gia xinh đẹp”. Ở đây các mối liên hệ có thể xác định là [‘Việt’, ‘Nam’, ‘PN’], [‘quốc’, ‘gia’, ‘PN’], [‘Việt’, ‘quốc’, ‘SE’],... Với kết quả trên, có thể thấy phần nào quan hệ đã được mô tả, tuy có rời rạc nhưng khi được đưa vào một mạng tổng thể thì chúng sẽ được kết nối với nhau chặt chẽ.

2.2.2. Xây dựng Co-occurrence network

Từ các cặp từ quan hệ đã được xây dựng ở bước trước, chúng tôi thực hiện tạo một network để lưu trữ một cách trực quan các mối quan hệ giữa các từ với nhau. Theo đó mỗi nút của đồ thị là một token trong tập dữ liệu, mỗi cạnh sẽ là các nút liên hệ đến các nút khác, với 2 quan hệ mà chúng tôi đã mô tả. Chúng tôi sẽ lưu trữ các mối quan hệ không hướng, tức là các từ source liên kết với các từ target chỉ thông qua một mối quan hệ duy nhất, không có đường ngược lại.

Để có thể lưu trữ và thực hiện các bước tính toán một cách dễ dàng chúng tôi tổ chức thông qua một thư viện là NetworkX, đối tượng được khởi tạo với source, target, edge. Ví dụ: [‘Việt’, ‘Nam’, ‘PN’] tương ứng chính là 3 giá trị cần truyền cho đối tượng NetworkX.



Hình 2.5 Hình vẽ đồ thị graph cho 1/12 tập dữ liệu huấn luyện

2.2.3. Tìm tập chủ đạo (dominating set)

➤ Dựa trên mã giả sau:

Algorithm 1 Finding a dominating set

Require: Đồ thị có kết nối $G = (V, E)$

$D = \emptyset$

loop

if IsEmpty($V - [D \cup \text{Neighbors}(D)]$) **then**

STOP

end if

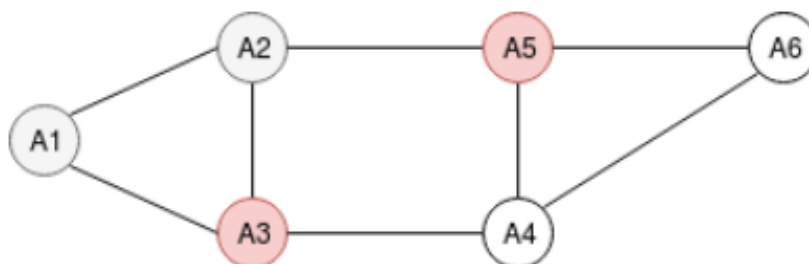
Chọn ngẫu nhiên một vector $w \in V - [D \cup \text{Neighbors}(D)]$

$D \leftarrow D \cup \{w\}$

end loop

return D

Theo đó, chúng tôi tạo một bản sao của graph, thực hiện tìm tập chủ đạo trên tập này và giữ nguyên tập gốc. Sau khi thực hiện các bước như thuật toán ở trên sẽ nhận được một tập thống trị (hay tập chủ đạo) phân bố đồng đều trên toàn bộ đồ thị. Một số nhận xét có thể rút ra ở đây là, tất cả các điểm thống trị không thể nối trực tiếp với nhau mà không thông qua một nút trung gian.



Hình 2.6 Một ví dụ của một đồ thị có 6 nodes, 8 edges và tập chủ đạo của nó.

2.2.4. Xây dựng random walks

Với việc có các nút thống trị từ bước trước, chúng tôi giới thiệu chiến lược random walks thông minh của mình trong tiểu mục này. Trước khi cung cấp đầy đủ chi tiết về các random walks được đề xuất, chúng tôi sẽ mô tả random walks là gì và tại sao nó lại quan trọng trong việc nhúng biểu đồ. Random walks trên biểu đồ được định nghĩa là một chuỗi ngẫu nhiên các nút trong đó các nút liên tiếp là lân cận (Liu và cộng sự, 2016). Random walks có thể lấy được thông tin ẩn trong cấu trúc biểu đồ. Tầm quan trọng của các random walks trong miền nhúng biểu đồ được áp dụng từ xử lý ngôn ngữ tự nhiên (NLP) sau thành công lớn của các mô hình nhúng từ. Trong việc nhúng biểu đồ, các thuộc tính của biểu đồ được bảo toàn bằng một tập hợp các đường đi ngẫu nhiên được lấy mẫu từ nó (Cai và cộng sự, 2018). Nói cách khác, mỗi random walks trong việc nhúng biểu đồ sẽ trình bày một khái niệm khác tương đương với định nghĩa câu trong miền NLP. Điều đó có nghĩa là random walks và câu có cùng trách nhiệm trong phạm vi của chúng. Ngoài ra, các nút của random walks đảm nhận vai trò của từ hoặc từ vựng trong câu. Có một số ưu điểm của phương pháp nhúng biểu đồ dựa trên random walks bao gồm mức độ phức tạp về thời gian và không gian có thể chấp nhận được (Pimentel và cộng sự, 2019), không cần kỹ thuật tính năng và điều tra các phần khác nhau của cùng một biểu đồ cùng một lúc bởi một số lượng đường dẫn được lấy mẫu (Grover và Leskovec, 2016; Cai và cộng sự, 2018; Liu và cộng sự, 2016). Do đó, nhiều phương pháp nhúng biểu đồ đã được đề xuất dựa trên các random walks như DeepWalk và Node2vec trong đó sự khác biệt của chúng đến từ chiến lược lấy mẫu. Tuy nhiên, những phương pháp này gặp khó khăn trong việc tìm kiếm quy trình lấy mẫu tối ưu. DeepWalk sử dụng các random walks thống nhất và không thể kiểm soát không gian tìm kiếm.

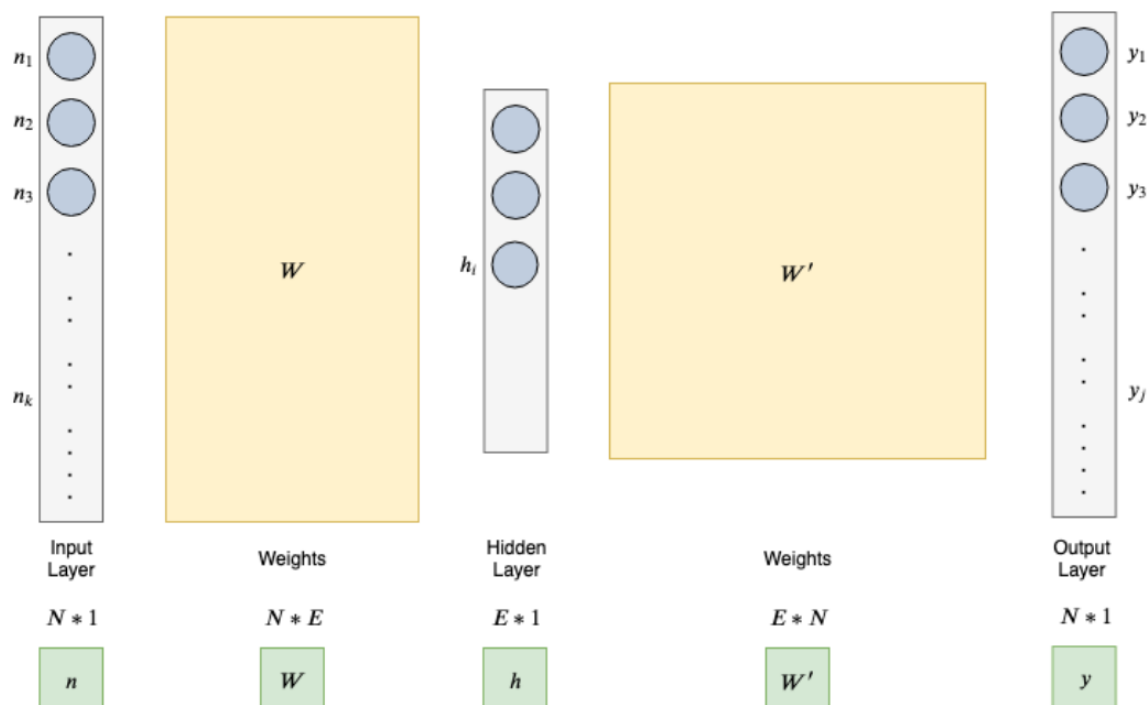
Node2vec gợi ý một random walks có sai lệch trong đó một số nút lân cận có xác suất được chọn cao hơn hoặc thấp hơn trong mỗi bước bởi hai tham số. Vấn đề là tìm ra các giá trị tốt nhất cho các tham số này để xác định khả năng quan sát các nút trong mỗi random walks cho mọi mạng.

ExEm là một kỹ thuật dựa trên random walks nhằm sửa đổi chiến lược random walks được sử dụng trong DeepWalk và Node2vec bằng cách thuê các nút thống trị. Các random walks thông minh được đề xuất của chúng tôi mang đến sự linh hoạt trong các nút lấy mẫu từ một mạng. Khái niệm về thông tin này xuất phát từ việc xuất hiện hai nút thống trị trong các đường dẫn được lấy mẫu. Đối với mỗi random walks, ExEm bắt đầu đường dẫn bằng cách chọn ngẫu nhiên một nút từ tập hợp ưu thế được tìm thấy ở bước trước. Sau đó, một trong những hàng xóm của nút thống trị này được chọn ngẫu nhiên và được thêm vào cuộc đi bộ. Sau đó, walk sẽ di chuyển đến các nút lân cận của nút được thêm cuối cùng. Quy trình thêm các nút mới vào bước đi tiếp tục cho đến khi đáp ứng được hai điều kiện sau. Điều kiện chính là sự xuất hiện của ít nhất một nút chiếm ưu thế khác trong đường dẫn được lấy mẫu. Yêu cầu khác để kết thúc quá trình là đạt được độ dài cố định LR. Khối con thứ hai trong khối thứ hai của Hình 1 hiển thị các ví dụ về đường dẫn ngẫu nhiên được tạo bởi ExEm từ biểu đồ Scopus. Trong trường hợp này, mỗi nút được trình bày bởi một id chuyên gia, các nút màu đỏ biểu thị các nút thống trị và độ dài của bước đi bằng 5. Rõ ràng, mỗi bước đi bắt đầu bằng một nút thống trị và nút thống trị thứ hai có thể được truy cập ở các vị trí khác nhau của bước đi ngoại trừ vị trí thứ hai. Giải thích là dựa trên Thuật toán 1, trong quá trình tìm các nút trội, chúng ta loại bỏ một nút và các nút lân cận của nó sau khi thêm nút này vào tập thống trị. Vì vậy, không có bất kỳ nút thống trị nào trong một bước nhảy của nút này và các nút thống trị khác. Lưu ý rằng có thể thấy nhiều hơn hai nút chiếm ưu thế trong mỗi đường dẫn giống như random walks thứ ba trong hình.

```
[ 'relevant', 'representations', 'structures', 'structure', 'beliefs' ]
[ 'syn', 'contents', 'semantic', 'structure', 'desires' ]
[ 'lot', 'semantic', 'cognitive', 'semantic', '##ally' ]
[ 'simple', 'content', 'atom', 'content', '##al' ]
[ 'simple', 'symbols', 'representation', '##al', 'system' ]
[ 'fred', 'l', '##s', 'mind', 'l' ]
[ 'lot', 'structure', 'cognitive', 'representations', 'beliefs' ]
[ 'fred', 'f', '##hood', '##s', 'relations' ]
[ 'syn', 'structures', 'structure', 'beliefs', 'desires' ]
[ 'fred', 'thesis', 'symbols', '##al', 'system' ]
```

Hình 2.7 Ví dụ cho random walks

2.2.5. Huấn luyện model Skipgram + CBOW:



Hình 2.8 Kiến trúc Skipgram + CBOW

Đầu vào bắt buộc duy nhất của bước này là một kho dữ liệu được tạo từ các random walks thông minh của bước trước đó. Như đã đề cập trước đó, trong các mô hình random walks, nút và random walks lần lượt được coi là một từ và một câu. Do đó, vùng lân cận của một nút có thể được coi là sự xuất hiện đồng thời của các từ trong câu. Hơn nữa, có nhiều cách tiếp cận dựa trên deep learning có thể ánh xạ các lần xuất hiện của từ vào mô hình không gian vector. Một trong những kỹ thuật đơn giản và hiệu quả nhất là mô hình Skip-gram (Mikolov và cộng sự, 2013b). Mục đích của Skip-gram là dự đoán các từ xung quanh từ mục tiêu. Nỗ lực tương tự có thể được thực hiện trong việc nhúng biểu đồ. Theo đó, khi nhúng biểu đồ, Skip-gram đếm số lần nút j xuất hiện trong một cửa sổ nhất định của w . Ví dụ: trong các random walks “ $n_1 n_2 n_3 n_4 n_5$ ”, chương trình bỏ qua lấy nút “ n_3 ” làm đầu vào và dự đoán đầu ra “ n_1 ”, “ n_2 ”, “ n_4 ” và “ n_5 ”, giả sử w là 5. Kiến trúc Skip-gram là mạng chuyển tiếp nguồn cấp dữ liệu, là mô hình học sâu đơn giản nhất để biểu diễn nút. Như được hiển thị trong Hình 3, mô hình này xem biểu đồ dưới dạng một túi các nút. Đối với nút n_i , nó thu thập vector E chiều y_i bằng cách sử dụng mô hình nhúng như Word2vec (Mikolov và cộng sự, 2013a) hoặc FastText (Joulin và cộng sự, 2016). Word2vec học cách chuyển đổi các nút xuất hiện trong các bước ngẫu nhiên tương tự thành các biểu diễn vector tương tự. Trong khi đó, FastText tận dụng lợi thế của một túi n -gram làm tính năng bổ sung để lấy thông tin thứ tự nút cục bộ.

Sentence	Word pairs
the pink horse is eating	(the , pink), (the , horse)
the pink horse is eating	(pink , the), (pink , horse), (pink , is)
the pink horse is eating	(horse , the), (horse , pink), (horse , is), (horse , eating)
the pink horse is eating	(is , pink), (is , horse), (is , eating)
the pink horse is eating	(eating , horse), (eating , is)

Hình 2.9 Ví dụ cho các cặp đầu vào và đầu ra của mô hình Skipgram + CBOW

Xem xét các giải thích ở trên, trong bước này của ExEm, các random walks từ bước trước đó được đưa dưới dạng kho dữ liệu vào đầu vào của mạng Skip-gram. ExEm khai thác ba phương pháp nhúng bao gồm Word2vec, FastText và ghép nối hai phương pháp này để trích xuất các phần nhúng, như được trình bày trong khối con thứ ba trong khối thứ hai của Hình 1. Có hai điểm quan trọng trong bước này cần lưu ý. Đầu tiên là có ít nhất ba cách phổ biến để kết hợp các vector nhúng và tạo ra một vector duy nhất bao gồm: tính tổng, lấy trung bình và ghép nối (Damoulas và Girolami, 2009). Trong nghiên cứu này, chúng tôi coi việc ghép hai phần nhúng là phương pháp kết hợp cơ bản và để nghiên cứu sâu hơn, chúng tôi kiểm tra tính tổng và tính trung bình của các phần nhúng Word2vec và FastText trong kết quả đánh giá. Chủ đề thứ hai là trong mô hình Skip-gram, cửa sổ ngữ cảnh có ảnh hưởng quan trọng đến các biểu diễn vector thu được. Cửa sổ ngữ cảnh xác định những hàng xóm nào sẽ được lưu ý khi tính toán các biểu diễn vector (Lison và Kutuzov, 2017). Do đó, việc có ít nhất hai nút thống trị trong cửa sổ ngữ cảnh sẽ đảm bảo rằng ExEm hiểu đúng thông tin biểu đồ cục bộ và tổng thể cũng như tôn trọng các mục tiêu vai trò cấu trúc và đồng tính luyện ái. Do quy trình này dựa trên cách lấy mẫu mỗi nút n_j với xác suất phụ thuộc vào khoảng cách $|j - i|$ đến nút tiêu điểm 10 ni, như được chứng minh bởi (Lison và Kutuzov, 2017):

$$p(n_i | n_j) = \sum_{\omega=1}^W p(n_i | n_j, \omega) p(\omega) = \frac{1}{\omega} (\omega - |j - i| + 1) \quad (1)$$

trong đó w là kích thước cửa sổ thực từ 1 đến W . Ví dụ: với kích thước cửa sổ 5, nút thống trị thứ hai ở vị trí 3 sẽ được lấy mẫu với xác suất là $\frac{3}{5}$ trong Word2vec (Lison và Kutuzov, 2017). Nói cách khác, mô hình Skip-gram tối đa hóa xác suất xảy ra đồng thời giữa các nút thống trị tồn tại trong một cửa sổ w (Cai và cộng sự, 2018)

2.3. Đại diện từ cuối cùng (Final word representation)

Sau khi nhận được đại diện vector của thông tin văn bản và cấu trúc mạng xuất hiện đối với mỗi từ, bước tiếp theo là kết hợp những thông tin này thành một đại diện duy nhất. Chúng tôi tin rằng việc nối vector dựa trên văn bản và cấu trúc có thể phát hiện tốt hơn tiềm năng của các từ cho việc trở thành từ khóa. Do đó, chúng tôi trình bày mỗi từ dưới dạng một vector đơn biệt kết hợp từ vector văn bản và cấu trúc. Ví dụ đối với từ w_i , chúng tôi chỉ ra rằng $E_{w_i} = W_{e_i} + N_{e_i}$ trong đó W_{e_i} và N_{e_i} lần lượt biểu

thị đại diện học từ thông tin văn bản và cấu trúc đối với từ w_i .

2.4. Gắn thẻ trình tự và phân loại (Sequence labeling and classification)

Phân loại trình tự là một loại nhiệm vụ nhận dạng mẫu trong lĩnh vực xử lý ngôn ngữ tự nhiên, phân loại các từ trong văn bản và gán một lớp hoặc nhãn cho mỗi từ trong một trình tự đầu vào cho trước [50,51]. Có nhiều kỹ thuật cho nhiệm vụ phân loại trình tự bao gồm mô hình Markov ẩn [52], Trường ngẫu nhiên có điều kiện (CRF) [53] và các phương pháp học sâu.

Trong bài báo này, chúng tôi xem xét bài toán trích xuất chuỗi từ khóa từ tài liệu như một nhiệm vụ phân loại trình tự. Cũng chúng tôi quan sát việc phân loại trình tự dưới dạng một nhiệm vụ phân loại sử dụng phương pháp encoder BIO như là nhãn ra. Do đó, mô hình của chúng tôi lấy Ew_1, Ew_2, \dots, Ew_m là đầu vào và gán mỗi từ một nhãn $O_i \in \{B, I, O\}$ trong đó B cho thấy w_i là bắt đầu của một chuỗi từ khóa, I chỉ ra rằng w_i nằm trong một chuỗi từ khóa, và O minh họa rằng w_i nằm ngoài chuỗi từ khóa. Có thể thấy từ Hình 1 rằng một cấu trúc hoàn toàn kết nối và lớp softmax được sử dụng để ra quyết định phân loại.

2.5. Tham số của mô hình

Trong kiến trúc của model chúng tôi sử dụng bộ tham số sau:

- *out_d_bert*: 768 là số chiều của vector đầu ra đối với mô hình BERT;
- *number_of_walks*: $1e4$ hoặc $1e5$ số random walks được thực hiện;
- *threshold*: 2 hoặc 3 là ngưỡng để xác định loại bỏ hay không loại bỏ từ đó khỏi đồ thị graph;
- *len_random_walks*: 5 là độ dài tối đa của mỗi random walks;
- *vector_size*: 100 hoặc 200 hoặc 300 là số lượng vector embedding trong mô hình Skipgram và CBOW;
- *max_length*: 512 là số lượng giới hạn token đầu vào của một abstract;
- *hidden_size* = 256 là kích thước lớp ẩn sử dụng trong kiến trúc feed forward;
- *num_classes* = 3 là số lượng lớp đầu ra ứng với ba nhãn B, I, O.

Chương 3. Phân tích và thiết kế hệ thống

Phần này trình bày kiến trúc của hệ thống, mô tả các module, phân tích yêu cầu và các chức năng của hệ thống. Đồng thời trình bày cơ sở dữ liệu và tổng quan quá trình xây dựng hệ thống.

3.1. Phân tích thiết kế hệ thống

3.1.1. Kiến trúc hệ thống

Bảng 3-1 Kiến trúc hệ thống

Module	Thành phần	Công nghệ sử dụng
Backend Web	API, xử lý dữ liệu	Laravel
Backend AI	API, model	Django
Frontend	Giao diện người dùng, xử lý sự kiện	Vue.js
Cơ sở dữ liệu	MySQL (users, trackings, passwords)	MySQL

Luồng Hoạt Động

1. Dữ liệu về bài báo khoa học được tự động crawl từ trang web <https://proceedings.neurips.cc> và được lưu vào file data_paper.csv sau đó cho qua Model đã được huấn luyện từ trước để trích xuất ra tập keywords của mỗi bài báo khoa học.
2. Người dùng truy cập vào trang web để sử dụng các chức năng cũng như tương tác với hệ thống.
3. Vuejs gửi các dữ liệu tracking được từ người dùng trên trang web cũng như gửi các yêu cầu đến cho server Laravel, server Django xử lý và lưu lại.
4. Server Laravel xử lý các yêu cầu về mặt chức năng chung của hệ thống cho Client Vuejs.
5. Server Django xử lý các yêu cầu về liên quan đến Model cho Client Vuejs.
6. Dữ liệu của người dùng sẽ được lưu trữ ở trên cơ sở dữ liệu MySQL.
7. Dữ liệu tracking được từ người dùng sẽ được lưu lại ở file tracking_data.csv để sử dụng cho việc huấn luyện lại Model.

3.1.2. Mô tả các module trong hệ thống

1. Module Backend Web:

- Mô tả: Xử lý yêu cầu API và quản lý dữ liệu MySQL.
- Chi tiết:

- API: Laravel được sử dụng để tạo các endpoint API.
- Xử lý dữ liệu: Laravel xử lý và xác thực dữ liệu trước khi gửi trả kết quả.

2. *Module Backend AI:*

- Mô tả: Xử lý yêu cầu API liên quan đến model và quản lý dữ liệu file CSV.
- Chi tiết:
 - API: Django được sử dụng để tạo các endpoint API.
 - Xử lý dự đoán: Django xử lý gọi model dự đoán, định dạng kết quả trả về phản hồi qua API.

3. *Module Frontend:*

- Mô tả: Cung cấp giao diện người dùng và xử lý sự kiện người dùng.
- Chi tiết:
 - Giao diện người dùng: Vue.js được sử dụng để xây dựng các component giao diện.
 - Xử lý sự kiện: Vue.js xử lý các sự kiện người dùng và cập nhật giao diện tương ứng.

4. *Module Cơ sở dữ liệu:*

- Mô tả: Lưu trữ và quản lý thông tin người dùng, các thông tin tracking và mật khẩu.
- Chi tiết:
 - MySQL: Hệ quản trị cơ sở dữ liệu để lưu trữ thông tin người dùng, các thông tin tracking và mật khẩu.
 - Dữ liệu được lưu trữ thông qua ID người dùng nên sẽ dễ dàng truy cập và quản lý.

3.2. Phân tích yêu cầu

Các yêu cầu của trang web là các tính năng sẽ giúp người dùng có thể dễ dàng thực hiện mục đích của mình khi đến với ứng dụng bao gồm:

- Trích xuất keyword từ một bài báo khoa học nói chung và một đoạn văn bản nói riêng.
- Cung cấp các sơ đồ, biểu đồ thể hiện các keywords (chủ đề) đang thịnh hành một cách trực quan.
- Cho phép nghiên cứu sinh tìm kiếm thông qua mô tả về công việc muốn thực hiện, hiển thị danh sách các bài báo khoa học liên quan nhất, có kèm link bài báo.
- Gợi ý các chủ đề đang thịnh hành và bài báo khoa học về chủ đề đó.
- Chọn keyword mà người dùng mong muốn thống kê, hệ thống sẽ thống kê xu hướng của keyword đó.
- Tìm kiếm các keyword có liên quan về mặt chủ đề, trực tiếp đi kèm với nhau

trong các bài báo.

3.3. Phân tích chức năng

3.3.1. Đối tượng sử dụng

Người dùng:

- Hệ thống cho phép người dùng truy cập vào trang chính với quyền người dùng, thực hiện đăng nhập để lưu trữ thông tin cá nhân và các hành động thực hiện với trang web.
- Người dùng là các nghiên cứu sinh muốn tìm hiểu các chủ đề đang là xu hướng hiện nay, hoặc bất cứ chủ đề nào người dùng quan tâm và muốn thống kê xu hướng của chủ đề đó.
- Hệ thống còn hướng đến những người dùng muốn tìm kiếm một số bài báo liên quan đến chủ đề mà mình muốn tìm hiểu.

Admin:

- Thực hiện quản lý người dùng, thêm, sửa, xóa thông tin người dùng.

3.3.2. Yêu cầu chức năng

1. **Chức năng đăng nhập:** Người dùng có thể đăng nhập vào hệ thống để thực hiện các chức năng chính của hệ thống, đồng thời lưu lại lịch sử người dùng khác nhau.
2. **Chức năng quên mật khẩu:** Người dùng có thể đặt lại mật khẩu cho tài khoản trong trường hợp bị quên mật khẩu thông qua email đã được xác thực.
3. **Chức năng chỉnh sửa thông tin cá nhân:** Người dùng có thể chỉnh sửa các thông tin cá nhân như: Tên, ngày sinh, giới tính, địa chỉ, ảnh đại diện và số điện thoại.
4. **Chức năng gợi ý bài báo:** Hệ thống lưu lại lịch sử người dùng cho từng người dùng khác nhau và thực hiện gợi ý, đề xuất các bài báo liên quan đến các từ khoá mà người dùng đã tìm kiếm trước đó.
5. **Chức năng tìm kiếm:** Người dùng sẽ nhập thông tin tìm kiếm vào ô tìm kiếm. Hệ thống sẽ thực hiện tìm kiếm dựa trên yêu cầu mà người dùng muốn tìm kiếm bao gồm các phương thức sau:
 - Tìm kiếm dựa trên từ khoá: Người dùng trực tiếp tìm kiếm từ khoá, hệ thống sẽ tìm các bài báo có danh sách từ khoá chứa từ khoá đó.
 - Tìm kiếm dựa trên dữ liệu khác: Người dùng nhập tên tác giả, tên tiêu đề bài báo thì kết quả trả về sẽ là các bài báo có tác giả đó hoặc có tiêu đề chứa tiêu đề đó, ngoài ra có nhiều thông tin khác của bài báo liên quan.
 - Tìm kiếm dựa trên mô tả: Người dùng nhập mô tả vào ô tìm kiếm, hệ thống sẽ thực hiện trích xuất từ khoá từ mô tả đó ra danh sách các từ khoá và người dùng

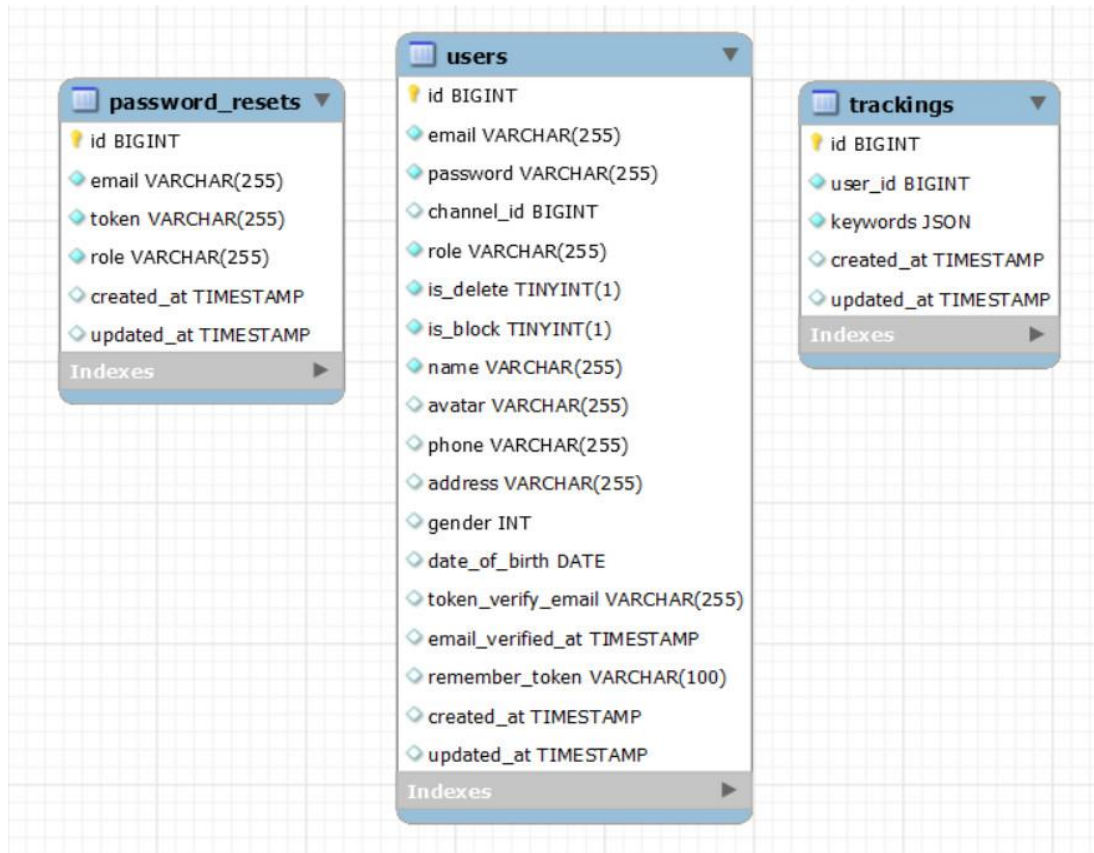
được hệ thống yêu cầu chọn một trong các từ khoá được trích xuất để thực hiện tìm kiếm bài báo.

6. **Chức năng thống kê top 10 từ khoá xu hướng:** Hệ thống sẽ thống kê tất cả các từ khoá xuất hiện trong tất cả các bài báo có sẵn trong hệ thống và hiển thị cho người dùng biểu đồ xu hướng trên trang chủ của hệ thống.
7. **Chức năng xem bảng xếp hạng xu hướng các từ khoá:** Trên trang chủ một bảng thống kê sẵn và sắp xếp giảm dần các từ khoá phổ biến, đồng thời có một wordcloud để cho giao diện sinh động hơn.
8. **Chức năng tracking lịch sử người dùng:** Các hành động trực tiếp của người dùng bao gồm tìm kiếm và chọn vào các từ khoá mà người dùng muốn tìm hiểu trong hệ thống sẽ được lưu lại và làm cơ sở để gợi ý các bài báo cho người dùng.
9. **Chức năng tracking để crawl dữ liệu huấn luyện:** Khi người dùng tìm kiếm một từ khoá hoặc tìm kiếm bằng một mô tả chứa các từ khoá thì hệ thống sẽ hiển thị các bài báo liên quan đến từ khoá, chủ đề mà người dùng đang tìm kiếm. Hành động xem thông tin chi tiết của bài báo được tracking lại và lưu trữ bài báo kèm với từ khoá mà người dùng đang tìm kiếm để thực hiện crawl dữ liệu huấn luyện giúp cải thiện model sau quá trình huấn luyện.
10. **Chức năng quản lý thông tin người dùng:** Admin thực hiện đăng nhập với tài khoản của mình và thực hiện quản lý các thông tin tài khoản của người dùng.

3.4. Thiết kế cơ sở dữ liệu

3.4.1. Lược đồ cơ sở dữ liệu

- Hệ thống sử dụng MySQL làm cơ sở dữ liệu để lưu trữ các thông tin cần thiết nhằm phục vụ các tác vụ truy xuất, tìm kiếm dữ liệu.
- Hình bên dưới là thiết kế cơ sở dữ liệu của hệ thống.



Hình 3.1 Lược đồ cơ sở dữ liệu

3.4.2. Mô tả bảng người dùng “users”

Bảng 3-2 Mô tả bảng người dùng “users”

STT	Thuộc Tính	Kiểu Dữ Liệu	Ý nghĩa
1	id	int	Khóa chính để phân biệt các người dùng
2	email	string	Email của người dùng
3	password	string	Mật khẩu của người dùng
4	name	string	Tên của người dùng
5	phone	string	Số điện thoại của người dùng
6	address	string	Địa chỉ của người dùng
7	avatar	string	Ảnh đại diện của người dùng
8	is_accept	boolean	Chấp thuận người dùng bởi quản trị viên
9	role	string	Quyền của người dùng : admin, manager
10	created_at	timestamp	Thời gian tạo bảng ghi
11	updated_at	timestamp	Thời gian cập nhật bảng ghi

3.4.3. Mô tả bảng tracking “trackings”

Bảng 3-3 Mô tả bảng tracking “trackings”

STT	Thuộc Tính	Kiểu Dữ Liệu	Ý nghĩa
1	id	int	Khóa chính để phân biệt các trackings
2	user_id	string	Tên người dùng
3	keywords	json	Dữ liệu trackings {từ khoá: tần suất}
4	created_at	timestamp	Thời gian tạo bảng ghi
5	updated_at	timestamp	Thời gian cập nhật bảng ghi

3.4.4. Mô tả bảng đặt lại mật khẩu “password_resets”

Bảng 3-4 Mô tả bảng đặt lại mật khẩu “password_resets”

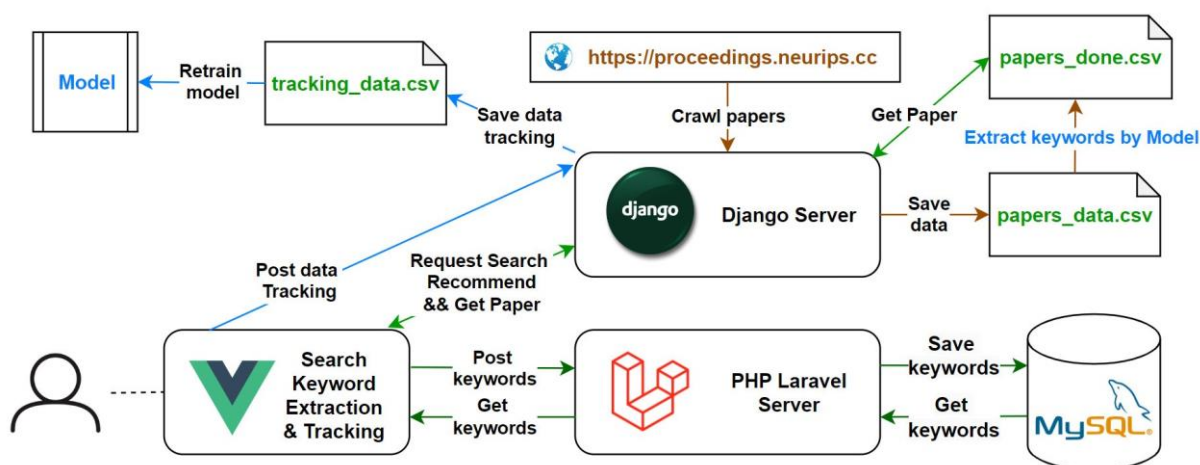
STT	Thuộc Tính	Kiểu Dữ Liệu	Ý nghĩa
1	id	int	Khóa chính để phân biệt các bảng ghi đặt lại mật khẩu
2	email	string	Email của tài khoản cần đặt lại mật khẩu
3	token	string	Mã bí mật
4	role	string	Quyền admin hoặc user
5	created_at	timestamp	Thời gian tạo bảng ghi
6	updated_at	timestamp	Thời gian cập nhật bảng ghi

3.5. Xây dựng hệ thống

3.5.1. Tổng quan về hệ thống

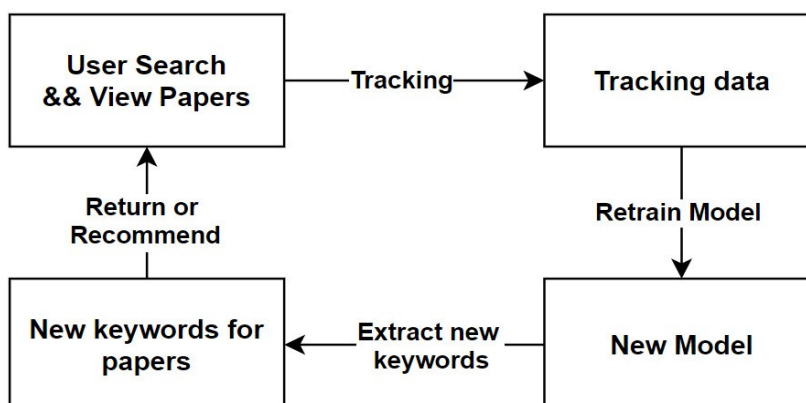
Giải thích một số luồng chính của tổng quan hệ thống:

- Người dùng sẽ tương tác với hệ thống thông qua giao diện sử dụng Vue.js;
- Vue.js sẽ gọi đến các API thuộc 2 Server khác nhau là Django và PHP Laravel;
- Đối với các API liên quan đến model AI thì sẽ được gửi đến Django, sau đó Django sẽ thực hiện truy xuất và update đến các file dữ liệu CSV khác nhau để trả về kết quả API mong muốn, ngoài ra Django còn có hệ thống tự crawl dữ liệu tương tác với website proceedings.neurips.cc;
- Đối với các API cơ bản liên quan đến thông tin cá nhân, mật khẩu thì sẽ tương tác API với PHP Laravel Server để thực hiện các yêu cầu thông qua cơ sở dữ liệu MySQL;
- Dữ liệu tracking_data.csv sẽ được dùng để retrain model để cải thiện hiệu suất.



Hình 3.2 Tổng quan về hệ thống

3.5.2. Quy trình tracking và huấn luyện trong hệ thống



Hình 3.3 Tổng quan quy trình tracking và huấn luyện

Dựa theo hình tổng quan ở trên, ta có một số mô tả chi tiết sau:

- Đây sẽ là vòng lặp tuần hoàn giúp cải thiện hiệu suất của hệ thống.
- Ban đầu model được huấn luyện trên bộ dữ liệu sẵn có sẽ thực hiện trích xuất các từ khoá cho các bài báo chưa có xác định từ khoá.
- Sử dụng các kết quả của model trích xuất từ khoá đó để thực hiện gợi ý hoặc trả về kết quả tìm kiếm cho người dùng.
- Từ đó người dùng thực hiện các hành động với hệ thống để thực hiện tạo các dữ liệu tracking mới.
- Dữ liệu mới này khi đạt đến một số lượng nhất định, đủ để huấn luyện sẽ được lấy và huấn luyện lại cho model.
- Model sau khi huấn luyện sẽ là model mới tốt hơn so với phiên bản trước và lặp lại quá trình trích xuất từ khoá. Cứ tuần hoàn như vậy ta đạt được mục đích về việc cải thiện hiệu suất của model.

Chương 4. Kết quả

Phần này trình bày tập dữ liệu dùng để huấn luyện, các tham số huấn luyện và mô tả quá trình huấn luyện, cuối cùng là kết quả của các baseline model trên các bộ tham số huấn luyện và tinh chỉnh mô hình.

4.1. Tập dữ liệu

Để đánh giá hiệu quả của BERTGraph, chúng tôi đã sử dụng ba bộ dữ liệu bao gồm Inspec [54], SE-2010 [55] và SE-2017 [56]. Tổng 2736 dòng dữ liệu huấn luyện ban đầu. Chúng tôi sẽ mô tả các bộ dữ liệu này trong các đoạn tiếp theo.

Inspec bao gồm các bản tóm tắt của các bài báo về Khoa học Máy tính được thu thập từ năm 1998 đến năm 2002.

SE-2010 chứa đầy đủ các bài báo khoa học được lấy từ Thư viện Kỹ thuật số ACM. Trong thử nghiệm của chúng tôi, chúng tôi đã sử dụng bản tóm tắt của các bài báo.

SE-2017 bao gồm các đoạn được chọn từ 500 bài báo tạp chí ScienceDirect từ các lĩnh vực Khoa học Máy tính, Khoa học Vật liệu và Vật lý.

Cần lưu ý rằng do xây dựng việc trích xuất cụm từ khóa như một nhiệm vụ gắn nhãn theo trình tự, chúng tôi xem xét các cụm từ khóa xuất hiện trong phần tóm tắt của các bài báo trong ba bộ dữ liệu. Bảng sau trình bày số liệu thống kê của ba tập dữ liệu trên.

Bảng 4-1 Số liệu thống kê của ba tập dữ liệu.

Dataset	loại doc	Doc	Keywords (mỗi doc)	Tokens
Inspec	Abstract	2000	29230 (14.62)	128.20
SemEval2010	Abstract	243	4002 (16.47)	149.34
SemEval2017	Paragraph	493	8969 (18.19)	178.22

Nhận xét tập dữ liệu: Số lượng dữ liệu của tập Inspec chiếm đa số (2000 rows) có một chút khác biệt về số lượng tokens mỗi doc, tuy nhiên nếu nhìn tổng quan có thể thấy, số lượng keywords mỗi doc cũng có xu hướng tăng, nếu số lượng tokens tăng lên. Do đó bộ dữ liệu đạt được độ cân bằng nhất định, thuận lợi cho việc huấn luyện và đánh giá chung cho cả 3 tập dữ liệu.

Abstract

Background Keyword extraction is a popular research topic in the field of natural language processing. Keywords are terms that describe the most relevant information in a document. The main problem that researchers are facing is how to efficiently and accurately extract the core keywords from a document. However, previous keyword extraction approaches have utilized the text and graph features, there is the lack of models that can properly learn and combine these features in a best way.

Methods In this paper, we develop a multimodal Key-phrase extraction approach, namely *Phraseformer*, using transformer and graph embedding techniques. In *Phraseformer*, each keyword candidate is presented by a vector which is the concatenation of the text and structure learning representations. *Phraseformer* takes the advantages of recent researches such as BERT and ExEm to preserve both representations. Also, the *Phraseformer* treats the key-phrase extraction task as a sequence labeling problem solved using classification task.

Results We analyze the performance of *Phraseformer* on three datasets including Inspec, SemEval2010 and SemEval 2017 by F1-score. Also, we investigate the performance of different classifiers on *Phraseformer* method over Inspec dataset. Experimental results demonstrate the effectiveness of *Phraseformer* method over the three datasets used. Additionally, the Random Forest classifier gain the highest F1-score among all classifiers.

Conclusions Due to the fact that the combination of BERT and ExEm is more meaningful and can better represent the semantic of words. Hence, *Phraseformer* significantly outperforms single-modality methods.

Keywords: Multimodal representation learning, Keyword extraction, Transformer, Graph embedding

Hình 4.1 Hình ảnh một mẫu dữ liệu Hình ảnh một mẫu dữ liệu

4.2. Các tham số trong quá trình huấn luyện

- Metric sử dụng đánh giá:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$f1 = 2 * \frac{precision * recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- *Loss function*: Cross Entropy với 3 bộ weight khác nhau cho 3 nhãn B, I, O bao gồm [1, 5, 5], [1, 9, 9], [1, 19, 19]. Các trọng số này là thực nghiệm ở 3 mức độ đánh trọng số khác nhau giữa các nhãn lần lượt ở mức thấp, mức trung, mức cao.
- *Optimizer*: Adam with decay (AdamW)
- *Early stop*: Có sử dụng dừng sớm nếu như độ chính xác trên tập validation không

tăng trong 5 *epochs* liên tiếp.

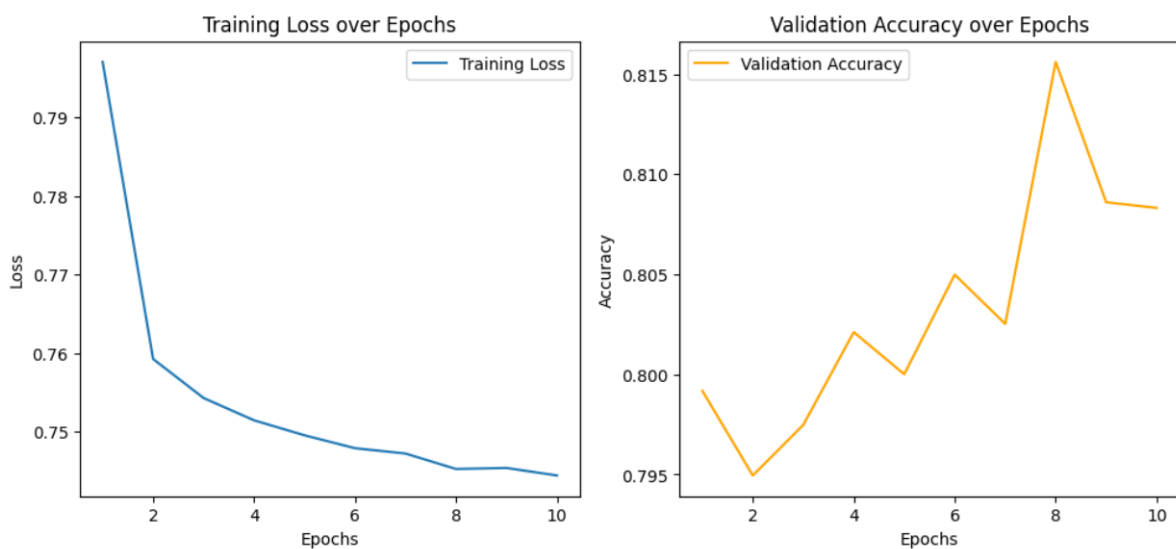
4.3. Baseline model:

BERT: Trong nghiên cứu của chúng tôi, việc xây tái xây dựng một BERT model riêng biệt sẽ khá tốn thời gian và khó đạt hiệu quả như model BERT pretrain trước do hạn chế về data cũng như phần cứng. Do đó chúng tôi sử dụng “google-bert/bert-base-cased” là model cơ sở cho phân học thông tin văn bản.

Word2Vec (WV) + FastText (FT): là các mô hình skip gram + CBOW dùng để học ngữ cảnh của từ và thực hiện nhiệm vụ embedding một từ thành chuỗi vector có độ dài được quy định chứa thông tin ngữ cảnh đã học được.

Word2Vec merge FastText: có thể hiểu đây là sự kết hợp của 2 phương pháp học ngữ cảnh của từ trên, tận dụng được ưu thế của cả 2 phương pháp trên.

4.4. Quá trình huấn luyện



Hình 4.2 Hình ảnh huấn luyện 10 epochs trên tập dữ liệu

Nhận xét: Quá trình huấn luyện cho thấy tốc độ hội tụ nhanh trong vòng chưa đến 10 epochs, sau đó độ chính xác trên tập validation biến động nhưng kết quả không cải thiện. Còn đồ thị loss của tập train vẫn có xu hướng giảm đều và hiệu quả.

4.5. Kết quả

Chú thích:

- $d = \text{vector_size}$ (tổng cho cả 2 WV + FT): Đây là tổng độ dài của cả 2 vector WV và FT, nếu kiến trúc tổng chỉ có một trong 2 vector trên thì có độ dài bằng $d/2$.
- $th = \text{threshold}$: Đây là ngưỡng để xác định loại bỏ từ đó ra khỏi nút của đồ thị

đồng xuất hiện co-occurrence graph. Loại bỏ nếu nhỏ hơn ngưỡng.

- walks = len_random_walks: Số lượng random walks thực hiện trên toàn bộ đồ thị để mô tả thông tin đồ thị.

Bảng 4-2 Kết quả BERT và BERTGraph (F1-score)

Kiến trúc tổng	Weight của loss	KQ (d=200+th=3+ walks=1e4)	KQ (d=400+th=3+ walks = 1e4)	KQ (d=600+th=3+ walks=1e4)	KQ (d=600+th=3+ walks = 1e5)	KQ (d=600+th=2+ walks = 1e5)
BERT	[1, 9, 9]	0.3021	x	x	x	x
	[1, 5, 5]	0.3051	x	x	x	x
	[1, 19, 19]	0.2549	x	x	x	x
BERT + WV	[1, 9, 9]	0.2947	0.3224	0.3274	0.3203	0.3290
	[1, 5, 5]	0.2959	0.3185	0.3308	0.3237	0.3251
	[1, 19, 19]	0.2801	0.3163	0.3175	0.3151	0.3223
BERT + FT	[1, 9, 9]	0.2855	0.3239	0.3296	0.3232	0.3291
	[1, 5, 5]	0.2905	0.3168	0.3325	0.3267	0.3256
	[1, 19, 19]	0.2760	0.3021	0.3164	0.3167	0.3220
BERT + WVFT	[1, 9, 9]	0.2922	0.3289	0.3308	0.3273	0.3324
	[1, 5, 5]	0.2975	0.3205	0.3341	0.3297	0.3324
	[1, 19, 19]	0.2778	0.3082	0.3192	0.3221	0.3275

Lưu ý: Các kết quả ở kiến trúc tổng chỉ là “BERT” sẽ không có các tham số d, th và walks vì không kết hợp với đồ thị nên các tham số đồ thị không ảnh hưởng.

➤ Nhận xét

Từ bảng kết quả ở trên ta có nhận xét:

- Kết quả cao nhất trên thang đo là F1-score là 0.3341 với kiến trúc sử dụng gồm BERT kết hợp với vector graph của cả Word2Vec và FastText có tổng số chiều là 600, ngưỡng chọn từ phổ biến là 3 và số lần random walks là 10000, đồng thời có bộ trọng số của loss là [1, 5, 5].
- Việc chỉ sử dụng BERT không sử dụng graph embedding thì độ chính xác 0.3051 thấp hơn 0.0290 so với kết quả cao nhất khi sử dụng graph embedding.
- Các kết quả trên BERT+WV và BERT+FT được cho là giá trị trung gian, dẫn đến việc kết hợp cả 2 graph embedding này lại đạt hiệu quả cao nhất.
- Với 3 trọng số [1, 5, 5], [1, 9, 9], [1, 19, 19] đều cho kết quả cao nhất trên bộ trọng số thấp [1, 5, 5].
- Các kết quả phản ánh vector_size (độ dài của graph embedding) = 600 sẽ đạt hiệu quả cao nhất, cho thấy vector_size tỉ lệ thuận với F1-score của mô hình.
- Ngưỡng là 3, số random walks là 10000 sẽ phù hợp với bộ dữ liệu nhất.

➤ Giải thích

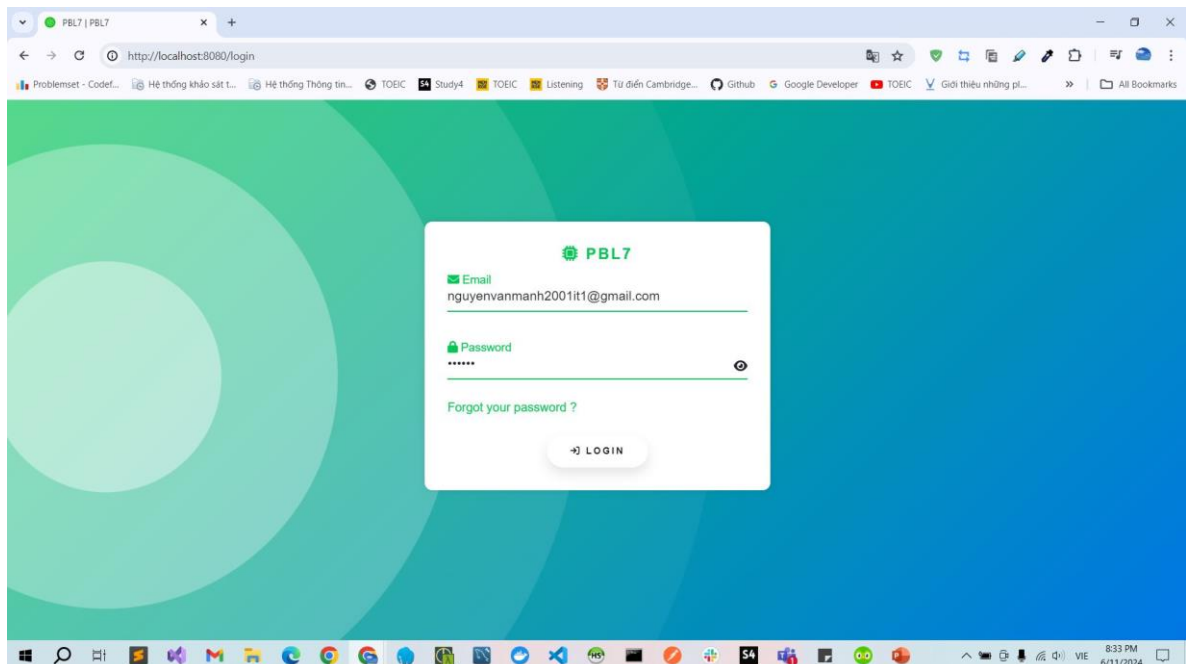
Sau đây là phần giải thích lần lượt cho các nhận xét rút ra từ bảng kết quả trên:

- Kết hợp các tham số tối ưu ta đạt được F1-score cao nhất 0.3341.

- Hiệu quả mà graph embedding mang lại đã cải thiện độ chính xác đáng kể, thay vì chỉ sử dụng ngữ cảnh của từ trong đoạn hiện tại, việc kết hợp với graph embedding là cho từ có được ngữ cảnh của từ đó trong nhiều ngữ cảnh bao quát, trong trường hợp này là toàn bộ các abstract được chọn làm dữ liệu.
- Cả WV và FT nếu đứng riêng lẻ cùng kết hợp với BERT đều mang đến cải thiện tốt, do đó việc kết hợp cả 2 sẽ mang đến kết quả tốt nhất.
- Bộ trọng số thấp [1, 5, 5] sẽ phù hợp để thay thế cho việc không sử dụng bộ trọng số (tức là trọng số [1, 1, 1]) còn các trọng số [1, 9, 9], [1, 19, 19] gây chênh lệch lớn trong việc tối ưu loss function, do đó mang đến kết quả không hiệu quả bằng [1, 5, 5].
- Vector size càng lớn thì thông tin ngữ cảnh của từ trong graph embedding được biểu diễn rõ ràng hơn, do đó độ dài của vector nên đủ để lưu thông tin ngữ cảnh đó. Xem xét đến các yếu tố về độ dài vector thì 768 (độ dài của BERT) và 600 (độ dài của graph embedding) được cho là cân bằng và mang lại hiệu quả.
- Bộ dữ liệu ở đây có số lượng 2736, thì việc mô tả thông tin graph với số lượng phù hợp với bộ dữ liệu là 10000 random walks và ngưỡng để loại bỏ các từ ít xuất hiện là 3. Tùy bộ dữ liệu có thể tăng giảm các tham số này cho thích hợp.

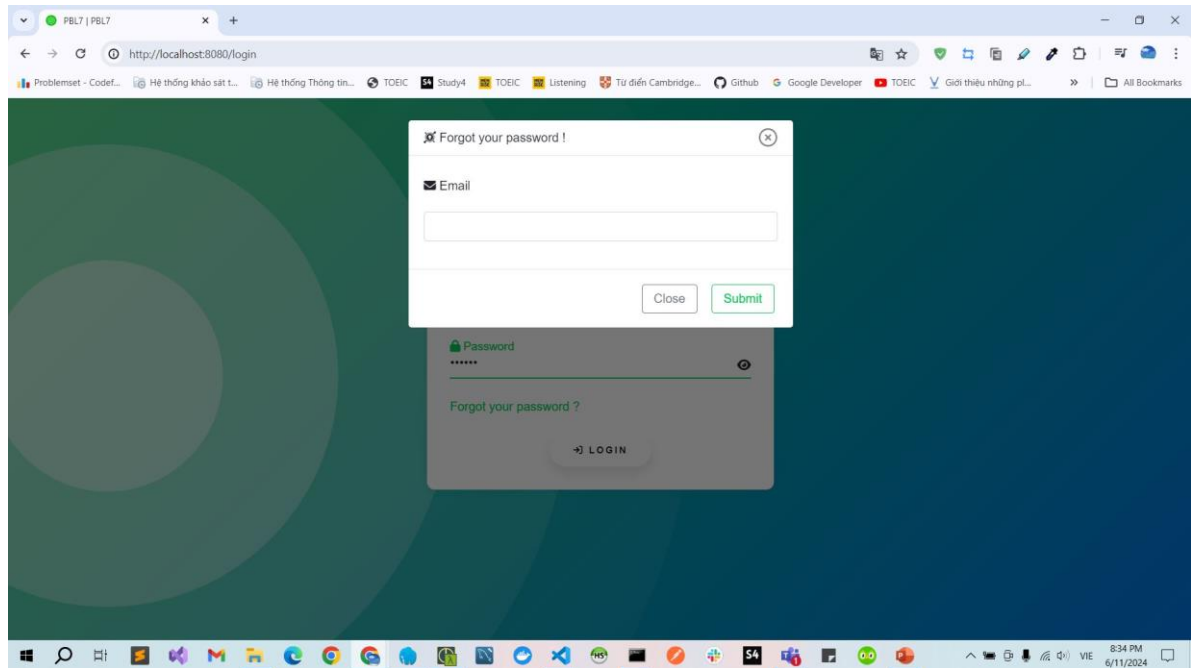
4.6. Kết quả ứng dụng

4.6.1. Chức năng đăng nhập, đăng kí



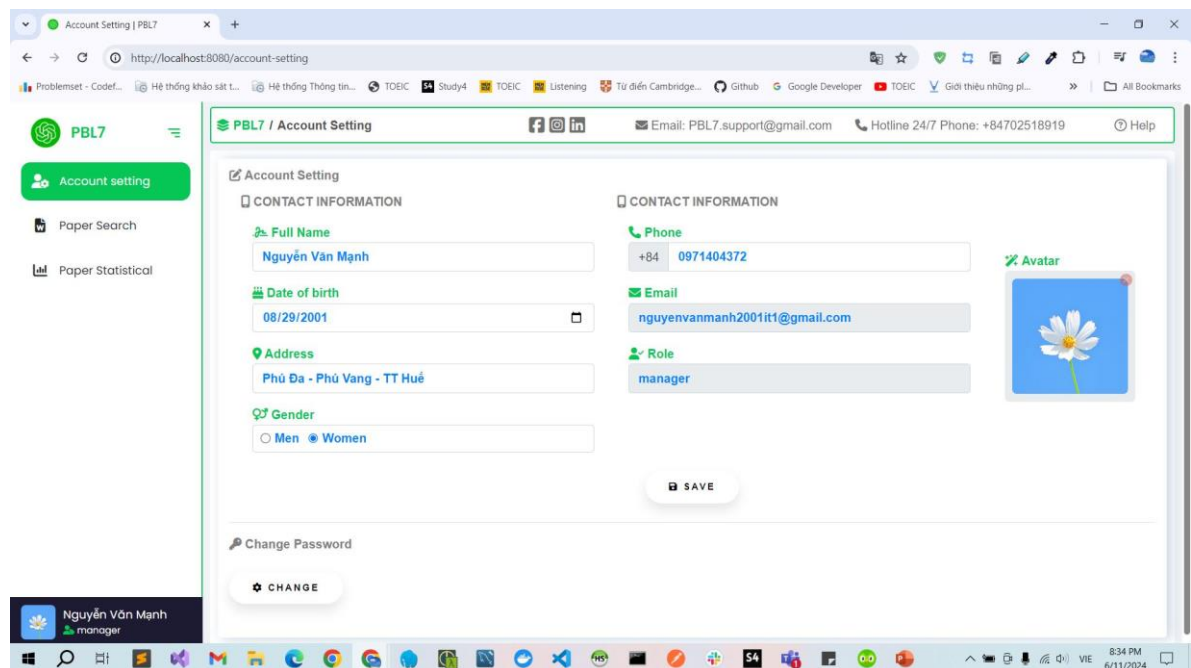
Hình 4.3 Chức năng đăng nhập

4.6.2. Chức năng quên mật khẩu



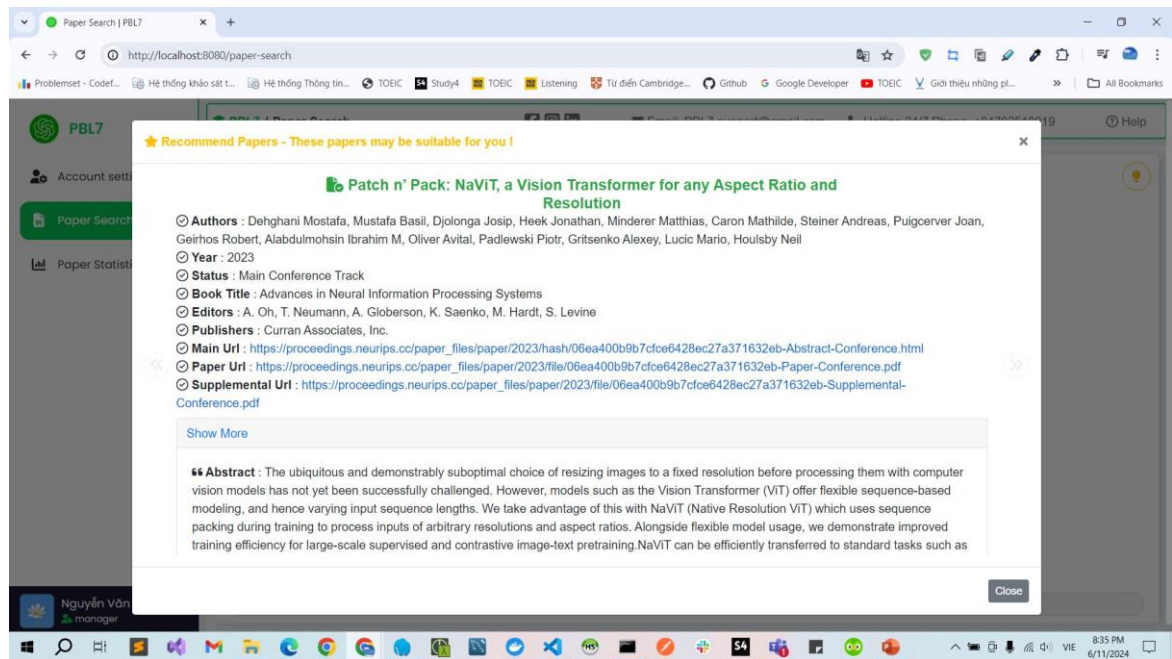
Hình 4.4 Chức năng quên mật khẩu

4.6.3. Chức năng chỉnh sửa thông tin cá nhân



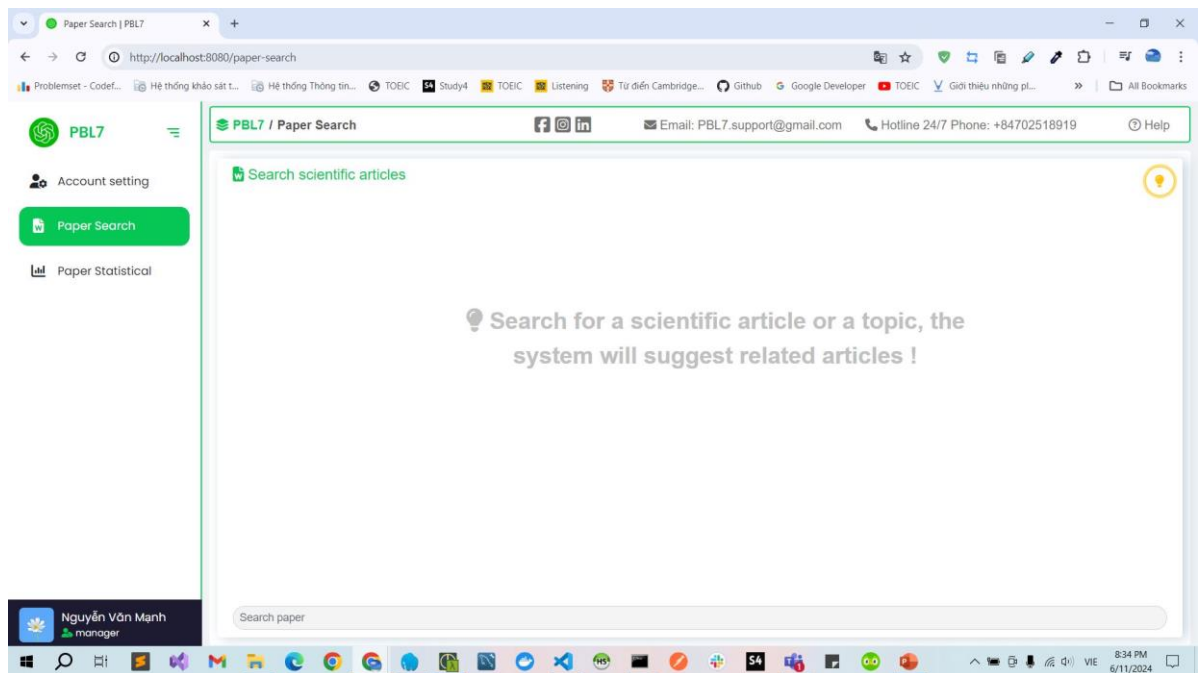
Hình 4.5 Chức năng chỉnh sửa thông tin cá nhân

4.6.4. Chức năng gợi ý bài báo



Hình 4.6 Chức năng gợi ý bài báo

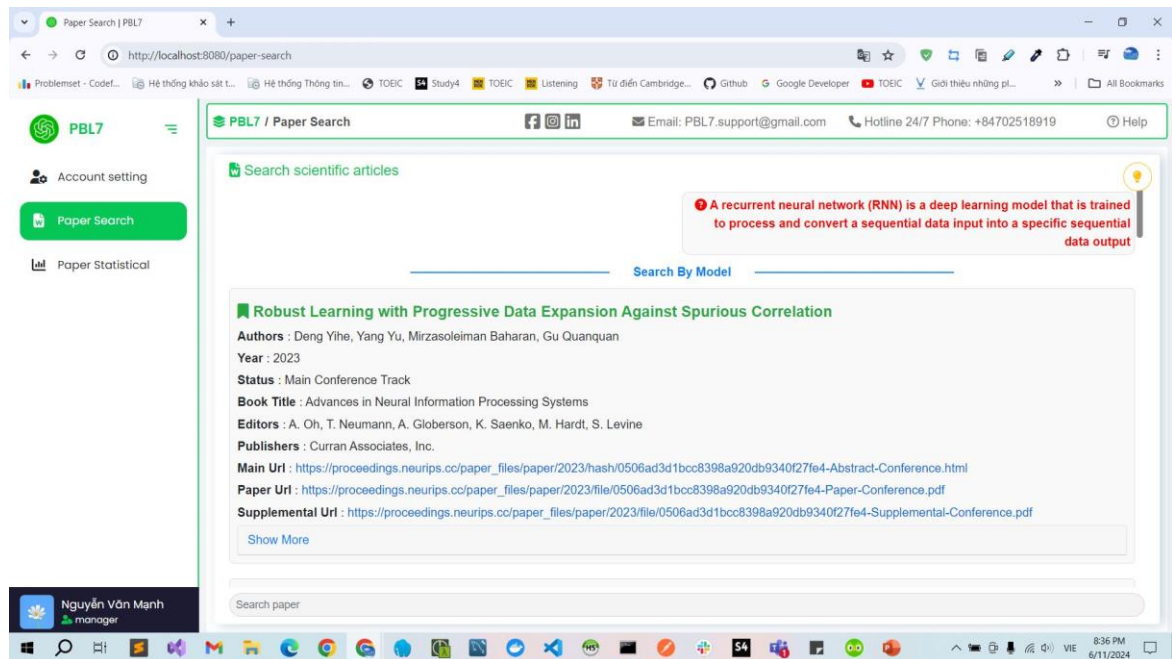
4.6.5. Chức năng tìm kiếm



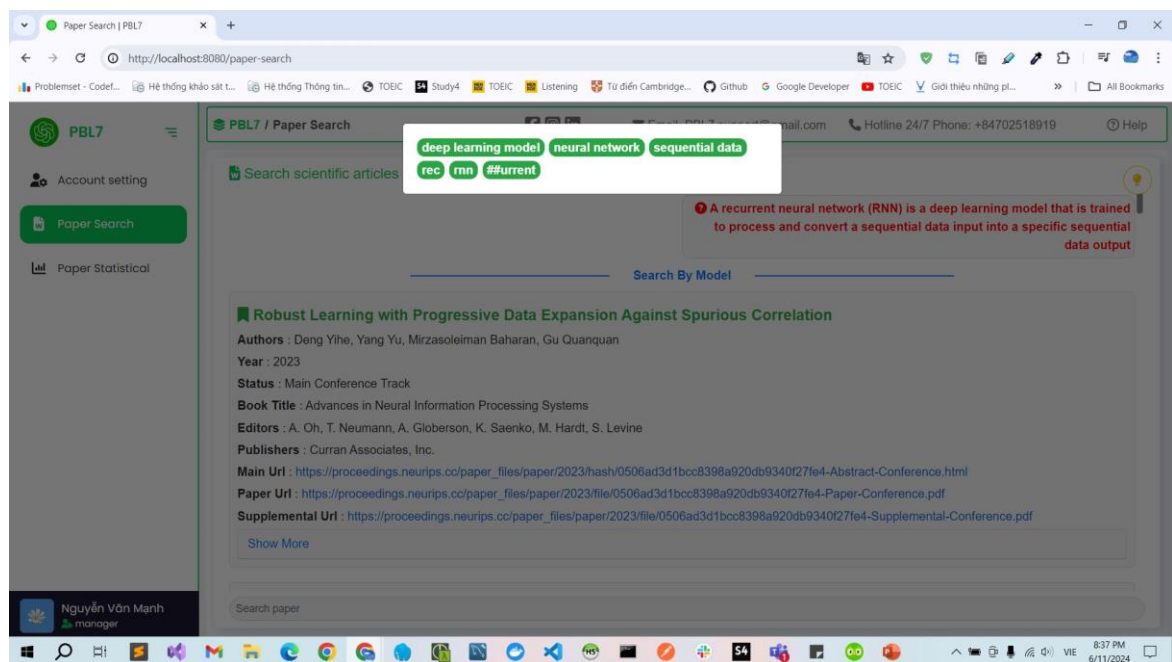
Hình 4.7 Giao diện ban đầu của trang tìm kiếm

Tuỳ thuộc vào cách mà người dùng tìm kiếm hệ thống sẽ hiển thị ra các bài báo theo từng kiểu tìm kiếm khác nhau như sau:

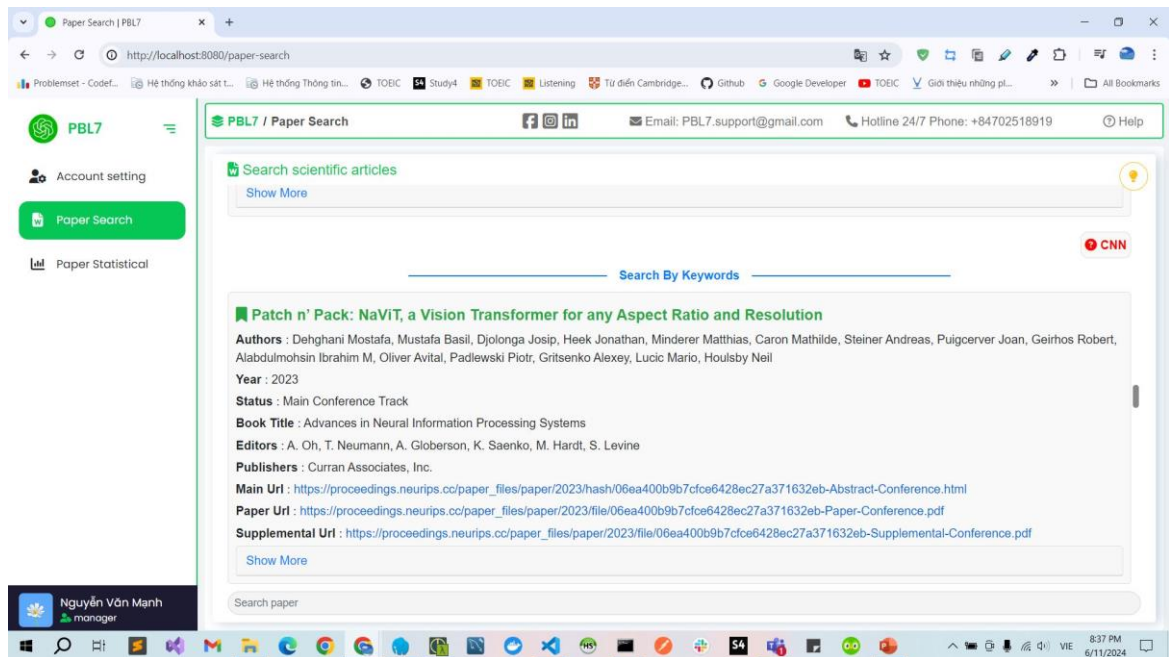
Báo cáo dự án chuyên ngành 2



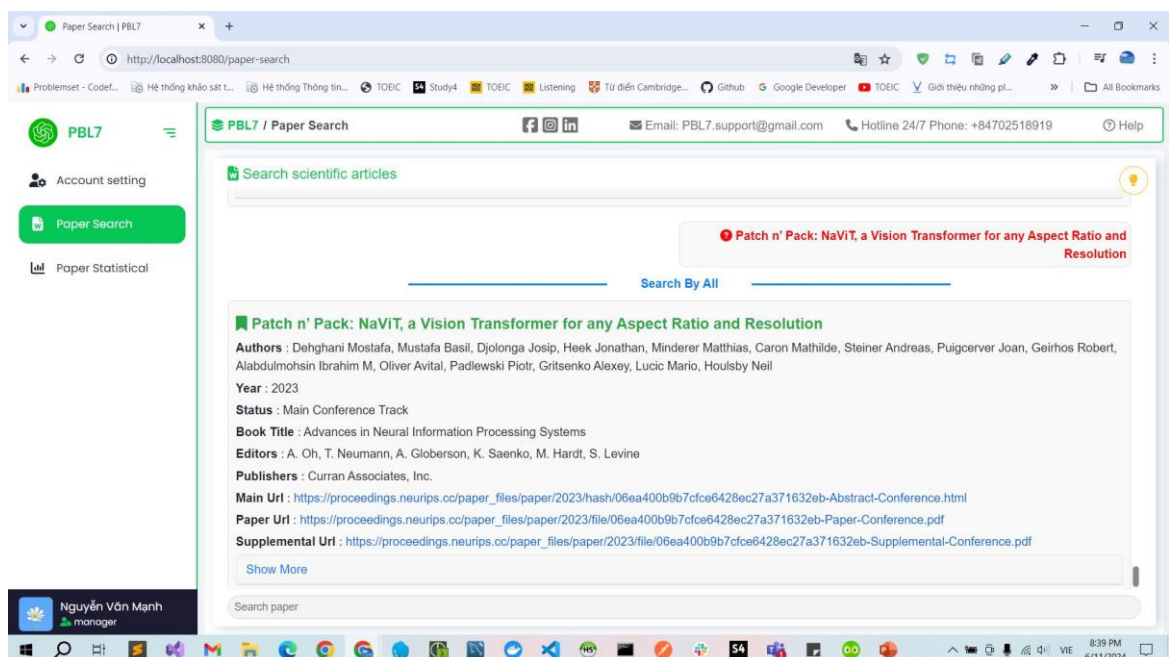
Hình 4.8 Tìm kiếm theo đoạn mô tả



Hình 4.9 Model sẽ trích xuất các keyword trong đoạn mô tả

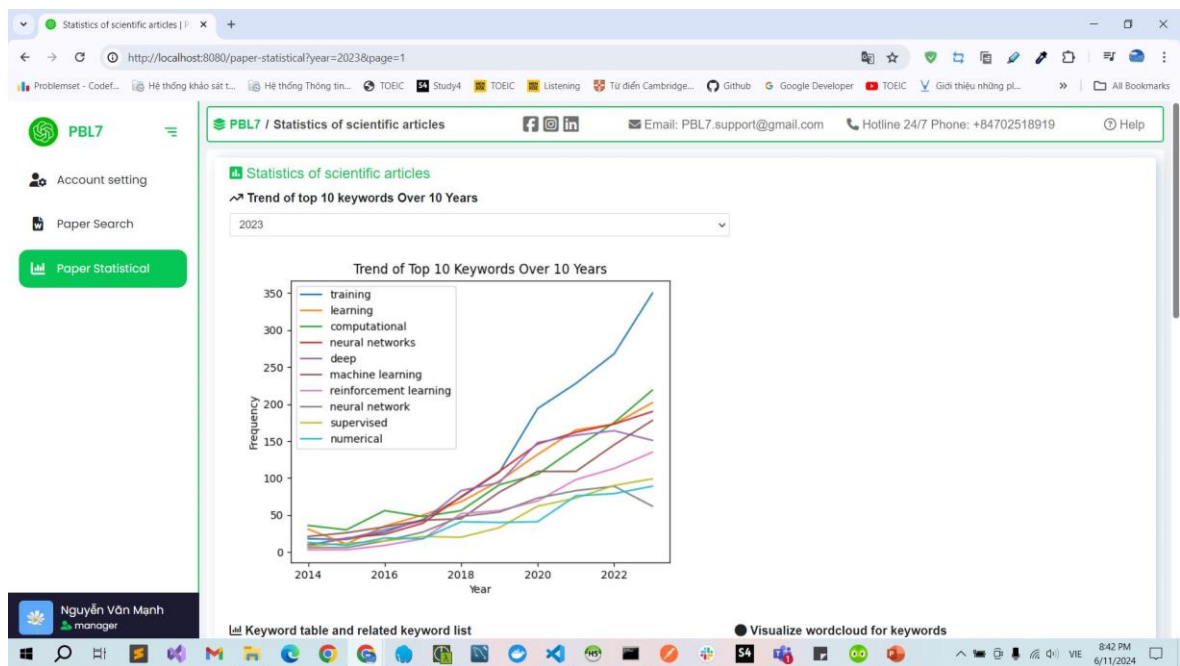


Hình 4.10 Search bởi một keyword duy nhất



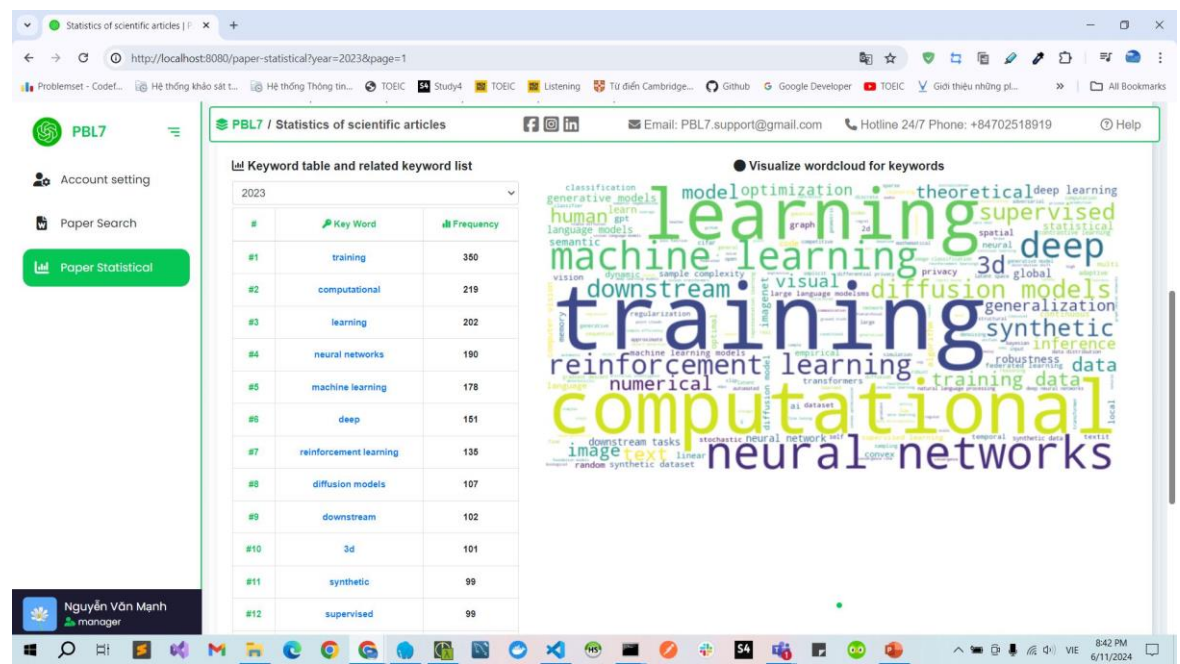
Hình 4.11 Tìm kiếm theo thông tin khác

4.6.6. Chức năng thống kê top 10 từ khoá xu hướng

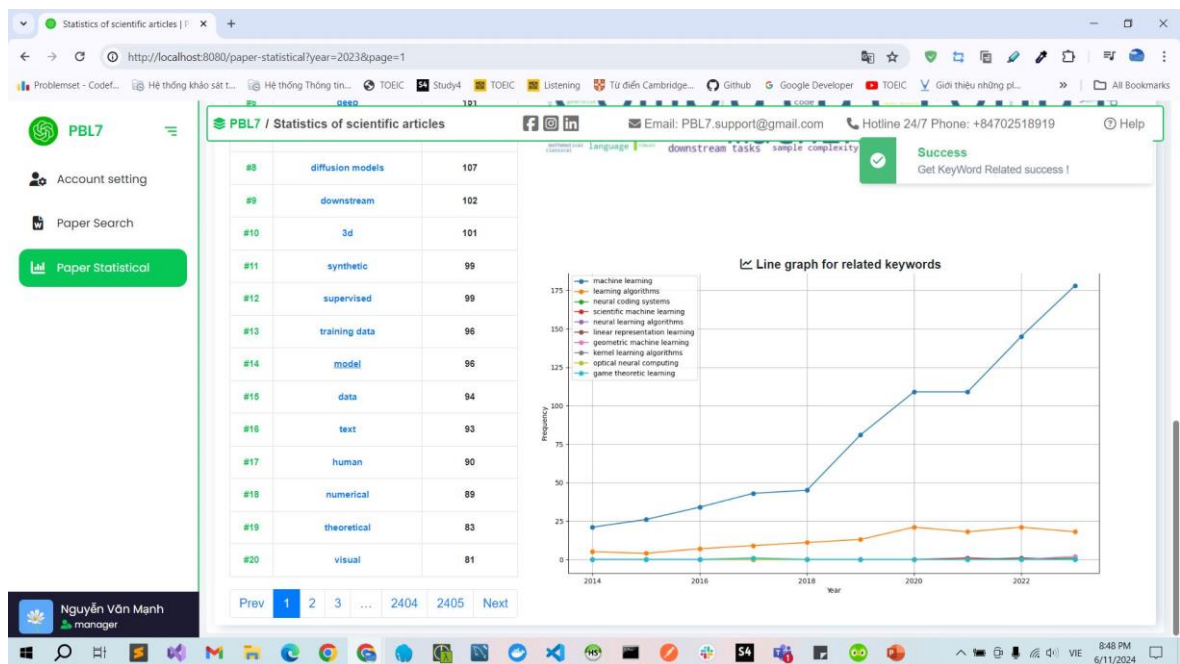


Hình 4.12 Thống kê top 10 từ khoá xu hướng

4.6.7. Chức năng xem bảng xếp hạng xu hướng các từ khoá

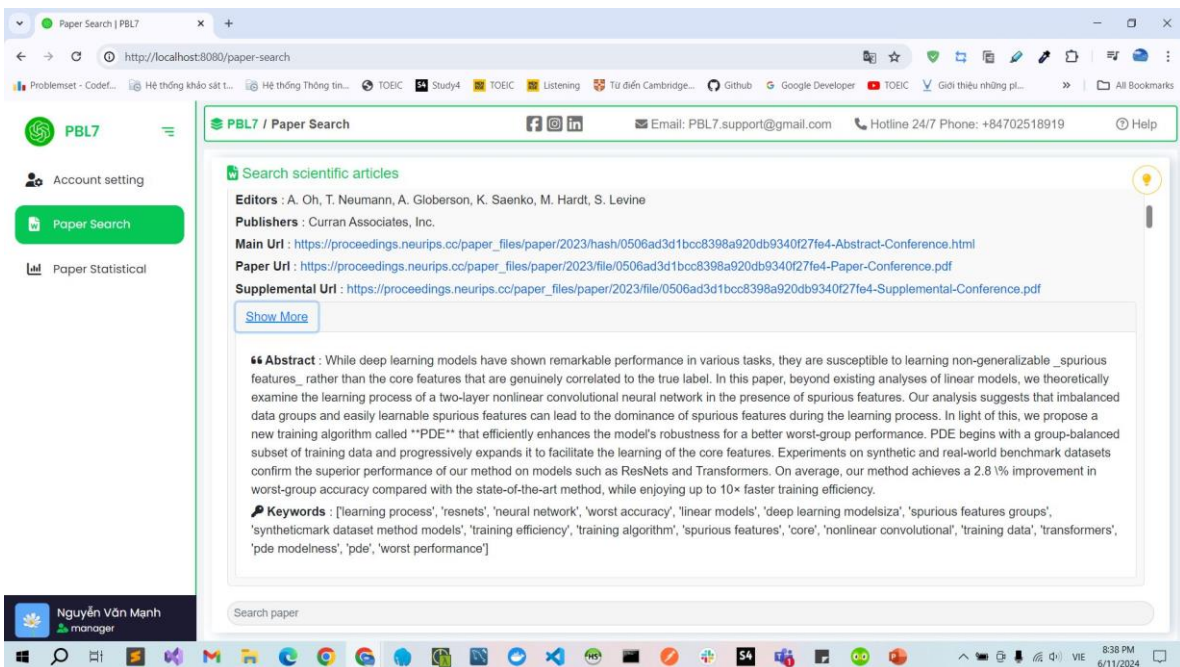


Hình 4.13 Bảng xếp hạng xu hướng từ khoá và wordcloud

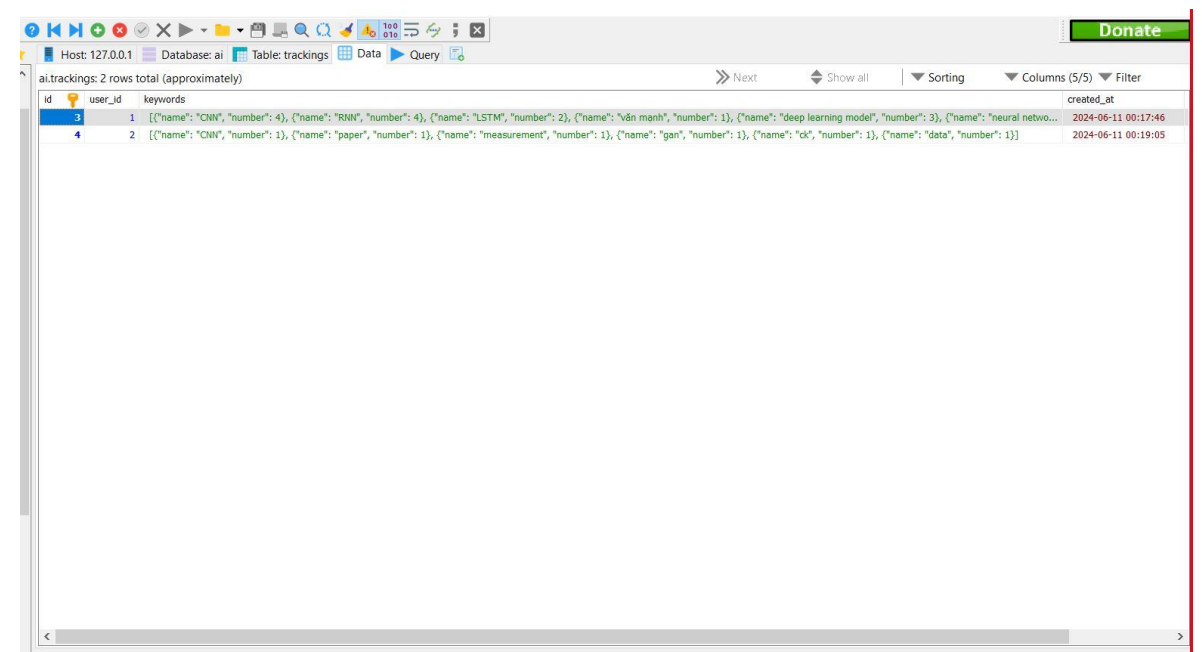


Hình 4.14 Biểu đồ xu hướng top 10 từ khoá

4.6.8. Chức năng tracking lịch sử người dùng



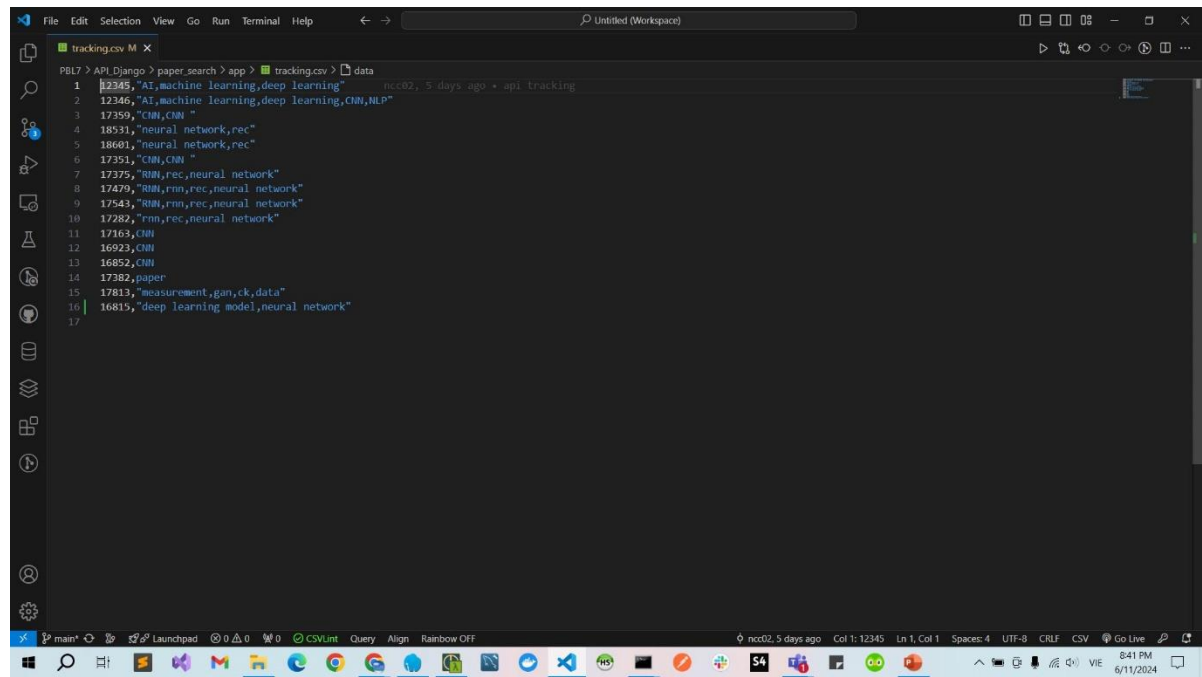
Hình 4.15 Khi người dùng tìm kiếm và chọn vào xem bài báo



id	user_id	keywords	created_at
3	1	[{"name": "CNN", "number": 4}, {"name": "RNN", "number": 4}, {"name": "LSTM", "number": 2}, {"name": "yán mǎnh", "number": 1}, {"name": "deep learning model", "number": 3}, {"name": "neural netwo...	2024-06-11 00:17:46
4	2	[{"name": "CNN", "number": 1}, {"name": "paper", "number": 1}, {"name": "measurement", "number": 1}, {"name": "gan", "number": 1}, {"name": "ck", "number": 1}, {"name": "data", "number": 1}]	2024-06-11 00:19:05

Hình 4.16 Lịch sử tìm kiếm các từ khoá của người dùng

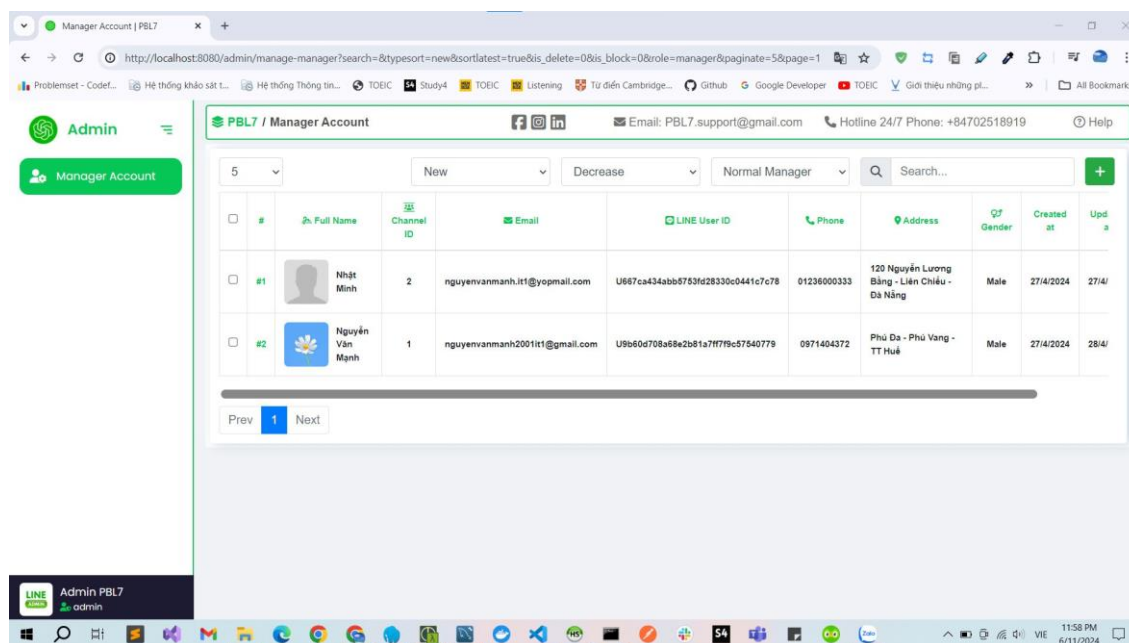
4.6.9. Chức năng tracking để crawl dữ liệu huấn luyện



id	keywords
12345	"AI, machine learning, deep learning"
12346	"AI, machine learning, deep learning, CNN, MLP"
17359	"CNN, CNN"
18531	"neural network, rec"
18601	"neural network, rec"
17351	"CNN, CNN"
17375	"RNN, rec, neural network"
17479	"RNN, rnn, rec, neural network"
17543	"RNN, rnn, rec, neural network"
17282	"rnn, rec, neural network"
17163	"CNN"
16923	"CNN"
16852	"CNN"
17382	"paper"
17813	"measurement, gan, ck, data"
16815	"deep learning model, neural network"

Hình 4.17 Dữ liệu tracking thu được để huấn luyện

4.6.10. Chức năng quản lý người dùng



Hình 4.18 Chức năng quản lý người dùng

Chương 5. Kết luận và hướng phát triển

5.1. Kết quả đạt được:

- Đề xuất một phương pháp đa phương thức gọi là BERTGraph kết hợp học tập ngữ cảnh từ văn bản bằng BERT Transformer và học tập đại diện ngữ cảnh từ đồ thị tương tác bằng kỹ thuật nhúng đồ thị.
- Giải quyết bài toán trích xuất từ khóa như một nhiệm vụ phân đoạn phụ đề, sử dụng encoder BIO.
- Đánh giá hiệu quả của BERTGraph trên kết hợp 3 tập dữ liệu và so sánh với phương pháp dựa trên ngôn ngữ duy nhất (chỉ sử dụng BERT).
- Kết quả cho thấy BERTGraph vượt trội hơn đáng kể so với các phương pháp dựa trên một phương thức.

5.2. Hướng phát triển

- Đổi mới các hướng tiếp cận của Graph embedding thay vì chỉ lọc ra các cụm danh từ, chuyển sang các phương pháp dùng cả động từ, trạng từ,...
- Phát triển model học ngữ cảnh tốt hơn thay BERT thành Roberta, DistilBert,...
- Không ngừng mở rộng tập dữ liệu và từ đó phát triển model trên dữ liệu chuẩn.
- Hướng mới: giải quyết bằng các model multi-task trên bộ dữ liệu lớn (xu hướng hiện nay) và fine-tune trên nhiệm vụ trích xuất keyword.

Danh mục tài liệu tham khảo

- [1] D. A. Vega-Oliveros, P. S. Gomes, E. E. Milios, L. Berton, A multi-centrality index for graph-based keyword extraction, *Information Processing & Management* 56 (6) (2019) 102063. doi:10.1016/j.ipm.2019.102063.
- [2] M. W. Berry, J. Kogan, *Text mining: applications and theory*, John Wiley & Sons, 2010.
- [3] S. Lahiri, *Keywords at work: Investigating keyword extraction in social media applications*, Ph.D. thesis (2018).
- [4] C. Zhang, Automatic keyword extraction from documents using conditional random fields, *Journal of Computational Information Systems* 4 (3) (2008) 1169–1180.
- [5] B. Wang, B. Yang, S. Shan, H. Chen, Detecting hot topics from academic big data, *IEEE Access* 7 (2019) 185916–185927. doi:10.1109/ACCESS.2019.2960285.
- [6] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, C. G. Nevill-Manning, Kea: Practical automated keyphrase extraction, in: *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, IGI global, 2005, pp. 129–152. doi: 10.4018/978-1-59140-441-5.ch008.
- [7] S. R. El-Beltagy, A. Rafea, Kp-miner: A keyphrase extraction system for english and arabic documents, *Information systems* 34 (1) (2009) 132–144. doi:10.1016/j.is.2008.05.002.
- [8] T. D. Nguyen, M.-T. Luong, Wingnus: Keyphrase extraction utilizing document logical structure, in: *Proceedings of the 5th international workshop on semantic evaluation*, 2010, pp. 166–169. URL <https://www.aclweb.org/anthology/S10-1035>
- [9] S. Rose, D. Engel, N. Cramer, W. Cowley, Automatic keyword extraction from individual documents, *Text mining: applications and theory* 1 (2010) 1–20. doi:10.1002/9780470689646.ch1.
- [10] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt, Yake! keyword extraction from single documents using multiple local features, *Information Sciences* 509 (2020) 257–289. doi:10.1016/j.ins.2019.09.013.
- [11] M. Martinc, B. Škrlj, S. Pollak, Tnt-kid: Transformer-based neural tagger for keyword identification, *arXiv preprint arXiv:2003.09166* (2020).
- [12] M. Basaldella, E. Antolli, G. Serra, C. Tasso, Bidirectional lstm recurrent neural network for keyphrase extraction, in: *Italian Research Conference on Digital Libraries*, Springer, 2018, pp. 180–187. doi:10.1007/978-3-319-73165-0_18.
- [13] R. Alzaidy, C. Caragea, C. L. Giles, Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents, in: *The world wide web conference*, 2019, pp. 2551–2557. doi:10.1145/3308558.3313642.

- [14] M. Tang, P. Gandhi, M. A. Kabir, C. Zou, J. Blakey, X. Luo, Progress notes classification and keyword extraction using attention-based deep learning models with bert, arXiv preprint arXiv:1910.05786 (2019).
- [15] J. Wang, F. Song, K. Walia, J. Farber, R. Dara, Using convolutional neural networks to extract keywords and keyphrases: A case study for foodborne illnesses, in: 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), IEEE, 2019, pp. 1398–1403. doi:10.1109/ICMLA.2019.00228.
- [16] Y. Kim, J. H. Lee, S. Choi, J. M. Lee, J.-H. Kim, J. Seok, H. J. Joo, Validation of deep learning natural language processing algorithm for keyword extraction from pathology reports in electronic health records, Scientific Reports 10 (1) (2020) 1–9. doi:10.1038/s41598-020-77258-w.
- [17] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, in: Proceedings of the 2004 conference on empirical methods in natural language processing, 2004, pp. 404–411. URL <https://www.aclweb.org/anthology/W04-3252>
- [18] X. Wan, J. Xiao, Collabrank: towards a collaborative approach to single-document keyphrase extraction, in: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), 2008, pp. 969–976. URL <https://www.aclweb.org/anthology/C08-1122>
- [19] M. Litvak, M. Last, H. Aizenman, I. Gobits, A. Kandel, Degext—a language-independent graph-based keyphrase extractor, in: Advances in intelligent web mastering–3, Springer, 2011, pp. 121–130. doi:10.1007/978-3-642-18029-3_13.
- [20] A. Bellaachia, M. Al-Dhelaan, Ne-rank: A novel graph-based keyphrase extraction in twitter, in: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Vol. 1, IEEE, 2012, pp. 372–379. doi:10.1109/WI-IAT.2012.82.
- [21] A. Bougouin, F. Boudin, B. Daille, TopicRank: Graph-based topic ranking for keyphrase extraction, in: Proceedings of the Sixth International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Nagoya, Japan, 2013, pp. 543–551. URL <https://www.aclweb.org/anthology/I13-1062>
- [22] C. Florescu, C. Caragea, Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1105–1115. doi:10.18653/v1/P17-1102.
- [23] F. Boudin, A comparison of centrality measures for graph-based keyphrase extraction, in: Proceedings of the sixth international joint conference on natural language processing, 2013, pp. 834–838. URL <https://www.aclweb.org/anthology/I13-1102>
- [24] W. D. Abilhoa, L. N. De Castro, A keyword extraction method from twitter messages represented

- as graphs, *Applied Mathematics and Computation* 240 (2014) 308–325. doi:10.1016/j.amc.2014.04.090.
- [25] A. Tixier, F. Malliaros, M. Vazirgiannis, A graph degeneracy-based approach to keyword extraction, in: *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 1860–1870. doi:10.18653/v1/D16-1191.
- [26] M. S. El BazzI, D. Mammass, T. Zaki, A. Ennaji, A graph-based ranking model for automatic keyphrases extraction from arabic documents, in: *Industrial conference on data mining*, Springer, 2017, pp. 313–322. doi:10.1007/978-3-319-62701-4_25.
- [27] F. Boudin, Unsupervised keyphrase extraction with multipartite graphs, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, p. 667–672. doi:10.18653/v1/N18-2105.
- [28] S. Danesh, T. Sumner, J. H. Martin, Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction, in: *Proceedings of the fourth joint conference on lexical and computational semantics*, 2015, pp. 117–126. doi:10.18653/v1/S15-1013.
- [29] Y. Zhang, Y. Chang, X. Liu, S. D. Gollapalli, X. Li, C. Xiao, Mike: keyphrase extraction by integrating multidimensional information, in: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1349–1358. doi:10.1145/3132847.3132956.
- [30] D. Mahata, J. Kuriakose, R. Shah, R. Zimmermann, Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 634–639. doi: 10.18653/v1/N18-2100.
- [31] D. Mahata, R. R. Shah, J. Kuriakose, R. Zimmermann, J. R. Talburt, Theme-weighted ranking of keywords from text documents using phrase embeddings, in: *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, IEEE, 2018, pp. 184–189. doi:10.1109/MIPR.2018.00041.
- [32] S. Siddiqi, A. Sharan, Keyword and keyphrase extraction techniques: a literature review, *International Journal of Computer Applications* 109 (2) (2015).
- [33] Z. A. Merrouni, B. Frikh, B. Ouhbi, Automatic keyphrase extraction: a survey and trends, *Journal of Intelligent Information Systems* (2019) 1–34doi:10.1007/s10844-019-00558-9.
- [34] Ö. Ünlü, A. Çetin, A survey on keyword and key phrase extraction with deep learning, in: *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, IEEE, 2019, pp. 1–6. doi:10.1109/ISMSIT.2019.8932811.
- [35] E. Papagiannopoulou, G. Tsoumakas, A review of keyphrase extraction, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10 (2) (2020) e1339. doi:10.1002/widm.1339.

- [36] N. Firoozeh, A. Nazarenko, F. Alizon, B. Daille, Keyword extraction: Issues and methods, *Natural Language Engineering* 26 (3) (2020) 259–291. doi:10.1017/S1351324919000457.
- [37] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, Deep learning based text classification: A comprehensive review, *arXiv preprint arXiv:2004.03705* (2020).
- [38] M. Asgari-Chenaghlu, M.-R. Feizi-Derakhshi, M.-A. Balafar, C. Motamed, et al., Topicbert: A transformer transfer learning based memory-graph approach for multimodal streaming social media topic detection, *arXiv preprint arXiv:2008.06877* (2020).
- [39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *arXiv preprint arXiv:1706.03762* (2017).
- [40] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [41] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018). URL <https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf>
- [42] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, *OpenAI blog* 1 (8) (2019) 9.
- [43] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, *arXiv preprint arXiv:1906.08237* (2019).
- [44] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, *arXiv preprint arXiv:1802.05365* (2018).
- [45] M. Asgari-Chenaghlu, M. R. Feizi-Derakhshi, L. Farzinvash, C. Motamed, A multimodal deep learning approach for named entity recognition from social media, *arXiv preprint arXiv:2001.06888* (2020).
- [46] N. Nikzad-Khasmakhi, M. Balafar, M. R. Feizi-Derakhshi, C. Motamed, Berters: Multimodal representation learning for expert recommendation system with transformer, *arXiv preprint arXiv:2007.07229* (2020).
- [47] N. Nikzad-Khasmakhi, M. Balafar, M. R. Feizi-Derakhshi, C. Motamed, Exem: Expert embedding using dominating set theory with deep learning approaches, *arXiv preprint arXiv:2001.08503* (2020).
- [48] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. arXiv:1607.00653, doi:10.1145/2939672.2939754.
- [49] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: Online learning of social representations, in:

Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. arXiv:1403.6652, doi:10.1145/2623330.2623732.

[50] Z. He, Z. Wang, W. Wei, S. Feng, X. Mao, S. Jiang, A survey on recent advances in sequence labeling from deep learning models, arXiv preprint arXiv:2011.06727 (2020).

[51] A. Akhundov, D. Trautmann, G. Groh, Sequence labeling: A practical approach, arXiv preprint arXiv:1808.03926 (2018).

[52] J. Kupiec, Robust part-of-speech tagging using a hidden markov model, Computer speech & language 6 (3) (1992) 225–242. doi:10.1016/0885-2308(92)90019-Z.

[53] J. Lafferty, A. McCallum, F. C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: 18th International Conference on Machine Learning 2001 (ICML 2001), 2001, p. 282–289. doi: 10.5555/645530.655813.

[54] A. Hulth, Improved automatic keyword extraction given more linguistic knowledge, in: Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003, pp. 216–223. doi:10.3115/1119355.1119383.

[55] S. N. Kim, O. Medelyan, M.-Y. Kan, T. Baldwin, Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles, in: Proceedings of the 5th International Workshop on Semantic Evaluation, 2010, pp. 21–26.

[56] I. Augenstein, M. Das, S. Riedel, L. Vikraman, A. McCallum, Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications, in: 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, p. 546–555. doi:10.18653/v1/S17-2091.

[57] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[58] D. Sahrawat, D. Mahata, M. Kulkarni, H. Zhang, R. Gosangi, A. Stent, A. Sharma, Y. Kumar, R. R. Shah, R. Zimmermann, Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings, arXiv preprint arXiv:1910.08840 (2019).