Hệ thống khuyến nghị các bài báo khoa học

Tieu Kim Hao[1,1*], Van Kim Ngan[1,1] and Nguyen Thi Bao Han[1,1]

[1*]University of Information Technology Vietnam National University, Ho Chi Minh City, Viet Nam.

*Corresponding author(s). E-mail(s): 19521480@gm.uit.edu.vn;
Contributing authors: 19520177@gm.uit.edu.vn;
19520071@gm.uit.edu.vn;

Tóm tắt nội dung

Ngày nay, những nhà nghiên cứu tốn khá nhiều thời gian và công sức để tìm ra bài báo phù hợp với chủ đề mà họ đang đang nghiên cứu. Vấn đề sẽ càng khó khăn hơn khi một nhà nghiên cứu là người mới tìm hiểu hoặc không có đủ kiến thức để tìm kiếm các bài báo nghiên cứu liên quan. Vì vậy, làm thế nào để khuyến nghị được những bài báo khoa học mà họ đang tìm kiếm? Trong các phương pháp khuyến nghị truyền thống, kết quả của truy vấn bỏ lỡ nhiều bài báo chất lượng cao, được xuất bản gần đây hoặc có số lượng trích dẫn thấp. Trong bài báo này, mục tiêu của chúng tôi là phát triển một hệ thống đề xuất bài báo nghiên cứu với chất lượng được cải thiện.

Keywords: Hệ thống khuyến nghị, bài báo khoa học, stemming

1 Giới thiệu

Sự quá tải về số lượng thông tin hiện hữu trên Internet làm cho quá trình tìm kiếm thông tin trở thành một nhiệm vụ phức tạp. Các nhà nghiên cứu cảm thấy khó khăn trong việc truy cập và theo dõi các tài liệu nghiên cứu có liên quan và mang tính cấp thiết mà họ quan tâm. Cách tiếp cận được biết đến phổ biến nhất để có được tài liệu nghiên cứu phù hợp là theo dõi danh sách tài liệu tham khảo được đề cập trong các tài liệu họ đã sở hữu. Mặc dù phương pháp này có thể khá hiệu quả trong một số trường hợp, tuy nhiên nó không

đảm bảo bao phủ đầy đủ các tài liệu nghiên cứu được đề xuất, ngoài ra còn cản trở việc theo dõi các tài liệu tham khảo trong trường hợp bài báo gốc bị chiếm hữu.

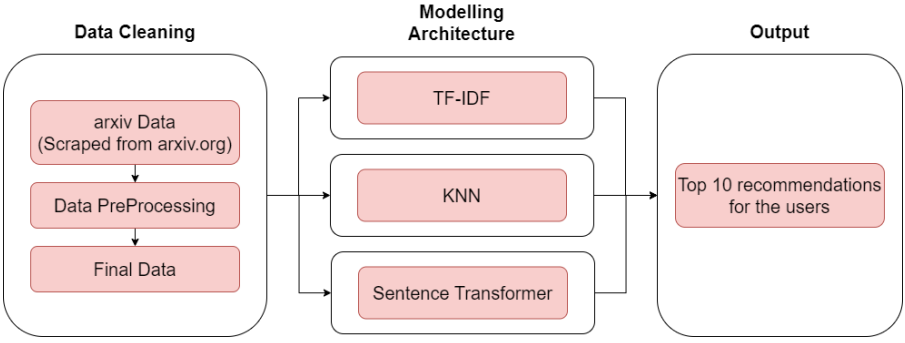
Tại đây, chúng tôi đã triển khai các phương pháp tiếp cận dựa trên nội dung, trong đó, các thuật toán sẽ so sánh nội dung và đồng thời khai thác các liên kết ẩn giữa tài liệu mục tiêu với các tham chiếu của nó bằng cách sử dụng một số phương pháp cộng tác.

Mục tiêu của chúng tôi là xác định các tương quan tiềm ẩn tồn tại giữa các bài báo khoa học, dựa trên mối quan hệ giữa các paper-citation. Đầu tiên, chúng tôi sẽ tìm kiếm sự tương đồng giữa bài báo mục tiêu đối với danh sách các bài báo, điều này giúp chúng tôi phát hiện ra các top đầu bài báo tương quan với nhau và sau đó, các bài báo này được xem xét để rút ra danh sách các bài báo tiêu biểu sử dụng tài liệu tham khảo và trích dẫn.

Một bài báo có thể được gọi là có liên quan nếu bài báo đó có thể trích dẫn đến bất kỳ tài liệu tham khảo nào của bài báo mục tiêu, sau đó đề xuất top-N các bài báo nếu các bài báo đó có sự giống nhau một cách đáng kể. Sự nghiêm ngặt trong việc xem xét sự tương quan giữa các bài báo giúp nâng cao hiệu suất tổng thể của phương pháp và khả năng trả về kết quả khuyến nghị sẽ có liên quan và hữu ích hơn.

Hình 1 dưới đây mô tả quy trình chung bài toán khuyến nghị của nhóm chúng tôi. Bài toán gồm các bước sau:

- Data Cleaning.
- Ba mô hình nhóm sẽ sử dụng: TF-IDF, KNN, Sentence Transformer.
- Output là top 10 bài báo liên quan.



Hình 1: Hình mô tả quy trình chung của bài toán khuyến nghị.

Các phần của báo cáo bao gồm các mục như sau. Phần 2 sẽ trình bày khảo sát các nghiên cứu liên quan. Phần 3 giới thiệu bộ dữ liệu cũng như các phương pháp tiền xử lý. Phần 4 sẽ nói về các thuật toán mà nhóm sẽ sử dụng. Phần 5 trình bày cách thiết lập thực nghiệm, kết quả thực nghiệm và quan sát kết quả. Phần 6 Kết luận và hướng phát triển.

2 Khảo sát các nghiên cứu liên quan

Các phương pháp khuyến nghị đang được sử dụng phổ biến hiện nay theo [1] bao gồm: hybrid based recommendation, lọc dựa trên nội dung, lọc cộng tác, citation based, sử dụng Google PageRank. Sau khi tìm hiểu các phương pháp khuyến nghị khác, chúng tôi nhận thấy rằng rằng mỗi phương pháp đều có những ưu điểm và nhược điểm riêng.

Chúng tôi đã nghiên cứu ưu nhược điểm của các phương pháp trên một cách kỹ lưỡng. Từ đó chọn lọc và thực hiện xây dựng hệ khuyến nghị các bài báo khoa học của chính mình bằng cách sử dụng phương pháp dựa trên mô hình lai và đa mô hình. Bảng 1 dưới đây mô tả tóm tắt cuộc khảo sát các nghiên cứu liên quan của chúng tôi.

Phương pháp	**Hạn chế**
KNN	Vấn đề Cold start (1) Đối với người dùng mới (new user problem): chưa đánh giá sản phẩm nào, chưa có các dữ liệu về các hành vi, sở thích của họ. (2) Đối với sản phẩm mới (new item problem): chưa được người dùng nào đánh giá, chưa được ai xem, đã mua, hay tìm kiếm, ...
TF-IDF	Tiêu hao nhiều thời gian và có thể sẽ trở nên khó khăn hơn khi kích cỡ danh sách tài liệu là cực kỳ lớn.
Sentence Transformer	Vấn đề đối với phương pháp này sẽ phát sinh khi không có trích dẫn trong văn bản tương ứng với các tài liệu tham khảo được thêm vào danh sách tài liệu tham khảo. Những trích dẫn này được gọi là trích dẫn sai và những trích dẫn như vậy cũng dẫn đến kết quả không phù hợp.
Google's PageRank	Phương pháp này gặp hạn chế do việc sử dụng tổng số trích dẫn làm chỉ số để gợi ý các bài báo. Điều này sẽ gây thất bại trong việc đề xuất các bài báo chất lượng khi bài báo xuất bản gần đây được chọn là bài báo có sức ảnh hưởng.
Content based filtering	Mô hình chỉ có thể đưa ra các đề xuất dựa trên sở thích hiện có của người dùng. Nói cách khác, mô hình bị hạn chế khả năng mở rộng nếu dựa trên sở thích của người dùng.

Bảng 1: Bảng so sánh, đánh giá hạn chế của các phương pháp khuyến nghị.

3 Dữ liệu và Tiền xử lý Dữ liệu

3.1 Bộ dữ liệu

Bộ dữ liệu này đã được nhóm tự thu thập, chúng tôi đã thu thập các bài báo nghiên cứu khoa học từ website www.arxiv.org. Từ trang web này, nhóm đã thu thập được tiêu đề, tóm tắt, tên tác giả, URL của các bài báo, tài liệu tham khảo, trích dẫn và từ khóa của bài báo.

Ở đây, nhóm sử dụng các giao diện lập trình ứng dụng (Application Programming Interface) viết tắt là API, được cung cấp bởi website arxiv để hỗ

trợ trong việc thu thập và trích xuất dữ liệu theo những định dạng có cấu trúc như XML hay JSON.

Sau khi thu thập, bộ dữ liệu của nhóm gồm có **10.000** điểm dữ liệu với năm thuộc tính sau: *Title, Date, Id, Summary, URL.*

3.2 Tiền xử lý dữ liệu

Vì số lượng n-gram khá nhiều và đòi hỏi khả năng tính toán của máy là rất lớn để thực hiện việc tìm kiếm. Cho nên việc tiền xử lý dữ liệu là một bước quan trọng giúp làm giảm kích thước n-gram và tối ưu hóa thuật toán. Các bước tiền xử lý được thực hiện tuần tự như sau.

3.2.1 Loại bỏ Null và các ký tự đặc biệt

Dữ liệu của nhóm là các bài báo khoa học, cho nên summary của các bài báo sẽ có các ký tự đặc biệt như các ký hiệu toán học, ký tự vô nghĩa,... Ở đây nhóm đã tiền xử lý dữ liệu bằng cách loại bỏ các ký tự đặc biệt trên.

Bên cạnh đó, dữ liệu cũng sẽ có các giá trị Null, Sparse Data, một số bài báo chỉ có mỗi tiêu đề hay thậm chí không có tiêu đề và tóm tắt. Vì vậy nhóm đã loại bỏ các giá trị Null, các điểm dữ liệu không có tiêu đề hay tóm tắt.

3.2.2 Thuật toán Stemming

Theo như nghiên cứu, thuật toán Stemming có khả năng cải thiện hiệu suất của mọi information retrieval system [5]. Stemming là kỹ thuật dùng để biến đổi một từ về dạng gốc (được gọi là stem hoặc root form) bằng cách cực kỳ đơn giản là loại bỏ một số ký tự nằm ở cuối từ mà nó nghĩ rằng là biến thể của từ. Có thể lấy ví dụ đơn giản như các từ ***worked, working, works*** chỉ khác nhau ở những ký tự cuối cùng, bằng cách bỏ đi các hậu tố ***-ed, -ing, -s,*** chúng ta đều được từ nguyên gốc là ***work***. Bởi vì nguyên tắc hoạt động của Stemming rất đơn giản nên tốc độ xử lý của nó rất nhanh và kết quả stem đôi khi cho ra những kết quả không như mong muốn.

Một ví dụ khác như từ ***goes*** sẽ được stem thành từ ***goe*** (bỏ chữ s cuối từ) trong khi đó stem của từ go vẫn là go, kết quả là 2 từ goes và go sau khi được stem thì vẫn không giống nhau. Một nhược điểm khác là nếu các từ dạng bất quy tắc như ***went*** hay ***spoke*** thì kỹ thuật Stemming sẽ không thể đưa các từ này về dạng gốc là ***go*** hay ***speak***. Tuy có các nhược điểm như trên nhưng trong thực tế Stemming vẫn được sử dụng khá phổ biến trong NLP vì nó có tốc độ xử lý nhanh và kết quả cuối cùng nhìn chung không hề tệ khi so với Lemmatization.

Phương pháp Stemming bao gồm hai bước: **thu thập hậu tố** và **stemming**. Bước một sẽ mô tả cách mà các hậu tố được thu thập thông qua việc so sánh giữa các từ giống nhau. Chi tiết của bước một sẽ được mô tả thông qua đoạn mã giả algorithm 1 dưới đây.

Sau khi loại bỏ các từ trùng lặp, lúc này các từ còn lại là duy nhất, sẽ được sắp xếp theo quy luật alphabet và theo độ dài của từ. Bước này giúp làm giảm số lần so sánh matching giữa các cặp từ trong quá trình stemming.

Hậu tố sẽ được tạo ra bằng cách so sánh giữa các từ có chữ cái giống nhau lần lượt từ trái sang phải. Nếu như cặp từ này có số lượng chữ cái khác nhau nhỏ hơn bằng ba, thì phần khác nhau của từ (sau khi loại bỏ các ký tự giống nhau) sẽ là hậu tố. Ví dụ, cặp từ **Change** và **Changed** chỉ có duy nhất một ký tự **-d** khác nhau. Do đó **-d** sẽ được xem xét có phải là thành phần hậu tố hay không bằng cách thêm d vào **danh sách hậu tố (suffix list)**. Ngược lại, nếu số lượng chữ cái không trùng khớp giữa các từ lớn hơn ba thì có thể chắc chắn rằng các chữ cái này là hậu tố. Các hậu tố sau khi được thu thập sẽ được tiến hành kiểm tra một cách thủ công trước khi áp dụng vào mô hình.

Bước thứ hai mô tả cách stemming hoạt động bằng cách sử dụng các hậu tố đã thu thập được và xóa tất cả inflected words (Paul → Paul's). Mã giả để chuẩn hóa các từ sẽ có trong algorithm 1.

Sau khi áp dụng phương pháp stemming, **assessment** đã được chuẩn hóa thành **assess**. Mốt số ví dụ khác có thể quan sát thấy như các từ 'change', 'changer', 'changes', 'changing' đều sẽ được chuẩn hóa thành 'change'.

Có tổng cộng **31,713** unigrams trong corpus sau khi loại bỏ null và các ký tự đặc biệt. Sau khi áp dụng kỹ thuật unsupervised stemming, unigram size giảm còn **23,178** (giảm ∼8500 unigrams). Tuy nhiên trong quá trình stemming nhóm nhận thấy có xuất hiện một số lỗi. Từ goes sẽ được stem thành từ goe, trong khi đó stem của từ go vẫn là go vì vậy trong trường hợp này đã stemming ra từ không giống nghĩa với từ gốc.

4 Các thuật toán

4.1 TF-IDF và Cosine Similarity

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

Trước tiên, chúng tôi chuyển tập huấn luyện về dạng ma trận với số dòng là độ dài, và số cột là tầng suất xuất hiện của từng từ trong tập huấn luyện (Unigram). Sau đó, chúng tôi sử dụng TF-IDF cho ma trận này và có được một ma trận để đánh giá.

Tiếp theo, chúng tôi đưa tập test về các vector có số chiều bằng với số chiều của các vector trong tập huấn luyện và tính cosine similarity của từng vector trong tập test với từng vector trong tập huấn luyện. Với mỗi vector trong tập test, chúng tôi sẽ chọn ra 10 samples có consine similarity score cao nhất để đánh giá sau cùng.

Algorithm 1 Pseudo code for unsupervised stemming

procedure COLLECTING SUFFIX
Store all unique words in *dict* after sorting them alphabetically and length wise
for each $word_i$ in *dict* do **do**
    for each $word_j$ in *dict* do **do**
        if $word_j$.startswith($word_i$) and len($word_j$)−len($word_i$) ≤ 3 **then**
            $word_j$ is inflected form of $word_i$
            suffix list.append($word_j$.replace($word_i$))
        else
            if $word_j$.startswith($word_i$) and len($word_j$)−len($word_i$) ≥ 4 **then**
                diff suffix list.append($word_j$.replace($word_i$))
            else
                countinue
Each *suffix* in *diff suffix list* is manually checked

procedure STEMMING
for each $word_i$ in *dict* do **do**
    for each $word_j$ in *dict* do **do**
        if $word_j$.startswith($word_i$) **then**
            x ←word$_j$.replace($word_i$)
            if x in *diff suffix list* **then**
                $word_j$ is removed from *dict*
            end if
        else
            countinue

4.2 KNN

K-Nearest Neighbors (KNN) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.

Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

Trước tiên, chúng tôi sử dụng hàm CountVectorizer trong thư viện Sklearn để chuyển các mẫu trong tập huấn luyện về dạng count vector, sau đó dùng hàm NearestNeighbors để huấn luyện mô hình, sau đó tập test cũng được chuyển về dạng count vector để đánh giá. Với mỗi vector trong tập test, chúng tôi sẽ chọn ra 10 samples có consine similarity score cao nhất để đánh giá sau cùng.

4.3 Sentence Transformer

Sentence Transformer là một Python framework cho state-of-the-art sentence, text và image embeddings. Các bước ban đầu được mô tả cụ thể trong [2]

Chúng ta có thể sử dụng framework để tính sentence/text embeddings cho hơn 100 ngôn ngữ. Các embeddings có thể được so sánh với nhau bằng cosine similarity để tìm ra các câu có độ tương đồng về ngữ nghĩa. Điều này rất hữu ích cho các bài toán như là semantic textual similar, semantic search, hoặc paraphrase mining.

Framework được viết dựa trên thư viện PyTorch và Transformers, và cung cấp số lượng lớn các pre-trained models với nhiều bài toán khác nhau, hơn nữa, nó rất dễ để fine-tune trên model của bạn.

Nhóm sử dụng pre-trained model paraphrase-MiniLM-L3-v2 với model size là 61MB và speed là 19000, chúng tôi chọn model nhẹ nhất trong thư viện, vì chúng tôi quan tâm nhiều về tốc độ của mô hình. Sau đó, Với mỗi vector trong tập test, chúng tôi sẽ chọn ra 10 samples có consine similarity score cao nhất để đánh giá sau cùng.

5 Thực nghiệm

5.1 Thiết lập thực nghiệm

Trong nghiên cứu này, nhóm tiến hành xây dựng mô hình khuyến nghị dựa trên 3 thuật toán: KNN, TF-IDF and Cosine Similarity, và Sentence Transformer. Cả 3 mô hình đều được nhóm áp dụng phương pháp Stemming trong quá trình đào tạo. Mô hình thực nghiệm được đào tạo trên bộ dữ liệu nhóm thu thập được ở mục 3.1, tỉ lệ dữ liệu sử dụng cho đào tạo và kiểm thử lần lượt là 0.999, 0.001. Trong quá trình đào tạo, các mô hình được cài đặt trên các bộ tham số mặc định với số n-gram là 1. Độ đo nhóm sử dụng là Cosin Similarity. Đầu ra của mô hình sẽ chọn lấy top 10 bài báo được cho là liên quan nhất với bài báo gốc. Tất cả các thực nghiệm trên được chúng tôi thiết lập trên môi trường google colab cùng với GPU được cung cấp sẵn.

5.2 Đánh giá

Về thang đo đánh giá, nhóm thực hiện đánh giá thủ công bằng cách cho 10 người tham gia đánh giá. Mỗi người gãn nhãn sẽ đánh giá 1 bài báo gốc với 10 bài báo được mô hình recommend và đánh giá trên cả 3 phương pháp. Người gán nhãn sẽ phải đọc tóm của bài báo được đề xuất lẫn bài báo gốc, từ đó có thể so sánh mức độ liên quan giữa 2 bài báo với nhau. Dựa trên sự tương đồng của bài báo được đề xuất với bài báo gốc, người tham gia đánh giá sẽ gán nhãn cho từng cặp với nhãn 0 là không liên quan, nhãn 1 là liên quan.

Dựa trên kết quả bảng 3 ta có thể thấy số lượng những bài báo được mô hình Sentence Transformer đề xuất có tỉ lệ chính xác cáo hơn hẳn 2 mô hình còn lại ($0.46 > 0.36 > 0.34$). Kết quả này được thống kê dựa trên số lượng nhãn 1 từ dữ liệu được gán nhãn thủ công.

Bài báo gốc	Bài báo được đề xuất	Nhãn
A new kind of Adversarial Example	Transferability Ranking of Adversarial Examples	1
	Towards Adversarial Purification using Denoising AutoEncoders	1
	Adversarial Vulnerability of Temporal Feature Networks for Object Detection	0
	Probabilistic Categorical Adversarial Attack & Adversarial Training	0
	Boosting the Transferability of Adversarial Attacks with Reverse Adversarial Perturbation	0
	On the Adversarial Transferability of ConvMixer Models	0
	Stateful Detection of Adversarial Reprogramming	1
	Friendly Noise against Adversarial Noise: A Powerful Defense against Data Poisoning Attacks	1
	Membership Inference Attacks Against Text-to-image Generation Models	1
	Nowhere to Hide: A Lightweight Unsupervised Detector against Adversarial Examples	0

Bảng 2: Minh họa việc gán nhãn 0,1 giữa từng cặp bài báo được đề xuất với bài báo gốc.

Mô hình	**Tỉ lệ**
KNN	0.34
TF-IDF	0.36
Sentence Transformer	0.46

Bảng 3: Bảng kết quả tỉ lệ số lượng bài báo đề xuất bởi mô hình được cho là có liên quan tới bài báo gốc dựa trên đánh giá của người tham gia đánh giá trên cả 3 mô hình.

5.3 Kết quả

Áp dụng stemming

TF-IDF		**KNN**		**Sentence Transformer**	
Paper	**Cosin Similarity**	**Paper**	**Cosin Similarity**	**Paper**	**Cosin Similarity**
1	0.336843	1	0.286149888	1	0.6892434418
2	0.378926	2	0.360650346	2	0.7009475708
3	0.472766	3	0.28261254	3	0.7115779
4	0.296526	4	0.229613741	4	0.69795923
5	0.268042	5	0.275099055	5	0.60604696
6	0.349001	6	0.44261648	6	0.71300960
7	0.34887	7	0.384431429	7	0.773307478
8	0.257381	8	0.487117583	8	0.70645249
9	0.323475	9	0.269686374	9	0.720546949
10	0.29321	10	0.396515267	10	0.653846371
	0.332504		**0.34144927**		**0.69729381**

Bảng 4: Bảng kết quả tương đồng giữa những bài báo được đề xuất với bài báo gốc của 3 mô hình KNN, TF-IDF, và Sentence Transformer có áp dụng phương pháp stemming, thang đo được sữ dụng là Cosine Similarity

Như trong bảng 4, chúng tôi thu được độ tương đồng trung bình giữa những bài báo được đề xuất bởi mô hình so với bài báo gốc trên cả 3 mô hình thực nghiệm. Ta có thể thấy được mô hình sử dụng Sentence Transformer hoạt động tốt hơn, Cosine Similarity của mô hình Sentence Transformer cao hơn gấp đôi so với 2 mô hình còn lại, điều này cho ta thấy được rằng những bài báo được đề xuất bởi mô hình Sentence Transformer thì . Dựa trên thống kê trong bảng 3, tỉ lệ đề xuất những bài báo có liên quan của mô hình Sentence Transformer cũng cao hơn hẳn 2 mô hình còn lại ($0.46 > 0.36 > 0.34$). Mô hình này hoạt động tốt hơn vì nó sử dụng vì nó sử dụng mô hình pre-train, đã

Không áp dụng stemming

TF-IDF		KNN		Sentence Transformer	
Paper	**Cosin Similarity**	**Paper**	**Cosin Similarity**	**Paper**	**Cosin Similarity**
1	0.309826	1	0.285572101	1	0.691350436
2	0.296853	2	0.349597842	2	0.682801461
3	0.35188	3	0.28029342	3	0.706672275
4	0.270421	4	0.242760213	4	0.682360959
5	0.23245	5	0.268552846	5	0.543643534
6	0.287358	6	0.423564217	6	0.658626813
7	0.321103	7	0.380502157	7	0.694353259
8	0.233424	8	0.471257252	8	0.673009169
9	0.278197	9	0.256509796	9	0.696698451
10	0.236436	10	0.363013336	10	0.635615587
	0.2817948		**0.332162318**		**0.666513194**

Bảng 5: Bảng kết quả tương đồng giữa những bài báo được đề xuất với bài báo gốc của 3 mô hình KNN, TF-IDF, và Sentence Transformer không áp dụng phương pháp stemming, thang đo được sữ dụng là Cosine Similarity

học qua một lượng lớn dữ liệu, điều này giúp mô hình học chính xác hơn ngữ cảnh của câu, để từ đó mô hình có thể đưa ra được những đề xuất gần nhất với nội dung bài báo gốc. Việc đưa phương pháp Stemming vào mô hình cũng được chứng minh là hiệu quả. Khi ta so sánh kết quả của bảng 4 và bảng 5, ta có thể thấy khi áp dụng stemming vào mô hình, kết quả thu được tốt hơn so với kết quả khi không áp dụng stemming.

6 Kết luận và hướng phát triển

6.1 Kết luận

Với mục đích tìm hiểu và nghiên cứu những phương pháp khuyến nghị các bài báo khoa học, nhóm đã xây dựng được một mô hình khuyến nghị các bài báo khoa học dựa trên summary của chúng. Mô hình Sentence Transformer của nhóm đã ghi nhận được được kết quả tốt nhất (khoảng 46% trăm số bài báo được đề xuất phù hợp) trên cả 3 mô hình. Phương pháp Stemming nhóm áp dụng vào mô hình cũng đạt hiệu quả, phương pháp này giúp làm giảm số lượng unigram. Từ đó tăng tính hiệu quả của mô hình, giúp mô hình cải thiện thời gian tính toán cũng như hiệu suất.

Mặc dù kết quả mà nhóm đạt được khá ổn, nhưng mô hình nhìn chung vẫn còn nhiều điểm hạn chế. Kết quả đánh giá nhìn chung vẫn còn bị lệ thuộc vào góc nhìn lẫn tư duy của người đánh giá, số lượng người tham gia đánh giá của nhóm vẫn còn ít nên việc đánh giá chưa thật sự mang lại kết quả đáng tin cậy như kì vọng của nhóm.

6.2 Hướng phát triển

Về hướng phát triển, nhóm sẽ cố gắng hoàn thiện mô hình khuyến nghị những bài báo khoa học cũng như có khả năng đào tạo được trên số n-gram cao hơn như bigram, trigram,.. cho ra những bài báo được khuyến nghị mang tính chính xác cao hơn. Khắc phục những điểm còn hạn chế trong sẽ là ưu tiên

hàng đầu của nhóm trong việc hoàn thiện nghiên cứu này. Nhóm cũng sẽ cố gắng tăng độ tin cậy của việc đánh giá tính chính xác của mô hình khuyến nghị bằng cách tăng cường số lượng người tham gia đánh giá, cho nhiều người cùng đánh giá trên 1 cặp bài báo gốc - đề xuất hơn so với hiện tại nhằm tăng mức độ tin cậy của dữ liệu đánh giá.

Ngoài ra, nhóm còn hướng tới việc khuyến nghị trên nhiều ngôn ngữ khác nhau, từ đó có thể phát triển mô hình khuyến nghị trên nền tảng đa ngôn ngữ. Trong tương lai xa hơn, mô hình khuyến nghị này sẽ được phát triển thành một ứng dụng thực tế có khả năng đề xuất các bài báo khoa học chính xác nhất phục vụ cho việc nghiên cứu của mọi người.

Tài liệu

[1] Beel, J. (2015, July 26). Research Paper Recommender systems: A literature survey. SpringerLink.

[2] Reimers, N. (2019, August 27). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv.org.

[3] Natural Language Processing (ICON-2017) (pp. 290-297). Gupta, S. (2021, August 20). Overview of Text Similarity Metrics in Python - Towards Data Science. Medium.

[4] Haruna, K. (2017, October 5). A collaborative approach for research paper recommender system. PLOS ONE.

[5] Cristian Moral, Angelica de Antonio, Ricardo Imbert, and Jaime Ramírez. (2014). A survey of stemming algorithms in information retrieval. Information Research: An International Electronic Journal, 19(1)

[6] Beel, J., Langer, S., Genzmehr, M., Nürnberger, A. (2013, July). Introducing Docear's research paper recommender system. In JCDL (pp. 459-460).

[7] Waheed, W., Imran, M., Raza, B., Malik, A. K., Khattak, H. A. (2019). A Hybrid Approach Toward Research Paper Recommendation Using Centrality Measures and Author Ranking. IEEE Access, 7, 33145-33158.

[8] Neethukrishnan, K. V., Swaraj, K. P. (2017, February). Ontology based research paper recommendation using personal ontology similarity method. In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-4). IEEE.

[9] Patra, B. G., Das, D., Bandyopadhyay, S. (2017, December). Retrieving similar lyrics for music recommendation system. In Proceedings of the 14th International Conference on