Xây Dựng Hệ Khuyến Nghị Cho Diễn Đàn Hỏi Đáp

Recommender Systems for Q&A Forum

Dương Thị Hồng Hạnh, Trần Thị Mỹ Linh,
Huỳnh Ngọc Tín, Huỳnh Văn Tín
University of Information Technology, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
{18520711, 18520999,}@gm.uit.edu.vn
{tinhn,tinhv}@uit.edu.vn

Tóm tắt. Trong bài báo cáo này, chúng tôi tiến hành xây dựng hệ khuyến nghị phục vụ cho diễn đàn hỏi đáp tiếng Việt. Cụ thể là mang đến những sự khuyến nghị chất lượng nhất cho người dùng về những vấn đề mà họ có thể đang quan tâm. Chúng tôi thực hiện thu thập và xây dựng nên bộ dữ liệu hữu ích sử dụng cho quá trình nghiên cứu. Tiến hành các bước tiền xử lý cơ bản và sử dụng các kiến thức hỗ trợ về Word Embedding, Clustering và Topic modeling để xây dựng nên các hệ khuyến nghị theo hai phương pháp chính là Content-based và Collaborative Filtering kết hợp với Neural Network. Cuối cùng tiến hành đánh giá kết quả thu được cùng và xác định các hướng phát triển tiếp theo trong tương lai.

1 Giới thiệu

Diễn đàn hỏi đáp trực tuyến không còn là một khái quá xa lạ đối với chúng ta trong cuộc sống hiện đại ngày nay. Nó được biết đến như một nơi mà bất kỳ ai có Internet đều có thể chia sẻ về những thông tin, những vấn đề mà bản thân gặp phải hoặc đang quan tâm. Những cuộc thảo luận, trao đổi nhóm trên diễn đàn sẽ giúp giải đáp thắc mắc một cách nhanh chóng và thuận tiện từ đó thúc đẩy được sự giao lưu, trao đổi, học hỏi và phát triển hiểu biết của mỗi cá thể tham gia diễn đàn. Và chính bởi những giá trị tốt đẹp trên đã khiến diễn đàn trở thành một nền tảng hỗ trợ vững chắc cho lĩnh vực học thuật và giáo dục.

Ngày nay, trên thế giới nói chung và Việt Nam nói riêng, ngày càng có nhiều diễn đàn hỏi đáp ra đời với đa dạng các lĩnh vực hỗ trợ khác nhau. Mỗi diễn đàn như một kho lưu trữ lớn và sẽ thật tốt nếu như chúng ta luôn tìm thấy được thứ bản thân đang quan tâm mà không cần tốn nhiều thời gian. Chính vì vậy, với mong muốn mang lại những trãi nghiệm thật tốt cho người dùng trên diễn đàn cũng như hỗ trợ xây dựng nên các diễn đàn chất lượng, chúng tôi đã quyết định tìm hiểu và nghiên cứu về việc xây dựng hệ khuyến nghị sử dụng cho các diễn đàn hỏi đáp dựa trên các kiến thức học được. Cụ thể, thứ chúng tôi muốn mang lại là một hệ thống khuyến nghị cá nhân hóa được không gian diễn đàn cho mỗi đối tượng người dùng, đưa học đến được với những thông tin mà họ đang thật sự cần.

Trong báo cáo đồ án này, chúng tôi tập trung vào giới thiệu các thông tin liên quan đến mục tiêu xây dựng hệ khuyến nghị tại mục 1 và mục 2 là nơi trình bày về bài toán mà chúng tôi đặt ra. Tiếp theo, ở mục 3, các thông tin về thu thập và xây dựng bộ dữ liệu được thể hiện chi tiết. Trong mục 4 là các giải pháp được áp dụng để tiếp cận vấn đề và đồng thời, các thử nghiệm và kết quả sẽ được đánh giá, phân tích ở mục 5. Mục 6 sẽ là kết luận và hướng phát triển trong tương lai cho các bài toán khuyến nghị. Và cuối cùng, phần demo và source code được thể hiện tại mục 7.

2 Bài toán

Mục tiêu chính của đồ án này là xây dựng hệ khuyến nghị cho diễn đàn thảo luận, hỏi đáp. Ở đồ án này chúng tôi tập trung vào việc khuyến nghị những bài viết, chủ đề mà người dùng có khả năng sẽ quan tâm khi họ truy cập diễn đàn, dựa vào lịch sử tương tác của người dùng. Bài toán được miêu tả cụ thể với đầu ra và đâu vào như sau:

- Đầu vào: Id của người dùng trong hệ thống.
- Đầu ra: Đề xuất bài viết, chủ đề liên quan.

3 Bộ dữ liệu

3.1 Thu thập dữ liệu

Với mục tiêu xây dựng hệ khuyến nghị sử dụng cho các diễn đàn học thuật tại Viêt Nam, chúng tôi thực hiện thu thập dữ liệu từ *vfo.vn* - một diễn đàn hỏi dáp được thành lập lâu đời và hoạt động khá sôi nổi tại Việt Nam. Vfo được biết đến với nhiều diễn đàn con thuộc các lĩnh vực khác nhau (xe, máy ảnh, công nghệ, lập trình, đồ họa, giải trí,...) và những bài viết tại đây có thể phân thành hai dạng chính: bài viết chứa nội dung chia sẽ kiến thức, kinh nghiệm và bài viết về hỏi đáp. Với mục tiêu xây dựng một hệ thống khuyến nghị phục vụ cho việc hỏi đáp, giúp người dùng tiếp cận được những bài viết có nội dung họ đang quan tâm cũng như đưa những họ đến với những bài viết mà họ có khả năng mang lại những câu trả lời chất lượng và nhanh chóng nhất, chúng tôi đã quyết định thực hiện thu thập dữ liệu tại diễn đàn con Hỏi đáp nhanh của Vfo. Tại đây có hơn 136,000 bài viết và bình luận được đăng với đa dạng các chủ đề nhưng phần lớn thuộc lĩnh vực tin học, máy tính. Quá trình thu thập dữ liệu được chúng tôi thực hiện với thư viện Beautiful Soup [2].

3.2 Thông tin bộ dữ liệu

Sau khi thu thập dữ liệu chúng tôi thu được 4 tập dữ liệu gồm: USER, POST, COMMENT and not_available_user. Thông tin chi tiết của các bộ dữ liệu được thể hiện bên dưới.

- POST: chứa dữ liệu của các bài viết được đăng lên diễn đàn.

- Loại file: Comma-separated values (csv).
- Số điểm dữ liệu: 18,643.
- số thuộc tính: 10.
- Codebook:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Miền giá trị
1	id_post	Mã định danh của bài viết	int64	[0, 21817]
2	title_post	Tiêu đề bài viết	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
3	url_post	Đường dẫn đến bài viết	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
4	views	Số lượt xem của bài viết	int64	[51, 351000]
5	comments	Số lượt bình luận của bài viết	int64	[1, 1000]
6	content_post	Nội dung bài viết	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
7	tags	Các nhãn được người viết gán cho mỗi bài viết	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
8	id_author_post	Mã định danh của tác giả bài viết	int64	[1, 468303]
9	url_author_post	Đường dẫn đến trang cá nhân của tác giả bài viết	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
10	datePublished_post	Thời gian bài viết được đăng lên diễn đàn	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.

– COMMENT: chứa dữ liệu các bình luận tương ứng với các bài viết trong POST.
  - Loại file: Comma-separated values (csv).
  - Số điểm dữ liệu: 114,226.
  - số thuộc tính: 5.
  - Codebook:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Miền giá trị
1	id_post	Mã định danh của bài viết được bình luận	int64	[0, 21817]
2	id_author_cmt	Mã định danh của người bình luận	int64	[1, 468320]
3	url_author_cmt	Đường dẫn đến trang cá nhân của người bình luận	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
4	content_cmt	Nội dung bình luận	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
5	datePublished_cmt	Thời gian bình luận được đăng tải	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.

– USER: chứa dữ liệu về những người dùng đã xuất hiện trong POST và COMMENT.

- Loại file: Comma-separated values (csv).
- Số điểm dữ liệu: 10,997.
- số thuộc tính: 7.
- Codebook:

STT	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu	Miền giá trị
1	id_user	Mã định danh của người dùng	int64	[1, 468320]
2	name_user	Tên người dùng	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
3	DateJoin	Ngày tham gia vào hệ thống của người dùng	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
4	DateLastAccess	Thời gian truy cập lần cuối của người dùng	object	Các ký tự chữ viết (Anh, Việt), số và ký tự đặc biệt.
5	comment_count	Số lượng bình luận người dùng đã bình luận trên diễn đàn	int64	[0, 26715]
6	Reaction_Score	Điểm tương tác của người dùng	int64	[0, 102]
7	Score	Điểm danh hiệu của người dùng	int64	[0, 1132]

– not_available_user: danh sách những người dùng hiện không còn tồn tại trên diễn đàn. Dữ liệu này được lưu trữ lại trong quá trình thu thập nhằm phục vụ cho quá trình làm sạch dữ liệu.
  - Loại file: Text (txt).
  - Số điểm dữ liệu: 2,390.
  - số thuộc tính: 1.

Xem thêm về dữ liệu tại đây.

4 Phương Pháp

Để giải quyết các bài toán đặt ra, chúng tôi tiếp cận theo 2 hướng chính: Content-based Filtering và Collaborative Filtering kết hợp với Neural Network (Model-based). Mỗi phương pháp chúng tôi sẽ tiến hành các bước tiền xử lý và xây dựng hệ khuyến nghị phù hợp. Cụ thể như sau:

4.1 Content-based Filtering

Đề xuất được đưa ra dựa vào profile của người dùng hoặc dựa vào nội dung/thuộc tính của những item tương tự như item mà người dùng đã chọn trong quá khứ.

4.1.1 Tiền xử lý dữ liệu là một bước quan trọng trong quá trình xây dựng, huấn luyện mô hình. Với tiền xử lý, bộ dữ liệu sẽ được làm sạch, phát hiện ra được những thuộc tính mới có ích và có khả năng trích xuất được nhiều thông tin hơn cho việc huấn luyện các mô hình, góp phần cải thiện các kết quả đạt được.

– Feature Generations: thuộc tính post_final tổng hợp title, content, tags của từng bài viết.
– Chuyển về chữ thường.
– Chuẩn hóa Unicode, chuẩn hóa kiểu gõ.
– Xóa bỏ các kí tự đặc biệt, teencode (vd: dấu @, dấu #,...).
– Tách từ: sử dụng Python Vietnamese Toolkit [1]
– Loại bỏ stopword.
– Chuyển đổi text thành vector thông qua Word Embedding.

4.1.2 Word embedding là một kỹ thuật trong Xử lý ngôn ngữ tự nhiên (NLP), bằng cách ánh xạ các từ hoặc cụm từ từ nhóm từ vựng thành các vector số thực. Nó giúp cải thiện độ chính xác của các mô hình ngôn ngữ tự nhiên khác nhau. Có nhiều kỹ thuật embedding khác nhau, ở đồ án này chúng tôi sử dụng 3 kỹ thuật lần lượt là TF-IDF, Word2Vec và PhoBERT. Trong đó, TF-IDF được triển khai dựa trên TfidfVectorizer được tích hợp sẵn trong sklearn, PhoBERT chúng tôi sử dụng pre-trained model vinai/phobert-base [2]. Riêng Word2Vec chúng tôi tiến hành train lại model dựa trên tập từ vựng của bộ dữ liệu chúng tôi thu thập được.

4.1.3 Clustering trước khi đưa các vector embedding vào xây dựng hệ khuyến nghị, chúng tôi tiến hành phân loại các vector ấy thành các cụm khác nhau, kết hợp với one-hot encoding và nối các vector phân loại vào đuôi của từng vector embedding. Quá trình được mô tả ở hình 1.

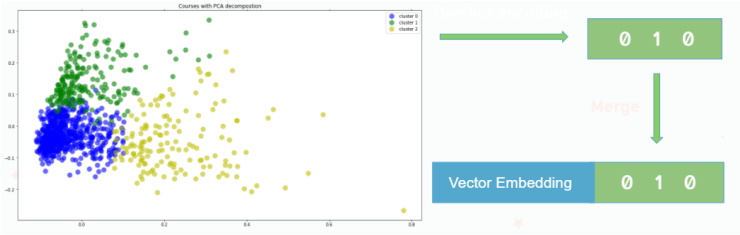

Fig. 1: Phương pháp Clustering

4.1.4 Topic modeling - LDA Topic modeling là một phương pháp được áp dụng khá nhiều trong việc xây dựng các hệ thống khuyến nghị dựa trên Content-based. Trong đó, LDA (Latent Dirichlet allcation) [3] được biết đến như một thuật ngữ phổ biến của lĩnh vựa này. LDA là một mô hình sinh xác suất nhằm xác định tập hợp các chủ đề tưởng tượng dựa trên các phân phối văn bản và phân phối từ của một tập ngữ liệu. Với LDA, mỗi văn bản được xem như sự

[1] Python Vietnamese. Toolkit-`https://pypi.org/project/pyvi/`
[2] VinAI PhoBERT-`https://github.com/VinAIResearch/PhoBERT`

pha trộn của nhiều chủ đề khác nhau và mỗi chủ đề lại được biểu diễn bằng một tập hợp các từ với mức độ đóng góp khác nhau. Việc áp một mô hình LDA có k chủ đề lên một văn bản sẽ giúp biểu diễn văn bản đó dưới dạng một vector có k chiều từ đó việc tính toán trở nên đơn giản và nhanh chóng hơn.
Để xây dựng mô hình LDA cần chú ý một số tham số sau:

- num_topics: số lượng chủ đề.
- alpha, eta: tham số đại diện cho hệ số mật độ giữa văn bản-chủ đề và chủ đề-từ, có thể điều chỉnh thủ công hoặc sử dụng 'auto'.
- Các tham số khác như passes, iterations, chunksize,...

Bên cạnh đó, để đánh giá hiệu suất và tìm ra mô hình LDA tối ưu nhất người ta sử dụng một độ đo được gọi là Topic Coherence. Đây là một thước đo giúp tính toán độ tương đồng về ngữ nghĩa giữa các từ trong một chủ đề và được hỗ trợ bởi thư viện Gensim. Chỉ số Coherence càng cao thì mô hình chủ đề xây dựng được càng tốt và ngược lại.

4.1.5 Tính toán tương đồng Để xây dựng hệ khuyến nghị dựa trên phương pháp Content-based, chúng ta cần xác định độ tương đồng giữa profile người dùng với nội dung của từng bài đăng có trong diễn đàn mà người dùng đó chưa tương tác. Tại đây, chúng tôi tổ chức xây dựng mỗi profile người dùng dưới dạng một danh sách các bài viết mà người dùng đã tương tác, trong đó mỗi bài đăng được tiếp cận theo hai hướng: tổng hợp tất cả thông tin của 3 thuộc tính title, content, tags và chỉ tổng hợp với 2 thuộc tính title và tags. Độ tương đồng của mỗi profile với mỗi bài viết sẽ được xác định bởi độ tương đồng trung bình của bài viết đó với từng viết có trong profile. Việc tính toán được thực hiện dựa trên độ đo Cosine, một thước đo giúp đánh giá độ tương tự giữa 2 vector, có công thức như sau:

$$similarity = \cos(\theta) = \frac{A \cdot B}{||A||\ ||B||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{1}$$

4.2 Collaborative Filtering + Neural Network (NCF)

4.2.1 Tiền xử lý dữ liệu dữ liệu được xây dựng cho phù hợp với đầu vào của mô hình, bao gồm 4 thuộc tính: userID là id ủa người dùng, itemID là id của bài post, timestamp thể hiện thời gian người dùng tương tác với bài post, label với 2 giá trị 0, 1 (0: chưa tương tác, 1: đã tương tác)

4.2.2 Mô hình NCF [3] đề xuất thông tin cho người dùng dựa trên việc người dùng có tương tác với bài post cụ thể hay không. Kiến trúc mô hình chủ yếu xây dựng trên các lớp Embedding và Fully Connected để dự đoán xác xuất người dùng tương tác với từng bài post trong hệ thống.

[3] Neural Collaborative Filtering-`https://arxiv.org/abs/1708.05031`

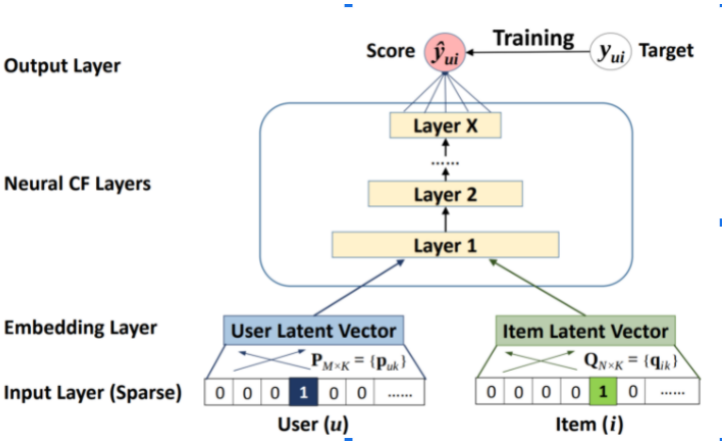

Fig. 2: Kiến trúc mô hình NCF

NCF được triển khai với các tham số cần chú ý như sau:

- n_factors: kích thước của không gian tiềm ẩn (latent space)
- layer_sizes: kích thước của lớp đầu vào
- n_epochs: xác định số lần lặp của SGD
- model_type: "MLP", "GMF" đơn lẻ hoặc mô hình kết hợp "NeuCF"

4.3 Đánh giá

Để đánh giá hiệu quả của các phương pháp khuyến nghị, chúng tôi sử dụng phương pháp "leave-one-out" được đề cập trong bài báo gốc của NCF. Đối với mỗi user, chúng tôi lấy 1 post mà user tương tác gần nhất và 100 bài post ngẫu nhiên user chưa từng tương tác trước đó. Ở đồ án này chúng tôi tiến hành đánh giá dựa trên 500 user, tương ứng với 50,500 cặp user-post và khuyến nghị top 10 bài post tương đồng với user nhất. Hiệu suất của một danh sách khuyến nghị cho từng user được đánh giá bằng Hit ratio (HR) và Normalized Discounted Cumulative Gain (NDCG). Hit ratio có thể hiểu là tỉ lệ xuất hiện của bài post mà user tương tác trong top bài post mà hệ thống khuyến nghị mà không quan tâm đến xếp hạng, NDCG tương tự nhưng xếp hạng cao hơn sẽ được đánh giá cao hơn, công thức của NDCG được thể hiện ở hình 3 . Cuối cùng, tính trung bình các giá trị trên 500 user để có được HR và NDCG tổng thể trên dữ liệu thử nghiệm. Chi tiết công thức tính toán NDCG được thể diện bên dưới.

$$\mathrm{DCG}_v = \sum_{i=1}^{v} \frac{g(rel_i)}{log(i+1)}$$
$$\mathrm{IDCG}_v = \sum_{k \in \mathrm{REL}_v} \frac{g(rel_k)}{log(k+1)}$$
$$\mathrm{nDCG}_v = \frac{\mathrm{DCG}_v}{\mathrm{IDCG}_v}$$

Fig. 3: Normalized Discounted Cumulative Gain (NDCG)

5 Thực nghiệm và Kết quả

5.1 Các thông số thực nghiệm

- TF-IDF: được cài đặt với TfidfVectorizer(max_features=15000), huấn luyện trên dữ liệu tổng hợp title, content, tag của các bài post.
- Word2Vec: thực nghiêm trên 2 mô hình được huấn luyện với 2 bộ tham số lần lượt là vector_size=200, window=2, epoch=50 và vector_size=300, window=2, epoch=30.
- Clustering: sử dụng Kmean để phân cụm các bài post.

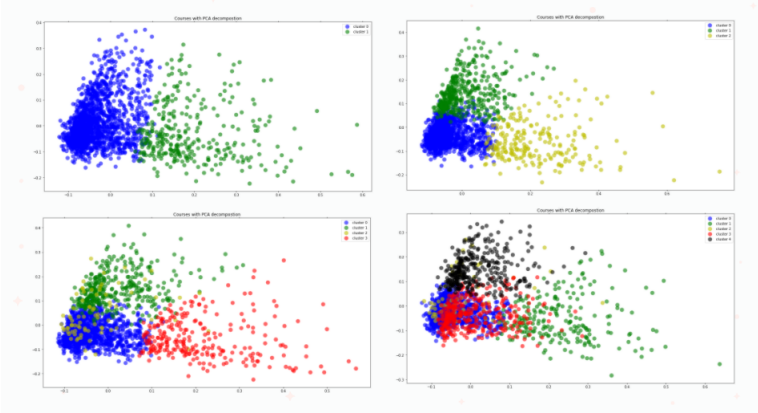

Fig. 4: Kết quả phân cụm với k lần lượt là 2, 3, 4, 5.

Kết quả phân cụm cho thấy k=2 và k=3 cho ra các nhóm bài post phân biệt rõ ràng, không bị pha trộn giữa các nhóm. Sau khi kết hợp clustering với embedding TF-IDF, k=3 cho kết quả khuyến nghị tốt hơn k=2 nên chúng tôi quyết định cố định giá trị phân cụm k=3 để thực nghiệm với các kỹ thuật embedding khác.

– Với LDA, chúng tôi thực hiện xây dựng mô hình dựa trên nội dung mỗi bài viết được tổng hợp bởi title và tags. Bên cạnh đó, mô hình được huấn luận trên dữ liệu đã được embedding bởi phương pháp TF-IDF với max_features là 1762. Sau quá trình tinh chỉnh các siêu tham số, mô hình LDA tốt nhất mà chúng tôi thu được đạt chỉ số Cohenrence là 0.41 với num_topics=13, passes=40, iterations=200, chunksize=10000, điều này được thể hiện chi tiết tại tại ảnh 5.

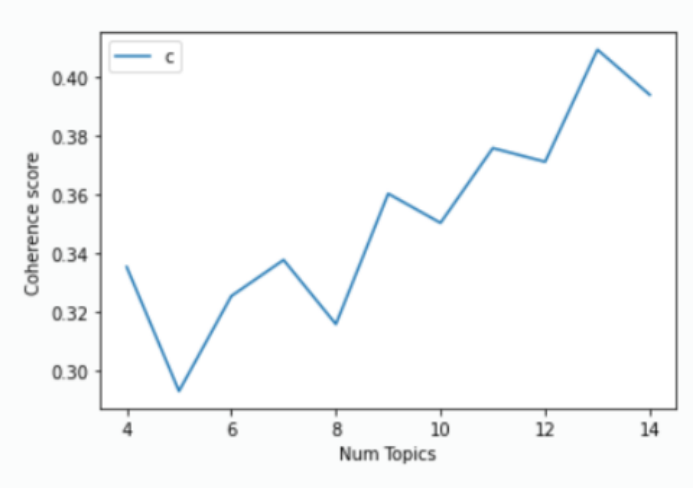

Fig. 5: .Kết quả thực nghiệm với các số lượng chủ đề khác nhau.

Xem thêm các từ ngữ cùng mức độ đóng góp của chúng trong mỗi chủ đề của mô hình LDA thu được tại ảnh 6.

```
[(0,
  '0.058*"phần_mềm" + 0.050*"video" + 0.036*"key" + 0.033*"youtube" + 0.031*"gỡ" + 0.030*"idm" + 0.022*"help" + 0.022*"me" + 0.018*"nhạc" + 0.016*"hoạt_động"'),
 (1,
  '0.029*"google" + 0.028*"chrome" + 0.028*"chữ" + 0.024*"giúp_đỡ" + 0.024*"trình_duyệt" + 0.024*"font" + 0.022*"link" + 0.021*"tải" + 0.019*"lỗi" + 0.018*"photoshop"'),
 (2,
  '0.027*"gta" + 0.017*"tin_học" + 0.016*"v" + 0.016*"tại_sao" + 0.015*"chip" + 0.014*"exe" + 0.012*"thông_tin" + 0.012*"lỗi" + 0.011*"flash" + 0.011*"hệ_thống"'),
 (3,
  '0.029*"office" + 0.029*"ổ_cứng" + 0.025*"ram" + 0.024*"lỗi" + 0.022*"nâng_cấp" + 0.022*"ổ" + 0.021*"thắc_mắc" + 0.021*"file" + 0.017*"dữ_liệu" + 0.017*"xóa"'),
 (4,
  '0.035*"xanh" + 0.031*"cài_đặt" + 0.027*"ứng_dụng" + 0.022*"phát" + 0.022*"iphone" + 0.022*"màn_hình" + 0.020*"lỗi" + 0.019*"main" + 0.019*"pc" + 0.019*"chặn"'),
 (5,
  '0.068*"tư_vấn" + 0.053*"mua" + 0.047*"laptop" + 0.024*"asus" + 0.022*"dell" + 0.021*"bàn_phím" + 0.018*"active" + 0.016*"pass" + 0.015*"mấy" + 0.015*"máy_tính"'),
 (6,
  '0.064*"khởi_động" + 0.040*"virus" + 0.027*"đen" + 0.024*"máy_tính" + 0.023*"diệt" + 0.023*"màn_hình" + 0.018*"phần_mềm" + 0.018*"chương_trình" + 0.017*"triệu" + 0.016*"hỏi_đáp"'),
 (7,
  '0.071*"usb" + 0.038*"boot" + 0.024*"đĩa" + 0.021*"lỗi" + 0.020*"win" + 0.019*"pin" + 0.018*"sửa" + 0.017*"laptop" + 0.017*"cài" + 0.016*"ổ_đĩa"'),
 (8,
  '0.031*"cpu" + 0.022*"net" + 0.020*"window" + 0.019*"liên_minh" + 0.019*"pascal" + 0.018*"tốc_độ" + 0.016*"kiểm_tra" + 0.014*"anh_em" + 0.014*"viết" + 0.014*"cũ"'),
 (9,
  '0.043*"máy_tính" + 0.040*"tắt" + 0.029*"tự_động" + 0.026*"máy" + 0.022*"bản_quyền" + 0.021*"bật" + 0.019*"online" + 0.016*"lỗi" + 0.015*"laptop" + 0.014*"biểu_tượng"'),
 (10,
  '0.065*"game" + 0.059*"màn_hình" + 0.050*"card" + 0.035*"cấu_hình" + 0.019*"trợ_giúp" + 0.018*"vga" + 0.018*"máy_tính" + 0.017*"icon" + 0.017*"chạy" + 0.016*"lỗi"'),
 (11,
  '0.060*"wifi" + 0.049*"giúp" + 0.041*"mạng" + 0.028*"kết_nối" + 0.023*"lỗi" + 0.020*"đổi" + 0.019*"bắt" + 0.018*"việt" + 0.018*"mật_khẩu" + 0.016*"trang"'),
 (12,
  '0.088*"win" + 0.063*"cài" + 0.041*"lỗi" + 0.035*"driver" + 0.034*"ghost" + 0.029*"xp" + 0.026*"facebook" + 0.024*"giúp" + 0.024*"windows" + 0.022*"update"')]
```

Fig. 6: Ảnh thể hiện phân phối các từ trong mỗi chủ đề của mô hình

– Cuối cùng, mô hình NCF chúng tôi thực nghiệm với 2 loại là "GMF" và "NeuCF", đều huấn luyện với batch_size=128 và epochs=50.

5.2 Kết quả

Sau khi thực hiện nhiều thực nghiệm với các sự kết hợp khác nhau giữa những phương pháp Embedding và các hướng tiếp cận tổng hợp thông tin profile đa dạng cũng như áp dụng nhiều loại mô hình với NFC, chúng tôi thu được những hệ khuyến nghị dựa trên Content-based và Collaborative với hiệu suất được thể hiện tại bảng 1.

RS					**Hit ratio**	**NDCG**
			Aggregate Text	**Aggregate Profile**		
CB	TF-IDF		Title + tags	post	0.319	0.262
				cmt	0.215	0.165
				post + cmt	0.225	0.174
			Full	post	0.319	0.277
				cmt	0.243	0.206
				post + cmt	0.261	0.212
		cluster		cmt	**0.336**	**0.271**
		LDA	Title + tags	post + cmt	0.162	0.110
	W2V	D200 + cluster	Full	cmt	**0.334**	**0.280**
		D300 + cluster			0.312	0.269
	PhoBert		Full	cmt	0.216	0.152
CF	NCF	GMF		cmt	0.266	0.250
		NeuMF			0.250	0.210

Table 1: Bảng kết quả thực nghiệm thu được.

▷ Nhận xét:

– Kết quả thu được dao động từ 0.2 đến 0.35 với hit ratio và 0.15 đến 0.28 với NDCG. Hiệu suất thu được có thể được xem là khá thấp so với thang điểm trên 1, tuy nhiên với phương pháp đánh giá "leave-one-out" được áp dụng thì hiệu suất như trên đã được xem là tạm ổn.
– Nhìn vào 6 kết quả đầu của bảng với phương pháp TF-IDF, chúng tôi nhận thấy khi thay đổi cách thức tổng hợp nội dung profile mang lại những kết quả khác nhau. Cụ thể, việc chỉ tổng hợp profile trên các bài viết mà người dùng đã đăng mang lại hiệu suất cao hơn so với chỉ tổng trên các bài viết người dùng bình luận và tổng hợp trên cả hai.
– Quan sát kết quả thu được dựa trên collaborative filtering có thể thấy độ chênh lệch giữa các thực nghiệm không chênh lệch quá nhiều.
– Bên cạnh đó, Content-based mang lại những hệ khuyến nghị với hiệu suất vượt trội hơn hẳn khi áp dụng cùng phương pháp Clustering, cụ thể là hai thực nghiệm TF-IDF và Word2Vec 200 chiều kết hợp cùng Clustering và đây

cũng là hai hệ khuyến nghị mang lại kết quả tốt nhất trên toàn quá trình thực nghiệm.
– Có thể thấy TF-IDF kết hợp với LDA mang lại kết quả tệ nhất trong các thực nghiệm. Điều này có thể đượclý giải bởi chất lượng mô hình xác định topic văn bản chưa thật sự tốt.

6 Kết luận và Hướng phát triển

6.1 Kết luận chung

– Với nhưng kết quả thực nghiệm hiện tại, chúng tôi nhận thấy phương pháp Content-based kết hợp với Clustering là phương pháp tiềm năng trên bộ dữ liệu của chúng tôi.
– Việc thay đổi cách thức tổng hợp nội dung cho trên từng bài viết không mang lại thay đổi lớn. Tuy nhiên, với các phương pháp tổng hợp profile khác nhau lại mang lại kết quả khác nhau. Chính vì vậy, nên tập trung hơn vào phát tiển các hệ thống khuyến nghị chỉ dựa trên nội dung mà người dùng đã đăng.

6.2 Hướng phát triển

– Tiếp tục tìm hiểu nhiều hơn để có thể khai thác hết tìm năng từ toàn bộ dữ liệu được cung câp nhằm xây dựng một hệ khuyến nghị hoàn hảo hơn.
– Tìm hiểu thêm nhiều phương pháp Word Embedding khác và áp dụng vào bài toán.
– Áp dụng triển khai các phương pháp học sâu.
– Tìm hiểu các hướng giải quyết mới cho bài toán.

7 Demo và Source code

7.1 Demo

Thực hiện xây dựng chương trình demo đơn giản với Flask [1]. Hệ khuyến nghị được sử dụng là hệ khuyến nghị dựa trên content-based với TF-IDF được áp dụng dựa trên thông tin được tổng hợp trên mỗi bài viết bao gồm title, content và tags. Chi tiết đầu vào và đầu ra của hệ thống khuyến nghị được thể hiện ở ảnh 7.

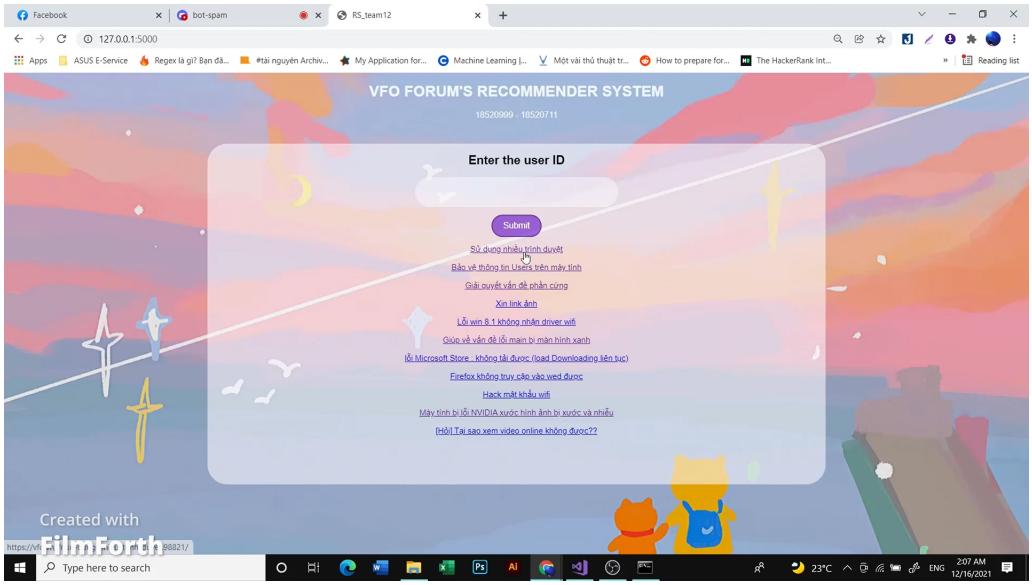

Fig. 7: Chi tiết giao diện, đầu vào và đầu ra của web app được tạo bởi Flask.

Xem thêm video demo tại đây.

7.2 Source code

Xem chi tiết tại đây.

References

1. Miguel Grinberg. *Flask web development: developing web applications with python.* " O'Reilly Media, Inc.", 2018.
2. Leonard Richardson. Beautiful soup documentation. *April*, 2007.
3. David M. Blei và Andrew Y. Ng và Michael I. Jordan và John Lafferty. Latent dirichlet delivery. *Journal of Machine Learning Research*, 3:2003, 2003.