

XÂY DỰNG HỆ THỐNG KHUYẾN NGHỊ KHÁCH SẠN DỰA TRÊN BÌNH LUẬN THỜI GIAN THỰC

Nguyễn Hoàng Minh^{1,2,3}, Nguyễn Thiện Thuật^{1,2,3}, Tạ Nhật Minh^{1,2,3}, and Đỗ Trọng Hợp^{1,2,4}

¹ University of Information Technology, Ho Chi Minh, Vietnam

² Vietnam National University, Ho Chi Minh City, Vietnam

³ {20521609,20521998,20521614}@gm.uit.edu.vn

⁴ hopdt@gm.uit.edu.vn

Tóm tắt nội dung Du lịch là nhu cầu cơ bản trong cuộc sống con người. Bên cạnh đó, chỗ ở cũng là một yếu tố quan trọng, đặc biệt là khách sạn giúp cải thiện trải nghiệm du lịch. Trong nghiên cứu này, chúng tôi đã giới thiệu một giải pháp cho việc xây dựng một mô hình gợi ý sử dụng dữ liệu tiếng Việt và dữ liệu người dùng nhằm đề xuất cho du khách chọn khách sạn phù hợp. Dữ liệu được thu thập từ trang web du lịch nổi tiếng Ivivu, bao gồm thông tin về khách sạn tại Việt Nam và lịch sử phản hồi của người dùng. Chúng tôi đã tiền xử lý và gán nhãn theo các chủ đề khác nhau, bao gồm dịch vụ, cơ sở hạ tầng, vệ sinh, vị trí và thái độ. Mô hình gợi ý của chúng tôi được xây dựng bằng cách sử dụng kỹ thuật lọc cộng tác và học sâu, chúng tôi đã đề xuất kết hợp các vector ngữ cảnh từ các bình luận tiếng Việt của du khách trong quá trình đề xuất. Mô hình ngữ cảnh được phát triển bằng cách sử dụng kỹ thuật học sâu để trích xuất chủ đề và cảm xúc từ các từ một cách hiệu quả. Kết quả của mô hình được đề xuất của chúng tôi, được đo bằng MSE, RMSE, MAE là tốt hơn rất nhiều so với mô hình không có ngữ cảnh sử dụng cùng các thông số. Bên cạnh đó chúng tôi đã nghiên cứu hệ thống thu thập dữ liệu trực tuyến vào Cassandra và xử lý trực tuyến đưa vào mô hình theo thời gian thực bằng Kafka và Spark Streaming. Nghiên cứu của chúng tôi cho thấy phương pháp của chúng tôi cải thiện độ chính xác của mô hình gợi ý và có tiềm năng cho sự phát triển thêm trong tương lai. Ý tưởng này có thể giới thiệu một Hệ thống Gợi ý mới có thể vượt qua những hạn chế hiện tại và áp dụng vào các lĩnh vực khác.

Keywords: Review-based recommendation system (RRS), Vietnamese review, deep learning, Kafka, Spark Streaming, Cassandra.

1 Giới thiệu

Hệ khuyến nghị khách sạn thời gian thực là một hệ thống đề xuất các khách sạn phù hợp với người dùng từ lịch sử mới nhất của họ thông qua các công nghệ xử

lý dữ liệu thời gian thực. Hệ thống đề xuất khách sạn giúp người dùng tiết kiệm thời gian và công sức trong việc tìm kiếm khách sạn phù hợp với nhu cầu của họ qua việc dự đoán sở thích cá nhân và đề xuất ra một danh sách các phương án phù hợp cho người dùng.

Một trong những lợi ích quan trọng của hệ thống đề xuất khách sạn là mang lại sự thuận tiện và linh hoạt cho người dùng bằng cách lựa chọn các tiêu chí và yêu cầu của mình, người dùng sẽ nhận được gợi ý về những khách sạn phù hợp nhất với nhu cầu của họ. Điều này giúp tiết kiệm thời gian và công sức khi xem xét hàng trăm khách sạn khác nhau trên trang web. Với hệ thống đề xuất khách sạn thời gian thực, nó có thể giúp tự động hóa và liên tục cập nhật kịp thời dữ liệu lịch sử của người dùng để đưa ra các đề xuất phù hợp nhất mà không mất quá nhiều công sức của con người. Đối với khách sạn, hệ thống này giúp họ tăng doanh số bán hàng bằng cách tiếp cận đúng khách hàng với đúng sản phẩm và dịch vụ.

Trong đề tài này chúng tôi thực hiện xây dựng một hệ thống đề xuất khách sạn có khả năng liên tục thu thập dữ liệu từ lịch sử khách hàng trên web và có thể đề xuất thời gian thực các khách sạn phù hợp với người dùng từ những lịch sử mới nhất của họ. Hệ thống của chúng tôi giúp cập nhật mới nhất lịch sử người dùng và đưa ra các đề xuất khách sạn thời gian thực, giúp cho người dùng có những đề xuất sớm nhất khi họ mới du lịch ở tại địa điểm mới và tránh tình trạng hệ thống phải đề xuất quá nhiều người dùng tại một thời điểm. Với hệ thống này, các doanh nghiệp du lịch có thể sử dụng để triển khai trực tiếp trên trang web của họ nhằm cải thiện trải nghiệm người dùng.

2 Nghiên cứu liên quan

Trong hệ thống đề xuất khách sạn, các nghiên cứu trước cũng đã có ý tưởng sử dụng Học cộng tác để xây dựng mô hình. Như trong nghiên cứu của Shambour et. al.[1] đã có đề xuất kết hợp dựa trên cách tiếp cận enhanced user-based CF và enhanced item-based CF có tên là a fusion-based multi-criteria CF (FBMCCF), mô hình được đánh giá trên tập dữ liệu TripAdvisor MC dataset [2] để so sánh với các mô hình Multi-Criteria Collaborative Filtering khác, với độ đo bằng MAE và RMSE cho kết quả vượt trội hơn các mô hình trước đó. Ở cách tiếp cận khác, Kaya, Buket [3] phát triển một hệ thống đề xuất khách sạn mới dựa trên link prediction bằng cách sử dụng the customer-hotel bipartite network.

Trong lĩnh vực rộng hơn liên quan đến du lịch, Các tác giả Al-Ghobari et. al.[4] có ý tưởng sử dụng a location-aware traveler assistance (LAPTA) làm ngữ cảnh để đề xuất thông qua mô hình KNN Item-Based Collaborative Filtering.

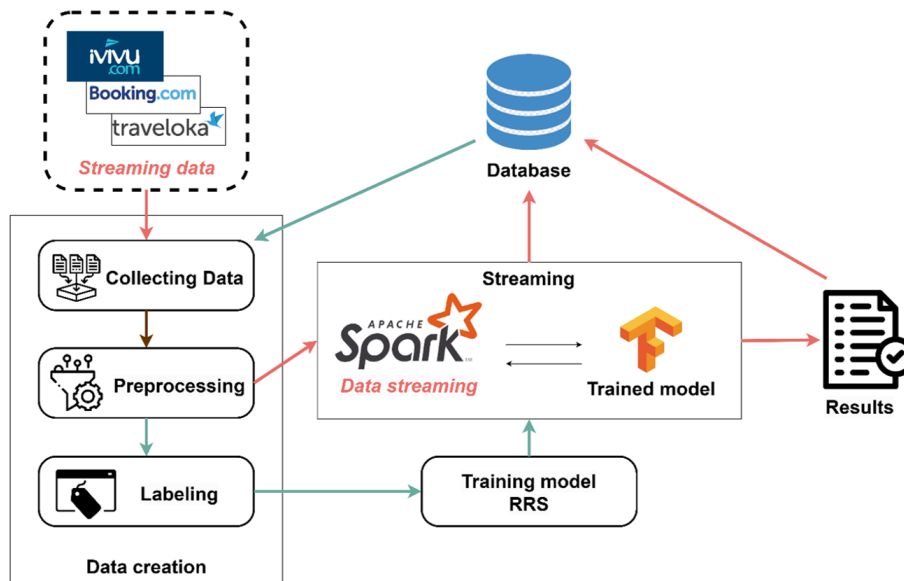
Abbasi et. al. [5] cũng có ý tưởng dùng kết hợp sentiment với Học cộng tác (CF) dựa trên Deep learning. Họ đã sử dụng kết hợp grouping recommender system và Deep learning để xây dựng hệ thống đề xuất. Thay vì đề xuất cho từng người dùng thì họ lại đề xuất cho một nhóm người dùng có chung sở thích [6] để giải quyết vấn đề ma trận thưa. Họ sử dụng Dense embedding riêng biệt cho từng user và item và sử dụng SVD để xây dựng ma trận item-user cho mô hình đề xuất. Nhưng với cách tiếp cận này, kết quả sentiment của bình luận được sử

dụng có vài trò giống như rating để thực hiện dự đoán. Nó sẽ không giải quyết được mức độ hài lòng hiện tại của người dùng như thế nào và họ đang quan tâm tới vấn đề gì từ khách sạn mà họ đã trải nghiệm.

Đối với bài toán về đề xuất khách sạn, chúng tôi cho rằng các yếu tố về lịch sử của người dùng, đặc biệt là bình luận về khách sạn họ đã đi sẽ ảnh hưởng đến các khách sạn ở chuyến đi tiếp theo. Bình luận thể hiện được những vấn đề mà hiện tại người dùng đang quan tâm và cảm xúc về chuyến du lịch gần nhất của họ. Chính vì vậy, chúng tôi đề xuất mô hình đề xuất có sự kết hợp với bình luận đánh giá như một ngữ cảnh để hỗ trợ dự đoán hiệu quả hơn.

3 Tổng quan hệ thống khuyến nghị khách sạn dựa trên bình luận thời gian thực

Trong phần này chúng tôi trình bày tổng quan hệ thống đề xuất khách sạn và định hướng triển khai dự đoán thời gian thực. Chúng tôi tập trung vào việc xây dựng mô hình khuyến nghị khách sạn dựa trên bình luận có tên là Hotel Review-based Recommendation System (RRS) và sau đó đề xuất một hệ thống để triển khai thực tế cho mô hình của chúng tôi. Hình 1 là tổng quan hệ thống bao gồm 3 phần chính: Thu thập và xây dựng bộ dữ liệu huấn luyện, Phương pháp cốt lõi của RRS, và Hệ thống đề xuất khách sạn thời gian thực.



Hình 1. Phương pháp tiếp cận của chúng tôi cho bài toán xây dựng hệ thống đề xuất khách sạn

Phần 1 – Thu thập và xây dựng bộ dữ liệu huấn luyện:

Ở phần này sẽ tập trung vào 2 nhiệm vụ: xây dựng bộ dữ liệu huấn luyện mô hình và tiền xử lý dữ liệu cho việc dự đoán streaming từ 2 nguồn là website hoặc database.

Phần 2 – Xây dựng và huấn luyện RRS:

Ở phần này sẽ thực hiện xây dựng mô hình dự đoán khách sạn dựa trên bình luận; tiến hành thực nghiệm và đánh giá mô hình.

Phần 3 – Xây dựng hệ thống RRS cho việc dự đoán dữ liệu thời gian thực:

Ở phần này sẽ triển khai mô hình đã được huấn luyện ở phần 2 cho việc đưa ra các đề xuất khách sạn cho người dùng đối với dữ liệu thời gian thực bằng các công nghệ như Kafka, Spark.

4 Hệ thống khuyến nghị khách sạn dựa trên bình luận

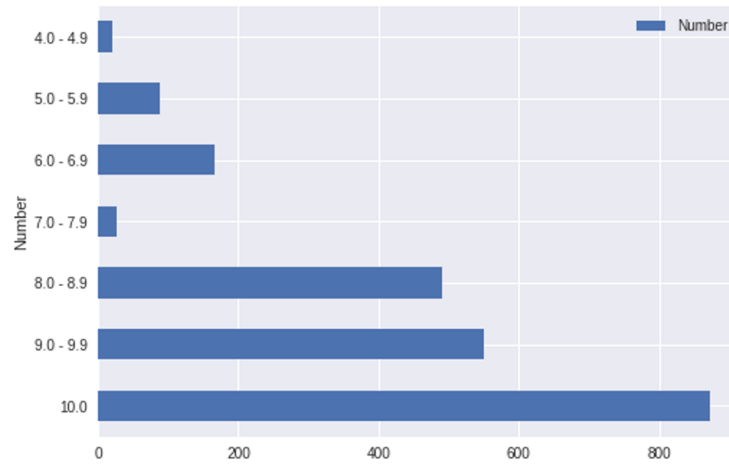
4.1 Thống kê bộ dữ liệu

Bộ dữ liệu gồm 2231 dòng dữ liệu là bình luận đánh giá khách sạn của người dùng đã được gán nhãn tương tự Hình 1, với thông tin của 368 user và 137 hotel khác nhau với mức Rating từ 4.5 – 10 được thống kê tại Bảng 2. Các bình luận liên quan đến các chủ đề về Service (dịch vụ), Infrastructure (cơ sở vật chất), Sanitary (vệ sinh), Location (vị trí), hoặc không xác định (được gán giá trị 0 tại cả 4 chủ đề). Thái độ (Attitude) của người dùng được đánh giá qua 3 mức độ là hài lòng (gán giá trị 2), bình thường (gán giá trị 1) và không hài lòng (gán giá trị 0). Thống kê số lượng các nhãn được thể hiện ở Bảng 2.

Bảng 1. Mẫu dữ liệu trích từ bộ dữ liệu hoàn chỉnh

User	Location	Hotel	Rating	Comment	Ser ¹	Inf ²	San ³	Loc ⁴	Att ⁵
Nguyen V. C. Phan	Thiêt	Khu nghỉ	10.0	gia_đình đi	0	0	0	0	2
		dưỡng		ok chuyển đi					
		Pandanus		thành_công					
		Phan Thiêt		vui_vẻ					
Phan T. T.	Nha Trang	Khách sạn	9.7	gia_đình đi	0	0	0	0	2
				ok chuyển đi					
				thành_công					
				vui_vẻ					

Note: ¹Service, ²Infrastructure, ³Sanitary, ⁴Location, ⁵Attitude



Hình 2. Biểu đồ đánh giá chất lượng khách sạn

Bảng 2. Thống kê số lượng nhân theo chủ đề và độ hài lòng

Chủ đề	Hài lòng	Bình thường	Không hài lòng
Dịch vụ	1237	199	191
Cơ sở vật chất	952	200	192
Vệ sinh	463	75	108
Vị trí	348	82	35
Không xác định	204	31	6

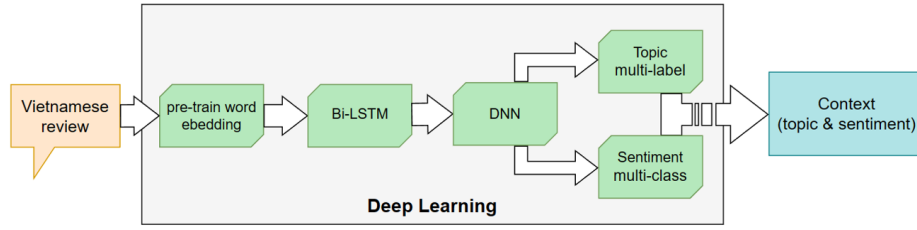
4.2 Tổng quan kiến trúc RRS

Trong bài báo này, chúng tôi đề xuất một cách tiếp cận mới dựa trên Học cộng tác và Deep learning. Đồng thời, các comment phản hồi của người dùng cũng được dùng làm ngữ cảnh cho hệ thống (Hình 4)

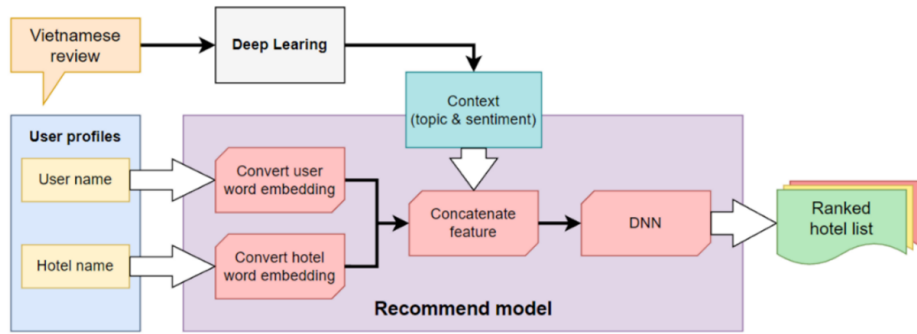
Hình 3 là mô hình xử lý ngôn ngữ để trích xuất các đặc trưng từ đánh giá của khách hàng. Kết quả dự đoán của mô hình này sẽ được sử dụng làm đầu vào như một ngữ cảnh cho hệ thống đề xuất.

Hệ thống khuyến nghị của chúng tôi dựa trên phương pháp học sâu và có quy trình xử lý gồm hai bước như sau:

- Mô hình xử lý ngôn ngữ sẽ học các bình luận của khách hàng để dự đoán các chủ đề sẽ xuất hiện trong câu và mức độ hài lòng để làm ngữ cảnh cho hệ thống gợi ý.
- Mô hình đề xuất sẽ dự đoán rating bằng cách học từ hồ sơ lịch sử của người dùng nhằm đem những khách hàng có sở thích và đặc điểm giống nhau đến gần nhau hơn trong không gian vectơ thông qua embedding (giống như cách tiếp cận của học cộng tác), với đầu vào là lịch sử của người dùng và các bình luận nhận xét khách sạn mới nhất của khách hàng.



Hình 3. Sơ đồ tổng thể của mô hình Deep Learning



Hình 4. Sơ đồ tổng thể của mô hình khuyến nghị khách sạn

Đầu ra của mô hình xử lý ngôn ngữ bao gồm các chủ đề được nhắc đến và cảm xúc tổng thể trong câu với mong muốn xác định được chủ đề khách hàng quan tâm và cảm xúc hiện tại của họ.

Đầu ra của hệ thống khuyến nghị là danh sách các khách sạn được sắp xếp theo thứ tự giảm dần dựa trên xếp hạng do mô hình đề xuất.

4.3 Thực nghiệm và kết quả

Cài đặt thử nghiệm

Dataset:

Chúng tôi thực hiện đánh giá phương pháp đề xuất trên bộ dữ liệu đã xây dựng trong nghiên cứu này được thu thập từ 2 website về du lịch: Ivivu.com và Traveloka.com. Thông tin của bộ dữ liệu đã được thống kê ở mục trên. Dữ liệu được chia thành các tập train và tập test với tỷ lệ 4:1.

Setting:

Với mô hình của chúng tôi, bộ hyper-parameter tốt nhất phù hợp để huấn luyện trên bộ dữ liệu như sau:

- Mô hình Deep learning: dropout=0.5, 0.4, 0.3; regularizers.l2=0.03, 0.01; learning_rate = 0.00007, 0.00005; optimizer = Adam; loss = CrossEntropy; epochs=150,400; batch_size = 16.
- Mô hình đề xuất khách sạn: learning_rate=0.00007; optimizer = Adam; epochs = 150; batch_size = 16; dropout=0.2; activation = 'relu'; loss = MeanSquaredError.

Các mô hình so sánh

User-Based Collaborative Filtering (UserKNN)[7] là một kỹ thuật được sử dụng để dự đoán các item mà người dùng có thể thích trên cơ sở xếp hạng cho item đó bởi những người dùng khác có cùng sở thích với người dùng mục tiêu. Để tính độ đo tương đồng chúng tôi sử dụng độ đo khoảng cách tương đồng Cosine và Pearson, Chúng tôi cài đặt cho K tương đồng trong thực nghiệm là 40 và 25.

Item-Based Collaborative Filtering (ItemKNN)[8] là kỹ thuật Lọc cộng tác giữa các item khớp từng mặt hàng đã mua và xếp hạng người dùng với các mặt hàng tương tự, sau đó kết hợp các mặt hàng tương tự đó vào danh sách đề xuất. Để tính độ đo tương đồng chúng tôi sử dụng độ đo khoảng cách tương đồng Cosine và Pearson, Chúng tôi cài đặt cho K tương đồng trong thực nghiệm là 30, 20.

Matrix Factorization (MF)[9] là một phương pháp Lọc cộng tác hoạt động bằng cách phân tách ma trận tương tác Item-User thành tích của hai ma trận hình chữ nhật có số chiều thấp hơn. Ma trận đầu tiên, được gọi là ma trận người dùng, đại diện cho người dùng về các đặc trưng tiềm ẩn của họ. Ma trận thứ hai, được gọi là ma trận vật phẩm, đại diện cho các vật phẩm theo các đặc điểm tiềm ẩn của chúng. Trong đề tài này, chúng tôi sử dụng embedding để phân tách ma trận.

Distributed Deep Learning (ALS)[9] là một thuật toán Lọc cộng tác sử dụng Deep learning để đề xuất các item cho người dùng. Nó hoạt động bằng cách phân tách ma trận tương tác giữa người dùng và mục thành tích của hai ma trận hình chữ nhật có số chiều thấp hơn, tương tự như Matrix Factorization. Tuy nhiên, ALS sử dụng phương pháp học sâu để tìm hiểu các tính năng tiềm ẩn của người dùng và vật phẩm, điều này có thể cải thiện độ chính xác của các đề xuất. ALS là một thuật toán phân tán, nó có thể được sử dụng để đào tạo trên các tập dữ liệu rất lớn. Trong đề tài này, chúng tôi cài đặt các siêu tham số: maxIter=15, regParam=0.01, alpha=0.01, rank=10 của mô hình cho thực nghiệm.

Singular value decomposition (SVD)[10] là một kỹ thuật phân tích ma trận được sử dụng trong lọc cộng tác để đề xuất các mục cho người dùng. Nó hoạt động bằng cách phân tách ma trận tương tác giữa người dùng và mục thành

ba ma trận U , Σ và V . Sau đó có thể dự đoán rating còn thiếu dựa trên ma trận được xây dựng lại. Nó thực hiện điều này bằng cách so sánh những người dùng đã xếp hạng các mặt hàng tương tự và bằng cách sử dụng các yếu tố tiềm ẩn được ghi lại trong ma trận U , Σ và V . Trong đề tài này, chúng tôi cài đặt mặc định phương pháp này để thực nghiệm.

Các độ đo đánh giá

MSE

MSE (Mean Squared Error) là một độ đo phổ biến được sử dụng để đánh giá độ sai lệch giữa dự đoán của một mô hình và giá trị thực tế được tính bằng cách tính trung bình của bình phương các sai số.

$$MSE = n^{-1} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

RMSE

RMSE (Root Mean Squared Error) là một độ đo thường được sử dụng để đánh giá độ sai lệch giữa dự đoán của một mô hình được tính bằng cách tính căn bậc hai của trung bình của bình phương các sai số.

$$RMSE = \sqrt{n^{-1} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

MAE

MAE (Mean Absolute Error) là một độ đo thường được sử dụng để đánh giá độ sai lệch giữa dự đoán của một mô hình và giá trị thực tế được tính bằng cách tính trung bình của giá trị tuyệt đối của các sai số.

$$MAE = n^{-1} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

Kết quả thực nghiệm chính

Dựa trên kết quả từ Bảng 3, ta có một số nhận xét sau:

- Trong các mô hình memory-based, UserKNN-cosine và ItemKNN-cosine có MSE, RMSE và MAE thấp hơn so với UserKNN-pearson và ItemKNN-pearson, cho thấy tính tương quan cosine hiệu quả hơn tính tương quan Pearson trên bộ dữ liệu này.
- Trong các mô hình model-based, Distributed Deep Learning (ALS) có MSE, RMSE và MAE thấp nhất trong số các mô hình khác, cho thấy hiệu suất tốt nhất trong việc dự đoán.

- Các mô hình dựa trên memory-based (UserKNN-cosine, UserKNN-pearson, ItemKNN-cosine và ItemKNN-pearson) kém chính xác hơn các mô hình dựa trên mô hình (Matrix Factorization, Distributed Deep Learning (ALS) và Singular value decomposition (SVD)).
- Mô hình của chúng tôi (có tên gọi là RRS và RRS-non) có MSE, RMSE và MAE thấp nhất trong bảng, cho thấy mô hình của chúng tôi đạt hiệu quả vượt trội nhất so với các mô hình được so sánh.

Bảng 3. Kết quả so sánh với baselines của các mô hình Lọc cộng tác

Model	MSE	RMSE	MAE
Memory-based			
UserKNN-cosine	0.3799	0.6163	0.4820
UserKNN-pearson	0.5606	0.7487	0.6600
ItemKNN-cosine	0.4334	0.6583	0.5420
ItemKNN-pearson	0.5305	0.7283	0.6347
Model-based			
Matrix Factorization	0.1056	0.3250	0.2449
Distributed Deep Learning (ALS)	0.0887	0.2978	0.2201
Singular value decomposition (SVD)	0.1039	0.3221	0.2030
Ours			
RRS	0.0268	0.1638	0.1256
RRS-non	0.0607	0.2465	0.1944

RRS-non cũng là một mô hình khá chính xác, nhưng không chính xác như RRS. Điều này chứng tỏ mô hình Deep Learning ngữ cảnh được chúng tôi thêm vào dựa trên phân tích bình luận có sự ảnh hưởng đến mức độ hiệu quả của mô hình đề xuất khách sạn. Nhìn chung, RRS là một mô hình khuyến nghị chính xác và hiệu quả. Nó có thể được sử dụng để tạo ra các đề xuất cho người dùng tốt nhất trong các mô hình được sử dụng để so sánh.

5 Tích hợp mô hình RRS vào xử lý dữ liệu trực tuyến

5.1 Công nghệ sử dụng

Kafka

Kafka là một hệ thống xử lý thông tin theo thời gian thực và phân tán, được phát triển bởi Apache Software Foundation. Được xây dựng trên kiến trúc publish-subscribe, Kafka được áp dụng rộng rãi trong xử lý dữ liệu của các ứng dụng phân tán, phân tích dữ liệu, ... Hệ thống này cho phép các ứng dụng gửi và nhận các thông điệp theo thời gian thực, từ đó tạo ra một hệ sinh thái dữ liệu liên tục và hoạt động theo thời gian thực.

Các ưu điểm nổi bật của Kafka bao gồm khả năng xử lý hàng triệu tin nhắn mỗi giây và duy trì tính toàn vẹn dữ liệu trong mạng lưới phân tán. Ngoài ra, Kafka cũng hỗ trợ việc lưu trữ dữ liệu dựa trên đa nền tảng, cho phép ứng dụng truy xuất và xử lý dữ liệu một cách linh hoạt.

Cassandra

Cassandra là một hệ quản trị cơ sở dữ liệu phân tán, có khả năng mở rộng cao và xử lý dữ liệu theo thời gian thực. Được phát triển bởi Apache Software Foundation, Cassandra được thiết kế cho những ứng dụng với lưu lượng truy cập lớn và yêu cầu tính sẵn sàng cao. Với việc sử dụng kiến trúc phân tán và khả năng đồng bộ theo mô hình hòm giao tiếp và cung cấp tính toàn vẹn dữ liệu trong mạng lưới phân tán.

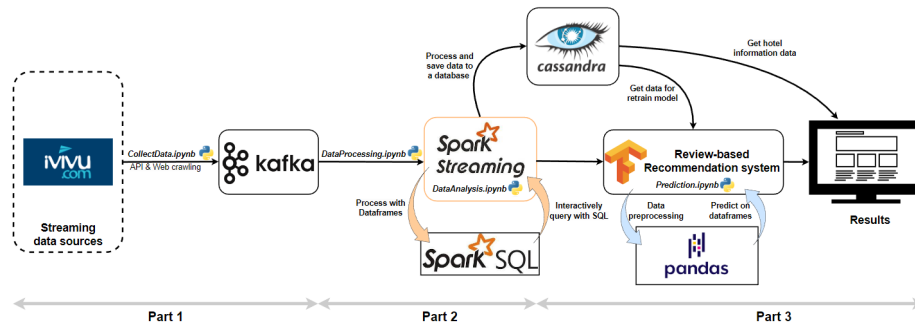
Cùng với khả năng xử lý dữ liệu nhanh, khả năng mở rộng linh hoạt và khả năng tự phục hồi, Cassandra còn cung cấp hiệu suất cao và độ tin cậy, làm nền tảng cho những ứng dụng thời gian thực và yêu cầu dữ liệu xử lý tương tác cao.

Spark

Spark là một hệ thống xử lý dữ liệu phân tán và tính toán cực mạnh. Được phát triển bởi Apache Software Foundation, Spark cung cấp một nền tảng linh hoạt và mở rộng để xử lý và phân tích các tập dữ liệu lớn. Với khả năng xử lý dữ liệu song song và tính toán trên bộ nhớ, Spark có thể thực hiện các tác vụ tính toán phức tạp nhanh chóng. Spark hỗ trợ nhiều ngôn ngữ lập trình như Scala, Java, Python và R.

Spark cung cấp một loạt các tính năng mạnh mẽ như xử lý dữ liệu thời gian thực, phân tích dữ liệu, học máy và xử lý đồ thị. Nó cũng tích hợp tốt với các công cụ dữ liệu như Hadoop và Cassandra. Spark cũng hỗ trợ giao diện tương tác (REPL) cho việc thử nghiệm và thực thi các lệnh trực tiếp trên dữ liệu. Spark giúp tăng tốc quá trình phân tích và tích hợp dữ liệu, giúp nhanh chóng tìm ra thông tin và insights từ dữ liệu lớn để ra quyết định kinh doanh tốt hơn.

5.2 Hệ thống Streaming của mô hình khuyến nghị khách sạn



Hình 5. Kiến trúc end-to-end của dữ liệu trực tuyến RRS

Một trong những yêu cầu hàng đầu của hệ thống đề xuất khách sạn hiệu quả là phải đề xuất được cho lượng lớn dữ liệu người dùng trên các trang web du lịch. Điều này giúp cho các hệ thống đề xuất khách sạn nói chung và hệ thống đề xuất khách sạn cho người Việt Nam nói riêng đề xuất được các khách sạn cho người dùng một cách tốt hơn, đồng thời giảm bớt khối lượng công việc cho người quản trị.

Chúng tôi đề xuất xây dựng một hệ thống đề xuất khách sạn thời gian thực có khả năng đề xuất các khách sạn phù hợp với người dùng thông qua lịch sử du lịch mới nhất của họ nhằm giải quyết bài toán trên.

Tổng quan

Chúng tôi đã xây dựng thành công hệ thống có khả năng xử lý lượng lớn dữ liệu theo thời gian thực từ website Ivivu.com bằng cách tiến hành thu thập và thử nghiệm xử lý dữ liệu truyền trực tuyến. Phần này trình bày các phần của hệ thống triển khai thời gian thực được đề xuất.

Phần 1 - Thu thập dữ liệu trực tuyến

Trong phần này, chúng tôi xây dựng hệ thống kết nối với Ivivu Data API bằng URL của nhà phát triển để lấy những thông tin được công khai. Đầu tiên, chúng tôi tìm kiếm URL và thiết lập các thông tin cần thiết để sử dụng Ivivu Data API, như "Connection", "Host", "User-Agent". Tiếp theo, trong phần QUERY, chúng tôi xây dựng hàm tạo đường dẫn URL để truy vấn đến từng địa điểm rồi đến các khách sạn. Ngoài ra, các tham số khác được thiết lập như maxHotel để quy định số lượng khách sạn tối đa của một địa điểm, minCmt và maxCmt để quy định số phạm vi số lượng bình luận của một khách sạn.

Sau đó, chúng tôi cũng xây dựng một luồng socket TCP liên kết giữa Ivivu Data API và Kafka, luồng socket này sẽ thực hiện việc chuyển dữ liệu trực tuyến từ Ivivu vào Kafka để lưu trữ dữ liệu thô nhằm phục vụ các mục đích tiếp theo trong dự án.

Phần 2 - Xử lý trực tuyến và lưu trữ dữ liệu

Dữ liệu được thu thập và lưu trữ trực tuyến thông qua Ivivu Data API và Kafka sẽ được tiến xử lý và chia làm hai luồng để truyền đến cơ sở dữ liệu Cassandra và hệ thống chính để tiến hành dự đoán và đề xuất khách sạn theo quy trình được mô tả trong Hình.

Dữ liệu trực tuyến được lấy từ Kafka thông qua Spark Streaming sau đó tiến hành tiền xử lý dữ liệu. Các comment được người dùng viết trên nhiều thiết bị và với khoảng thời gian khác nhau. Điều này dẫn đến việc không đồng bộ về unicode trong tiếng Việt và cần được xử lý. Tiếp theo là quá trình xử lý cơ bản như đưa các kí tự chữ in hoa về chữ thường, xóa các ký tự không cần thiết (những dấu câu dư thừa, các icon bộc lộ cảm xúc) hay xóa đi các khoảng trắng thừa, khôi phục các chữ viết tắt, chuyển đổi tiếng anh sang tiếng Việt. Phần tiếp theo sẽ giúp cho mô hình có thể xác định tốt hơn ý nghĩa của từ vựng chính là tách từ (word tokenize). Khi đó một từ được tạo nên từ hai tiếng hoặc hơn sẽ được kết nối lại với nhau để thể hiện rõ đó chỉ là một từ.

Sau khi tiền xử lý, dữ liệu được chuẩn hóa với chất lượng cao sẽ được lưu trữ trong cơ sở dữ liệu Cassandra và sử dụng để dự đoán nhân và đề xuất với mô hình RRS. Đối với luồng dữ liệu được lưu trữ vào Cassandra, chúng tôi xây dựng một hệ thống kết nối thông qua Spark Streaming để kết nối đúng với Key trong cơ sở dữ liệu. Sau đó đẩy những dữ liệu đã được xử lý vào trong Cassandra thông qua luồng liên kết được tạo.

Phần 3 - Đề xuất khách sạn với dữ liệu thời gian thực

Sau khi dữ liệu được xử lý trực tuyến và lưu trữ, chúng tôi tiến hành quá trình đề xuất thời gian thực. Quá trình được thực hiện qua ba giai đoạn: xử lý dữ liệu, dự đoán chủ đề và thái độ khách hàng từ bình luận, và thực hiện đề xuất khách sạn.

Giai đoạn thứ nhất, tên người dùng (User name) và bình luận đánh giá (Comment) sẽ được chúng tôi trích xuất và tiếp tục xử lý. Đầu tiên, tiến hành loại bỏ stopwords ra khỏi bình luận và phần còn lại sẽ được tách thành các tokens thông qua phoBERT-pretrained. Kế tiếp, bình luận sẽ đưa về độ dài chuẩn là 80 thông qua việc thêm các giá trị đệm vào cuối bình luận. Sau đó đưa dữ liệu về dạng tensor để thực hiện trích xuất đặc trưng thông qua phoBERT-pretrained.

Giai đoạn hai, bình luận và đặc trưng của nó được đưa vào mô hình dự đoán chủ đề và cảm xúc người dùng. Đầu ra của mô hình thể hiện thái độ người dùng đối với khách sạn và các chủ đề được nhắc đến trong bình luận đó. Đầu ra được sử dụng làm ngữ cảnh cho quá trình đề xuất khách sạn.

Giai đoạn cuối cùng, thực hiện trích xuất lịch sử khách hàng thông qua tên người dùng. Lịch sử này sẽ được kết hợp với ngữ cảnh (đầu ra của giai đoạn 2) để tạo ra đầu vào cho mô hình đề xuất của chúng tôi. Mô hình tiến hành dự đoán Rating của các khách sạn mà khách hàng chưa đến. Lọc ID của 10 khách sạn có Rating cao nhất và tiến hành trích xuất thông tin khách sạn từ cơ sở dữ liệu trong Cassandra. Kết quả cuối cùng của mô hình là danh sách 10 khách sạn được hệ thống đề xuất.

6 Kết quả thực nghiệm mô hình

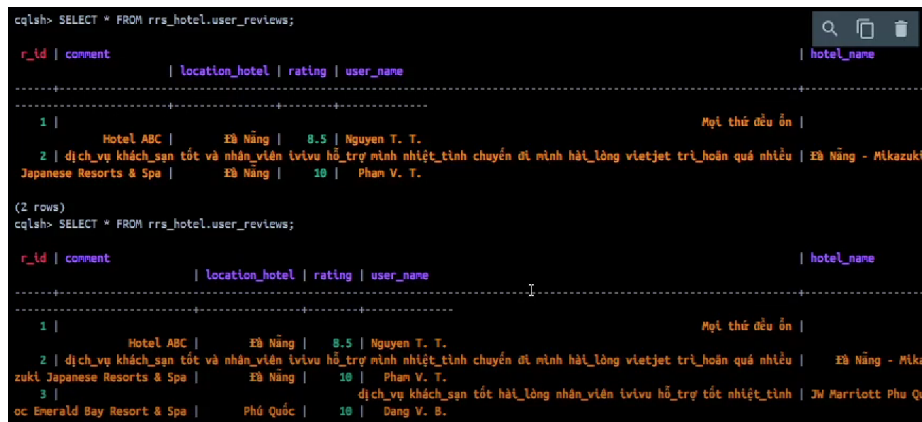
Ở giai đoạn này, chúng tôi thực hiện một thử nghiệm mẫu để triển khai quy trình truyền dữ liệu vào Cassandra và truyền vào hệ thống đề xuất RRS hoạt động trên một hệ thống với GPU NVIDIA GeForce GTX 1650, CPU AMD Ryzen 7 3750H và RAM dung lượng 8G. Thử nghiệm này được triển khai để phân tích độ hiệu quả của hệ thống. Quy trình được thực hiện liên tục trong một khoảng thời gian nhất định nhằm lấy những dữ liệu mới theo thời gian thực và đưa vào hai luồng hệ thống để xử lý ngay lập tức. Việc xây dựng và áp dụng thành công hệ thống đề xuất dựa trên thông tin bình luận trên trang web ivivu chứng tỏ giá trị từ nghiên cứu của chúng tôi. Hơn nữa, hệ thống đề xuất của chúng tôi đặt nền tảng cho các nghiên cứu tương lai về các hệ thống ứng dụng xử lý thời gian thực cho các dịch vụ du lịch.

Ở giai đoạn xử lý trực tuyến, kết quả được đánh giá bằng cách xác minh các kết quả dự đoán được trả về bởi hệ thống. Đầu tiên, chúng tôi lấy 500 bình

luận mới nhất theo thời gian thực từ Ivivu thông qua Ivivu Data API và được lưu trữ dưới dạng JSON. Các dòng dữ liệu thu thập được chứa nhiều thuộc tính khác nhau. Trong nghiên cứu này, chúng tôi tập chung vào 5 thuộc tính UserName, LocationHotel, HotelName, Rating, Comment để phân loại các nhận được truyền tải.

Sau khi có được dữ liệu, đối với luồng dữ liệu lưu trữ trên Cassandra, chúng tôi thực hiện lưu trữ liên tục như Hình 6. Kết quả cho thấy luồng xử lý này hoạt động tốt tự động cập nhật theo thời gian trung bình sau mỗi 0.86 giây và có thể phục vụ cho việc khôi phục dữ liệu nhiều mục đích khác trong tương lai.

Đối với luồng dữ liệu được đưa vào hệ thống đề xuất RRS, sau khi dữ liệu thu



```
cqlsh> SELECT * FROM rrs_hotel.user_reviews;
```

r_id	comment	location_hotel	rating	user_name	hotel_name
1		Hotel ABC	8.5	Nguyen T. T.	Một thứ đầu ổn
2	dịch_vụ khách_sạn tốt và nhân_viên tận_vụ hỗ_trợ mình nhiệt_tình chuyển đi mình hài_lòng vietjet tri_hoàn quá nhiều	Xi Nang	10	Phan V. T.	Xi Nang - Mikazuki Japanese Resorts & Spa

```
(2 rows)
cqlsh> SELECT * FROM rrs_hotel.user_reviews;
```

r_id	comment	location_hotel	rating	user_name	hotel_name
1		Hotel ABC	8.5	Nguyen T. T.	Một thứ đầu ổn
2	dịch_vụ khách_sạn tốt và nhân_viên tận_vụ hỗ_trợ mình nhiệt_tình chuyển đi mình hài_lòng vietjet tri_hoàn quá nhiều	Xi Nang	10	Phan V. T.	Xi Nang - Mikazuki Japanese Resorts & Spa
3	dịch_vụ khách_sạn tốt hài_lòng nhân_viên tận_vụ hỗ_trợ tốt nhiệt_tình	Phu Quoc	10	Dang V. B.	JW Marriott Phu Quoc Emerald Bay Resort & Spa

Hình 6. Kết quả triển khai hệ thống truyền dữ liệu vào Cassandra

thập được xử lý, chúng sẽ được đưa vào mô hình đề xuất để xử lý và kết quả thu được như hình 7 của mỗi dòng dữ liệu có độ trễ theo thời gian trung bình trong 1.47 giây. Thời gian này phù hợp để ứng dụng hệ thống với mục đích cập nhật thông tin dự đoán liên tục mỗi khi dữ liệu thay đổi. Hỗ trợ cho việc trực quan hóa hệ thống đề xuất RRS cho người dùng.

7 Kết luận

Từ kết quả nghiên cứu, chúng tôi đã chứng minh rằng mô hình với ngữ cảnh được trích xuất từ nhận xét của người dùng mang lại kết quả tốt hơn. Chúng tôi đề xuất một phương pháp xây dựng mô hình hệ thống gợi ý dựa trên lịch sử phản hồi của người dùng trên các trang web du lịch điện tử. Đồng thời, trong quá trình nghiên cứu, chúng tôi cũng đóng góp một tập dữ liệu được tiền xử lý bao gồm thông tin như tên người dùng, tên khách sạn, nhận xét tiếng Việt, v.v. Ngoài ra, một phần nhỏ đã được chú thích với tỷ lệ đồng thuận tương đối cao hơn 80% cho hệ thống gợi ý khách sạn cho mục đích nghiên cứu. Kết quả của

```

Showing live view refreshed every 5 seconds
Seconds passed: 15
Data raw:
-----
Comment      khách sạn hơi xa trung tâm, phòng ốc tốt, buff...
Hotel_name    Khách Sạn LADALAT
Location_hotel Đà Lạt
Rating        6.0
User_name     Dang T. D.
Name: 141, dtype: object

5/5 [=====] - 0s 3ms/step
History:
-----


| UserId | HotelId    | Location_hotel                     | Comment                                                  | Rating |
|--------|------------|------------------------------------|----------------------------------------------------------|--------|
| 3      | Dang T. D. | Khách Sạn LADALAT                  | Đà Lạt khách sạn hơi xa trung tâm phòng ốc tốt buffet... | 6.0    |
| 1735   | Dang T. D. | Khu nghỉ dưỡng Pandanus Phan Thiết | Phan Thiết tôi thấy phòng hơi cũ dịch vụ cũng tạm được   | 8.0    |
| 1856   | Dang T. D. | The Sunriver Boutique Hotel Hue    | Huế chỗ nghỉ quá tuyệt vời trung tâm thành phố khá...    | 10.0   |


-----
Top 10 hotel recommendations:
-----
Dalat Edensee Lake Resort & Spa
Khu nghỉ dưỡng Movenpick Cam Ranh
Intercontinental Đà Nẵng Sun Peninsula Resort
New Style House Hotel
Khu nghỉ dưỡng Amiana Nha Trang
Vinpearl Condotel Beachfront Nha Trang
Khu nghỉ dưỡng Crown Retreat Quy Nhơn
Khu nghỉ dưỡng Holiday Inn Hồ Tràm
Khách sạn Four Points by Sheraton Đà Nẵng
Khu nghỉ dưỡng Seava Hồ Tràm

```

Hình 7. Kết quả triển khai hệ thống đề xuất RRS

mô hình đề xuất của chúng tôi được đo bằng MSE là 0,0268, RMSE là 0,1638, MAE là 0,1256 đáng kể hơn so với mô hình không có ngữ cảnh trên cùng tập dữ liệu. Kết quả dự đoán thực nghiệm của hệ thống gợi ý dựa trên đánh giá (RRS) của chúng tôi cũng cho thấy rằng các khách sạn được gợi ý bởi RRS của chúng tôi tương đồng với lịch sử của người dùng. Cùng với đó là hệ thống lấy dữ liệu trực tuyến từ Ivivu Data API truyền vào Kafka và xử lý trực tuyến với Spark Streaming với độ trễ trong luồng lưu vào Cassandra là 0.86 giây, với độ trễ trong luồng xử lý hệ thống đề xuất RRS là 1.47 giây.

Tuy nhiên, chủ đề vẫn còn nhiều hạn chế ảnh hưởng đến chất lượng của tập dữ liệu và phương pháp luận không liên kết. Những sai sót trong quá trình gán nhãn, sự mất cân bằng của nhãn trong việc thu thập dữ liệu.

Trong tương lai, chủ đề có thể được phát triển thêm để cải thiện tính chặt chẽ của phương pháp luận, chẳng hạn như giải quyết các vấn đề mà chủ đề đã gặp phải, gán nhãn cho toàn bộ tập dữ liệu, xây dựng một bản demo web của kết quả nghiên cứu.

Tài liệu

1. Shambour, Qusai Y., Ahmad Adel Abu-Shareha, and Mosleh M. Abualhaj. "A Hotel Recommender System Based on Multi-Criteria Collaborative Filtering." In-

- formation Technology and Control 51.2 (2022): 390-402.
2. Jannach, Dietmar, Markus Zanker, and Matthias Fuchs. "Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations." *Information Technology & Tourism* 14 (2014): 119-149.
3. Kaya, Buket. "A hotel recommendation system based on customer location: a link prediction approach." *Multimedia Tools and Applications* 79 (2020): 1745-1758.
4. Al-Ghobari, Mohanad, Amgad Muneer, and Suliman Mohamed Fati. "Location-Aware Personalized Traveler Recommender System (LAPTA) Using Collaborative Filtering KNN." *Computers, Materials & Continua* 69.2 (2021).
5. Abbasi, Fatemeh, Ameneh Khadivar, and Mohsen Yazdinejad. "A grouping hotel recommender system based on deep learning and sentiment analysis." *Journal of Information Technology Management* 11.2 (2019).
6. Dara, Sriharsha, C. Ravindranath Chowdary, and Chintoo Kumar. "A survey on group recommender systems." *Journal of Intelligent Information Systems* 54.2 (2020): 271-295.
7. Lü, Linyuan, et al. "Recommender systems." *Physics reports* 519.1 (2012): 1-49.
8. Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." *IEEE Internet computing* 7.1 (2003): 76-80.
9. Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.
10. Paterek, Arkadiusz. "Improving regularized singular value decomposition for collaborative filtering." *Proceedings of KDD cup and workshop*. Vol. 2007. 2007.