Hotel Recommender System Menggunakan Metode Pendekatan Graph pada Dataset Trivago

Ricky Sunartio[1] , Henry Novianus Palit[2] , Andre Gunawan[3]
Program Studi Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra
Jl. Siwalankerto, 121-131 Surabaya 60236, Indonesia
Telp(031)2983455 Fax.(031)8417658
miscavel@gmail.com, hnpalit@petra.ac.id, andre.gunawan@petra.ac.id

ABSTRAK

Trivago adalah media pencarian hotel beranah global, di mana pengguna akan disodorkan daftar hotel berdasarkan atas tempat yang ingin dituju, beserta kondisi lain yang disertakan (opsional), misalkan pada jarak harga tertentu. Sebagai sebuah media pencarian, salah satu kendala terbesar yang dialami Trivago dalam memberikan daftar hotel adalah : mengetahui "keinginan" pengguna. Trivago menggunakan sebuah recommender system untuk menentukan urutan hotel yang ditampilkan pada sebuah sesi pencarian berdasarkan pada "keinginan" pengguna di sesi tersebut. Dalam upaya untuk meningkatkan kualitas recommender system tersebut, Trivago bekerjasama dengan peneliti dari TU Wien, Politecnico di Milano, dan Karlsruhe Institute of Technology, untuk mengadakan RecSys Challenge 2019 sebagai perlombaan data science tahunan di bawah ACM Recommender Systems Conference.

Penelitian kali ini akan berfokus pada penggunaan pendekatan berbasis graph dalam menghasilkan recommender system dengan menggunakan dataset Trivago pada RecSys Challenge 2019. Pendekatan berbasis graph terbukti sesuai untuk digunakan dalam dataset yang berbasis time-series, misalkan penggunaan fitur graph dalam memprediksi apabila pengunjung akan melakukan pembelian pada sebuah sesi online shopping [3]. Penelitian tersebut membandingkan tingkat akurasi prediksi saat menggunakan fitur graph dan fitur tradisional, dan terlihat peningkatan sebesar 5 – 10%.

Pendekatan graph yang diadopsi untuk penelitian ini adalah Markov Chain, sebuah probabilistic graphical model. Penelitian menguji beberapa model Markov Chain dengan length, depth, dan order yang berbeda, dan setiap model diuji dengan menggunakan metris Mean Reciprocal Rank (MRR) sesuai dengan ketentuan challenge. Pada akhir penelitian, ditemukan bahwa model Markov Chain dengan nilai MRR tertinggi didapatkan pada saat nilai length = 1, depth = 0, dan order = 1.

Kata Kunci: *Big Data, Recommender System,* Trivago, Markov Chain

ABSTRACT

Trivago is a global online application that provides service in searching for accomodations based on the user's destination, and other specific filters such as price range, ratings, and available features. As an accommodation searching app, one of the major challenges that Trivago faces is providing a list of hotels that match the user's preference, in order to increase the rate at which the user engages in a transaction. Trivago uses a recommender system that can deduce a user's preference in a session, and displays a number of hotels that are considered suitable for that user. In an attempt to improve the quality of their system, Trivago works together with researchers from TU Wien, Politecnico di Milano, and Karlsruhe Institute of Technology, to conduct RecSys Challenge 2019 as an annual science competition under ACM Recommender Systems Conference.

This paper is going to be focused on the use of graph-based models in creating a recommender system using Trivago's dataset provided in RecSys Challenge 2019. Graph-based models have been proven to be fairly effective in dealing with time series data, as shown in a research that studied online shoppers' behavior using graphs [3]. Baumann's research has shown an increase in accuracy by 5 – 10% when using graph features in predicting whether an online shopper would purchase a product as compared to using traditional features.

The graph-based model used in this research would be Markov Chain, a probabilistic graphical model. This research would test several Markov Chain models with varying length, depth, and order, measuring each model with a metric called Mean Reciprocal Rank (MRR) as per mentioned in the challenge. At the end of the research, it is concluded that the model that yields the highest MRR uses Markov Chain with length = 1, depth = 0, and order = 1.

Keywords: *Big Data, Recommender System,* Trivago, Markov Chain

1. PENDAHULUAN

Trivago adalah media pencarian hotel beranah global, di mana pengguna akan disodorkan daftar hotel berdasarkan atas tempat yang ingin dituju, beserta kondisi lain yang disertakan (opsional), misalkan pada jarak harga tertentu. Sebagai sebuah media pencarian, salah satu kendala terbesar yang dialami Trivago dalam memberikan daftar hotel adalah : mengetahui "keinginan" pengguna. Trivago menggunakan sebuah recommender system untuk menentukan urutan hotel yang ditampilkan pada sebuah sesi pencarian berdasarkan pada "keinginan" pengguna di sesi tersebut. Dalam upaya untuk meningkatkan kualitas recommender system tersebut, Trivago bekerjasama dengan peneliti dari TU Wien, Politecnico di Milano, dan Karlsruhe Institute of Technology, untuk mengadakan RecSys Challenge 2019 sebagai perlombaan data science tahunan di bawah ACM Recommender Systems Conference.

Penelitian kali ini akan berfokus pada penggunaan pendekatan berbasis graph dalam menghasilkan *recommender system* dengan menggunakan dataset Trivago pada RecSys Challenge 2019. Pendekatan berbasis graph terbukti sesuai untuk digunakan dalam dataset yang berbasis time-series, misalkan penggunaan fitur graph dalam memprediksi apabila pengunjung akan melakukan pembelian pada sebuah sesi online shopping [3]. Penelitian tersebut membandingkan tingkat akurasi prediksi

saat menggunakan fitur graph dan fitur tradisional, dan terlihat peningkatan sebesar 5 – 10%.

2. DASAR TEORI

2.1 Trivago

Trivago adalah aplikasi pencarian hotel yang pertama kali dikonseptualisasikan pada tahun 2005 di Düsseldorf oleh Rolf Schrömgens, Peter Vinnemeier dan Stephan Stubner. Trivago memiliki misi untuk menjadi sumber informasi independent bagi wisatawan dalam mencari hotel yang sesuai di destinasi mereka, sekaligus membantu hotel dengan berbagai skala dan popularitas dalam mengembangkan bisnis mereka melalui advertising. Aplikasi Trivago dapat diakses baik melalui website, maupun melalui application yang tersedia pada Google Playstore dan Apple Store [1].

Penelitian kali ini menggunakan dataset milik Trivago sebagai dasar pembentukan sistem rekomendasi, serta sebagai wadah untuk menilai kebaikan sistem.

2.2 Mean Reciprocal Rank

Reciprocal Rank (RR) adalah metode perhitungan statistik yang umum digunakan untuk menghitung tingkat kebenaran jawaban pada sebuah daftar jawaban. Nilai RR sesuai dengan inverse dari posisi jawaban di daftar jawaban yang diberikan [1].

Tabel 1. Contoh Perhitungan MRR

Jawaban	Daftar Jawaban	RR
B	A, **B**, C	1 / 2
A	**A**, B, C	1 / 1
C	A, B, **C**	1 / 3

Pada dataset Trivago, jawaban adalah item_id yang di clickout, sedangkan daftar jawaban adalah hasil rekomendasi. Semakin nilai MRR mendekati 1, semakin tinggi posisi hotel yang diinginkan pada daftar rekomendasi, sehingga semakin akurat hasil rekomendasi. Tabel 1 menunjukkan contoh perhitungan nilai MRR.

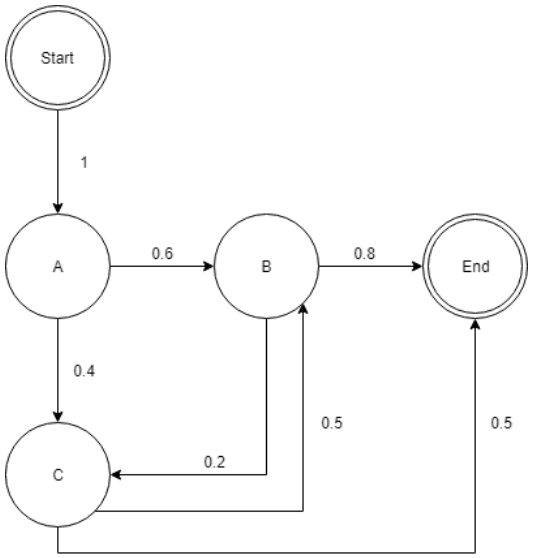
2.3 Markov Chain

Markov Chain adalah sebuah sistem matematika yang terdiri dari dua atau lebih state, di mana pergerakan dari satu state ke state lainnya dipengaruhi oleh probabilitas [4].

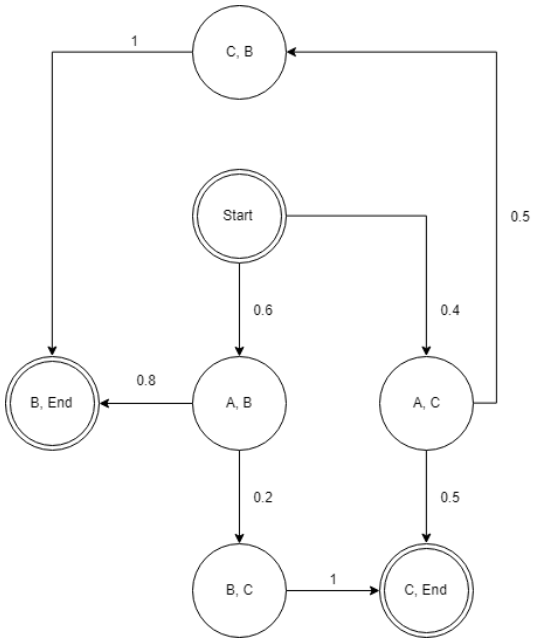
Pemetaan graph akan menggunakan Markov Chain sebagai dasar teori. Dalam memberikan rekomendasi, Markov Chain melihat state akhir dari sebuah sequence, kemudian mengurutkan state lain disekitarnya berdasarkan besar probabilitas transisi.

Pada *recommendation* menggunakan Markov Chain, terdapat 2 parameter yang dapat divariasi, yaitu order dan depth. Recommendation dilakukan dengan melihat probabilitas n state terakhir menuju state berikutnya, di mana n adalah order dari model. Depth merujuk pada jumlah iterasi yang dilakukan dalam mencari jalur menuju state lain.

Gambar 1 menunjukkan ilustrasi dari *1st Order Markov Chain*, di mana pada model ini terdapat probabilitas transisi dari obyek A menuju ke obyek B sebesar 0.6. Gambar 2 menunjukkan ilustrasi dari *3rd Order Markov Chain*, di mana didapati bahwa probabilitas sekuens {A, B} diikuti oleh sekuens {B, C} adalah 0.2. Meningkatkan *order* dari *Markov Chain* memperpanjang kombinasi sekuens dari setiap *node* sesuai dengan *order*.



Gambar 1. First Order Markov Chain



Gambar 2. Second Order Markov Chain

2.4 Cosine Similarity

Cosine Similarity adalah metode untuk mencari kedekatan antar item dengan melihat nilai cos(angle) dari kedua item tersebut. Cosine Similarity sering digunakan untuk item-based recommendation, di mana item – item yang berhubungan akan direkomendasikan bersama.

Tabel 2. User Item Matrix

	M1	M2	M3	M4	M5
U1	4	4.5			4.5
U2		2	4.5	3.5	5
U3		4	3.5	4	
U4	4.5	4		3	
U5	3	2.5	4.5		4

Pada Tabel 2, terdapat 5 user{U1, U2, U3, U4, U5} dan 5 item {M1, M2, M3, M4, M5}, dengan setiap user memberikan nilai untuk setiap item. Kedekatan antar item tersebut dapat dihitung dengan formula :

$$\cos\theta = \frac{4.5 \times 4 + 4 \times 4.5 + 3 \times 2.5}{\sqrt{4.5^2 + 4^2 + 3^2}\sqrt{4^2 + 4.5^2 + 2.5^2}} \approx 0.992\ (3.s.f) \quad (1)$$

Untuk melihat nilai kedekatan antara {M1} dengan {M2}, dapat digunakan formula tersebut dengan memasukkan {M1} dan {M2} sebagai M1 dan M2, seperti contoh pada (1).

Dengan menghitung nilai cos(angle) antara {M1} dengan item lainnya, dapat dilakukan rekomendasi dengan mengurutkan {M2, M3, M4, M5} sesuai dengan nilai cos(angle) tersebut, dengan nilai tertinggi pada posisi teratas. Semakin nilai cos(angle) mendekati 1, artinya angle antar item semakin mendekati 0, yang berarti kedua item tersebut semakin memiliki similarity.

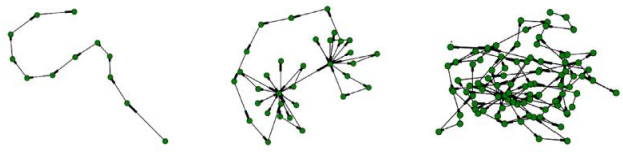
2.5 Apache Hadoop

Apache Hadoop adalah sebuah opensource software framework untuk memproses data dalam jumlah besar dengan memanfaatkan pemrosesan secara paralel. Hadoop pertama kali dirilis pada tahun 2005 oleh Doug Cutting dan Mike Cafarella. Apache Hadoop terdiri dari beberapa modul :

1. Hadoop Common yang memuat library dan file – file dasar untuk keperluan modul lainnya.
2. Hadoop Distributed File System (HDFS) sebagai file system yang terdistribusi untuk men-support data dengan bandwith yang besar.
3. Hadoop YARN sebagai resource management platform yang mengatur pembagian resource pada setiap cluster serta melakukan scheduling.
4. Hadoop MapReduce sebagai programming model untuk memproses data dalam jumlah besar.

2.6 Tinjauan Studi

Baumann mengadakan penelitian untuk menggunakan graph metrics dalam menentukan apakah seorang pengunjung pada sebuah sesi online shopping akan melakukan pembelian [3]. Menurut Baumann, pengguna dalam sebuah sesi online shopping memiliki end result yang terkait dengan behaviour pengguna dalam sesi. Behaviour ini ditinjau dari pola interaksi pengguna di sesi tersebut. Baumann memetakan pola interaksi ini dalam bentuk graph, di mana setiap node melambangkan page yang diakses oleh pengguna.



Gambar 3. Bentuk Graph yang menunjukkan perilaku yang berbeda – beda

Penelitian Baumann menunjukkan adanya peningkatan akurasi prediksi sebesar 5 – 10% ketika menggunakan fitur graph pada algoritma prediksi state-of-the-art seperti Generalized Linear Model (GLM), Random Forest (RF), dan Gradient Boosting (GB) dibandingkan dengan menggunakan fitur tradisional.

Shudong Liu mengadakan penelitian untuk memberikan rekomendasi place of interest pada pengguna Location Based Social Networks (LBSN) dengan menggunakan dynamic multi-order Markov Model dengan tambahan parameter geological closeness dan popularity [7]. Liu memetakan Markov Model untuk setiap pengguna di mana node berisikan tempat – tempat yang pernah dikunjungi oleh pengguna tersebut. Rekomendasi diberikan dengan melihat n tempat terakhir yang dikunjungi pengguna, di mana n adalah order dari Markov Model yang dipilih.

Selain multi-order Markov Model, Liu juga memfaktorkan kedekatan lokasi yang akan direkomendasikan dengan n tempat yang terakhir dikunjungi, serta tingkat popularitas lokasi tersebut pada saat itu. Ketiga faktor ini digabungkan menjadi sebuah unified model dengan weight yang berbeda – beda.

Penelitian Shudong menunjukkan bahwa multi-order Markov Model mendapatkan hasil F-measurement 15% di atas LORE dan 5% di atas Rank-GeoFM dengan menggunakan unified model tersebut.

3. DESAIN SISTEM

3.1 Desain Arsitektur Sistem

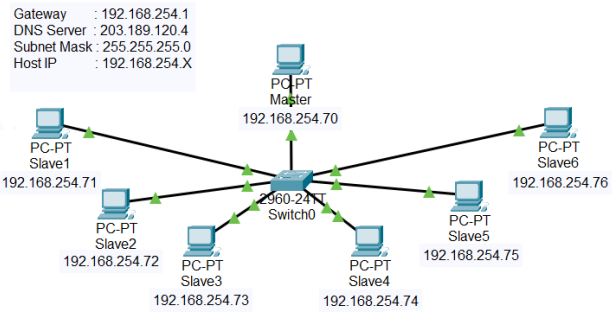
Komputer yang digunakan adalah komputer pada laboratorium Jaringan Komputer. Jumlah komputer yang digunakan adalah 7 buah, dengan pembagian 1 komputer sebagai master dan 6 komputer sebagai slave merujuk pada Gambar 2.7. Spesifikasi hardware dari setiap komputer tersebut adalah sebagai berikut :

- Processor : Intel Core i5-4570 CPU @ 3.20GHz x 4
- Graphics : Intel Haswell Desktop
- Memory : 7,7 GiB
- Gnome : 3.28.2
- OS type : 64-bit
- Disk : 479,8 GB
- Connection : Fast Ethernet 100 Mbps

Dengan spesifikasi software sebagai berikut :

- Ubuntu 18.04.3 LTS
- Java SE Development Kit 12
- Hadoop 3.2.1
- SSH

Sistem yang dibentuk mengikuti topologi logika seperti gambar berikut :



Gambar 4. Desain Topologi Logika

Merujuk pada Gambar 4, komputer master memiliki alamat IP 192.168.254.70, dan 6 komputer slave memiliki range IP dari 192.168.254.71 – 192.168.254.76. Jaringan menggunakan subnet /24, serta gateway dan dns server digunakan untuk koneksi internet saat melakukan update dan instalasi software yang diperlukan.

3.2 Pengolahan Data

Inti proses adalah mengubah bentuk data dari format :

user_id,session_id,timestamp,step,action_type
,reference,platform,city,device,current_filters,impressions,price

Menjadi format berikut :

session_id;city|step,action_type[reference]->step,action_type[reference]->…->step,action_type[reference]

Format baru memudahkan pemetaan kedalam Markov Chain dengan mengelompokkan data berdasarkan atas session_id, dan menekankan urutan pelaksanaan melalui step dan letak dalam string yang dipisahkan dengan delimiter '->'. Format data sedikit berubah ketika aksi berupa clickout menjadi :

step,action_type[reference][impressions][prices]

Karena kolom impressions dan prices akan digunakan dalam membuat hasil rekomendasi, dan dua kolom tersebut hanya ditemukan pada baris dengan aksi clickout.

3.3 Dataset Splitting

Dataset akan di-split sebesar 90% untuk training, dan 10% untuk menguji keakuratan model. Dataset train.csv disusun berdasarkan timestamp secara ascending, dengan jarak waktu selama 1 minggu antara session dengan timestamp terkecil dan session dengan timestamp terbesar.

Seluruh percobaan dieksperimentasi terlebih dahulu dengan menggunakan 90% data awal sebagai training data dan 10% data akhir sebagai testing data, sehingga artinya akurasi menguji kemampuan model dalam menebak future data dengan menggunakan past data. Untuk mengurangi bias, dilakukan proses cross-validation di mana bagian 90% training data dan 10% testing data ditukar secara bergilir hingga seluruh bagian data sempat menjadi training dan testing set.

3.4 Additive Markov Chain

Model ini memberi nilai pada sebuah hotel berdasarkan total nilai conditional_probability menuju ke hotel tersebut berdasarkan atas hotel – hotel yang telah di-interaksikan :

$$\text{Score}(x) = \sum_{i=0}^{n} P(x \mid interacted[i]) \quad (2)$$

x = the hotel being evaluated
n = limit, the maximum being the number of steps (or interacted hotels) - 1, and the minimum being 0
i = step i, calculated backwards (last hotel has i = 0)
interacted[i] = hotel interacted at step i
P(x | interacted[i]) = probability of interacting with hotel x after interacting with interacted[i]

Proses rekomendasi adalah sebagai berikut:

1. Hitung total probabilitas untuk setiap hotel yang muncul untuk menentukan urutan ranking sesuai dengan rumusan (2).
2. Menyusun hasil rekomendasi hotel berdasarkan total probabilitas (descending)

Contoh Rekomendasi :

i. 2367266; 0.605
ii. 4290518; 0.460
iii. 2382352; 0.351
iv. 9010140; 0.0638
v. 2567194; 0.0568
vi. 2197228; 0.0568
vii. 2362306; 0.0355
viii. 7219768; 0.0306
ix. 2717648; 0.0306

3. Menghitung hasil akurasi dengan nilai MRR

Jika hotel yang di-clickout adalah 2367266. Untuk menghitung nilai MRR, maka kita ambil nilai (1 / posisi_hotel_2367266) pada daftar rekomendasi.

Karena hotel 2367266 berada pada posisi 1, maka nilai MRR adalah 1/1 = 1. Semakin MRR mendekati 1, maka semakin tinggi tingkat akurasi prediksi. Proses perhitungan MRR ini dilakukan untuk setiap session pada test data, dan kemudian di rata – rata.

4. ANALISA DAN PENGUJIAN

Tabel 3. Perbandingan MRR Setiap Model

1	AMC (Item), n = 0, depth = 1	0.208
2	AMC (Item), n = session_length – 1 with Linear Regression	0.217
3	AMC (Item), n = session_length - 1	0.245
4	AMC (Item), n = 0, order = 3	0.252
5	AMC (Item), n = 0, order = 2	0.259
6	AMC (Action[Item]), n = 0	0.264
7	AMC (Item), n = 0	0.286
8	AMC (Item), n = 0, ignore non-hotel nodes	0.304
9	AMC (Item), n = 0, ignore non-hotel nodes, ignore if not within impressions	0.346
10	Impressions only	0.458
11	AMC (Item), n = session_length - 1, ignore non-hotel nodes, ignore if not within impressions, stack remaining impressions	0.466
12	AMC (Item), n = 0, order = 2, ignore non-hotel nodes, ignore if not within impressions, stack remaining impressions	0.468
13	AMC (Item), n = 0, ignore non-hotel nodes, ignore if not within impressions, stack remaining impressions	0.472
14	AMC (Item), n = 0, ignore non-hotel nodes, ignore if not within impressions, stack remaining impressions, ignore Markov Chain if session is empty	0.539
15	AMC (Item), n = 0, ignore Chain if session is empty, sort remaining by average session price	0.544
16	AMC (Item), n = 0, ignore Chain if session is empty, sort remaining by last interacted price	0.552
17	AMC (Item), n = 0, ignore Chain if session is empty, stack last seen hotel, sort remaining by last interacted price	0.554
18	Stack last seen hotel, AMC (Item), n = 0, ignore Chain if session is empty, sort remaining by last interacted price	0.563
19	Stack last seen hotel, padding = 4, AMC (Item), n = 0, ignore Chain if session is empty, sort remaining by last interacted price	**0.575**

Perbandingan antar model Markov Chain murni dapat dilihat pada Tabel 3 pada indeks 1 – 8. Dari hasil eksperimen tersebut, didapati bahwa nilai MRR tertinggi didapatkan ketika menggunakan model [AMC (Item), n = 0, ignore non-hotel nodes]. Hal ini berarti dalam konteks dataset Trivago, intent dari pengguna dipengaruhi mayoritas hanya oleh hotel terakhir

yang diinteraksi. Ketika hotel – hotel selain hotel terakhir diikutsertakan dalam pertimbangan rekomendasi, seperti pada [AMC (Item), n = session_length – 1], [AMC (Item), n = 0, depth = 1], [AMC (Item), n = 0, order = 2], dan [AMC (Item), n = 0, order = 3], didapati bahwa nilai MRR menurun. Spekulasi penurunan ini adalah karena pada sebagian besar kasus, hotel – hotel yang diinteraksi pada awal sesi tidak mewakili preference dari pengguna, atau terjadi pergantian preference pada saat pengguna berinteraksi di sesi tersebut. Hasil analisa menunjukkan bahwa preference tersebut didapatkan mayoritas dari hotel terakhir yang diinteraksi, sehingga mengikutsertakan hotel – hotel lainnya hanya akan mem-pollute hasil rekomendasi dengan hotel – hotel yang tidak diinginkan.

Peningkatan nilai MRR yang signifikan didapatkan ketika menambahkan impressions filter, yaitu menghilangkan hotel – hotel pada daftar rekomendasi jika tidak ditemukan dalam daftar impressions. Impressions adalah kolom yang tidak terpakai pada model Markov Chain murni karena atribut tersebut tidak digunakan dalam probability graph. Analisa menunjukkan bahwa pada 99.2% sesi yang ada, jawaban ditemukan dalam daftar impressions. Fakta tersebut didukung oleh kenyataan bahwa pengguna hanya dapat melakukan clickout pada hotel yang tampil di layar aplikasi. Properti ini menjawab masalah utama dari daftar rekomendasi Markov Chain pada mulanya, yaitu daftar rekomendasi yang ter-pollute oleh hotel – hotel yang tidak relevan. Ketika impressions filter ini digunakan, ditemukan bahwa model Markov Chain dengan nilai MRR tertinggi tetap model [AMC (Item), n = 0, ignore non-hotel nodes], model yang hanya memperhitungkan hotel terakhir yang diinteraksi.

Peningkatan nilai MRR berikutnya didapatkan dengan mengubah baseline rekomendasi dari Markov Chain pada city node menjadi impressions. Hal ini karena city node memberikan hasil rekomendasi yang terlalu luas dan polluted dengan hotel – hotel yang tidak diinginkan. Impressions mengisi posisi baseline ini dengan sempurna. Baseline digunakan untuk mengisi kekosongan pada daftar rekomendasi yang berjumlah 25 hotel, dan karena jawaban dari 99.2% sesi terdapat dalam impressions, menggunakan impressions sebagai baseline memastikan bahwa nilai minimum RR dari sebuah sesi adalah 1/25, atau 0.04, dibandingkan dengan nilai minimum RR dengan city node baseline yaitu 0 jika hotel tidak berada dalam daftar. Pada tahap ini, nilai MRR model adalah 0.539, yang berarti jawaban berada pada posisi 1.86 pada umumnya, sebuah peningkatan yang cukup signifikan dari rekomendasi awal dengan rata – rata posisi 3.29 pada MRR 0.304.

Limitasi model kali ini ditemukan pada rata – rata panjang daftar rekomendasi yang pendek, yaitu 2.83. Menurut Tabel 5.23, hal ini berarti terdapat 20.1 hotel yang memiliki full reliance pada kemampuan prediksi impressions. Impressions sendiri memiliki nilai MRR sebesar 0.458, sehingga untuk dpat menembus MRR 0.539, impressions tersebut harus diolah. Analisa – analisa sebelumnya menunjukkan bahwa mengolah data lebih banyak dengan Markov Chain melalui perlebaran scope (meningkatkan nilai n), atau peningkatan depth berujung pada penurunan MRR. Untuk itu, dicarilah algoritma diluar Markov untuk mengolah sisa data tersebut.

Kandidat utama dari algoritma tambahan ini adalah menggunakan prices dari hotel – hotel yang telah diinteraksi pada sebuah sesi untuk mencari hotel – hotel yang serupa. Hipotesa yang dibuat adalah ketika melakukan pencarian, pengguna kerap memiliki budget tertentu, dan hotel – hotel yang dipilih tentunya berada dalam budget tersebut. Pemilihan ini dilakukan dengan cara mengurutkan daftar rekomendasi diluar Markov Chain berdasarkan pada perbedaan harga hotel dengan average price dari hotel – hotel yang telah diinteraksikan. Hipotesa ini terbukti memiliki kebenaran dari peningkatan nilai MRR dari 0.539 menjadi 0.544. Melihat analisa – analisa awal yang menitikberatkan relevansi pada hotel terakhir, dilakukan juga eksperimen dengan menggunakan last interacted hotel price, dan didapati bahwa nilai MRR naik menjadi 0.554. Hasil analisa ini sekali lagi menunjukkan bahwa pilihan terakhir pengguna diwakili mayoritas oleh hotel terakhir yang diinteraksikan pada sesi tersebut.

Melihat tingginya relevansi hotel terakhir pada model – model yang telah dieksperimentasikan, dicobalah model yang menambahkan last seen hotel ke dalam daftar rekomendasi, dan percobaan ini menghasilkan peningkatan MRR ke 0.563, dan 0.575 jika ditambahkan padding antara model Markov Chain dengan last seen hotels.

5. KESIMPULAN

Setelah dilakukan eksperimentasi dan analisa menggunakan berbagai model Markov Chain dalam membuat sistem rekomendasi hotel, dapat ditarik beberapa kesimpulan sebagai berikut :

- Markov Chain memiliki potensi dalam membuat daftar rekomendasi hotel pada dataset Trivago, menghasilkan nilai MRR 0.304 saat digunakan secara murni, dan 0.575 saat dikombinasi dengan fitur impressions, prices, dan last seen.
- Dataset Trivago memiliki Markov property,yang menyatakan bahwa probabilitas dari sebuah kejadian di masa depan hanya dipengaruhi oleh kejadian tepat 1 periode sebelum masa tersebut. Dalam hal ini, kejadian clickout pada sebuah hotel dipengaruhi, meskipun tidak hanya, namun mayoritas oleh hotel yang terakhir diinteraksikan pada sesi tersebut. Markov Chain memiliki banyak pengembangan, seperti variasi pada nilai n, depth, atau order dalam upaya untuk mempertimbangkan state selain state terakhir, namun pada penelitian kali ini terbukti bahwa model terbaik didapatkan ketika menggunakan Markov Chain dengan n = 0, depth = 1, dan order = 1 yang hanya mempertimbangkan state terakhir.
- Impressions memiliki pengaruh yang sangat besar dalam menentukan daftar rekomendasi. Hal ini dikarenakan 99.2% pengguna melakukan clickout pada hotel yang tampil pada daftar impressions, sehingga apapun model rekomendasi yang digunakan wajib menjadikan impressions sebagai himpunan semesta untuk mengurangi noise pada daftar rekomendasi.
- Karena properti impressions yang berlaku sebagai semesta, kemampuan model rekomendasi menjadi terbatasi oleh daftar impressions yang ada. Sebagai contoh, impressions filter mengurangi rata – rata panjang daftar rekomendasi Markov Chain dari 9.5 menjadi 2.83, mengurangi kemampuan Markov Chain sebanyak 70.2%. Di satu sisi, impressions memudahkan generasi daftar rekomendasi dengan membatasi semesta sehingga terjadi kenaikan nilai MRR, namun di sisi lain impressions membatasi performa model rekomendasi dengan semesta yang terbatas. Selain itu, hal ini juga berarti jika impressions memberikan daftar yang tidak sesuai, maka model rekomendasi juga akan terpengaruh oleh semesta tersebut.
- Aksi clickout tersentralisasi pada hotel – hotel yang berada pada posisi awal impressions, karena efek dari viewport aplikasi yang menampilkan hotel urut berdasarkan indeks. Karenanya, clickout pada sebuah hotel tidak hanya terpengaruh oleh preferensi pengguna, namun juga pada urutan hotel tersebut ditampilkan di layar oleh aplikasi Trivago.

6. DAFTAR REFERENSI

[1] ACM. 2019. URI = https://recsys.trivago.cloud/.

[2] Bappalige, S. P. 2014, August 26. *An introduction to Apache Hadoop for big data*. URI = https://opensource.com/life/14/8/intro-apache-hadoop-big-data.

[3] Baumann, A., Haupt, J., Gebert, F., & Lessmann, S. 2018. Changing perspectives: Using graph metrics to predict purchase probabilities. *Expert Systems With Applications 94*, 137 - 148.

[4] Brémaud, P. 2008. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues.* Springer.

[5] Chernev, A., Böckenholt, U., & Goodman, J. 2014. Choice overload: A conceptual review and meta-analysis. *Journal of Consumer Psychology*.

[6] Kumar, R., Raghu, M., Sarlós, T., & Tomkins, A. 2017. Linear Additive Markov Process. *The 26th International Conference on World Wide Web*, (hal. 411-419).

[7] Liu, S., & Wang, L. 2018. A self-adaptive point-of-interest recommendation algorithm based on a multi-order Markov model. *Future Generation Computer Systems 89*, 506 - 514.

[8] Melnyk, S., Usatenko, O., & Yampol'skii, V. 2006. Memory functions of the additive Markov chains: applications to complex dynamic systems. *Physica A: Statistical Mechanics and its Applications*, 405-415.

[9] Sejal, D., Ganeshsingh, T., Venugopal, K. R., Iyengar, S. S., & Patnaik, L. M. 2016. Image Recommendation Based on ANOVA Cosine Similarity. *Procedia Computer Science 89*, 562 - 567.

[10] Trivago. 2019. *Trivago*. URI = https://company.trivago.com/our-story/.