

OPTIMASI REKOMENDASI RANGKING HOTEL MENGGUNAKAN RETRIEVAL MODEL TENSORFLOW

¹Joni Bastian, ²Made Hanindia Prami Swari

Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran
Jawa Timur

Email: ¹19081010071@student.upnjatim.ac.id, ²madehanindia.fik@upnjatim.ac.id

Abstrak. *Beberapa orang memilih hotel untuk liburan mereka namun beberapa diantara mereka kesulitan untuk menentukan hotel mana yang cocok dan bagus untuknya, ada banyak rekomendasi hotel yang merekomendasikan berdasarkan harga termurah atau hotel terpopuler. Namun, faktanya tidak semua orang menginginkan hotel hanya berdasarkan harga termurah atau bahkan berdasarkan popularitas dari sebuah hotel. Beberapa orang ada yang memilih hotel berdasarkan user preferences nya atau berdasarkan lokasi terdekat user dengan sebuah hotel. Optimasi rekomendasi rangking hotel ini menggunakan Retrieval Model dan package TensorFlow untuk memberikan rekomendasi hotel yang sesuai dengan preferences user sehingga output dari rekomendasi hotel setiap user dapat berbeda-beda. Hasil dari penelitian ini akan memudahkan user untuk memilih hotel yang cocok untuknya. Sehingga memudahkan user untuk mengambil keputusan saat memilih hotel untuk liburannya.*

Kata kunci: *Retrieval Model, User, User Preferences, Hotel, Optimasi*

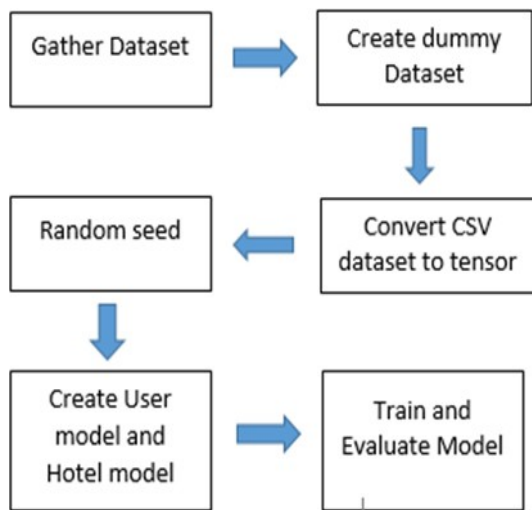
Setelah pandemi Covid-19, industri perhotelan tumbuh karena orang mulai bepergian dan berlibur. Kebanyakan orang mencari hotel yang mereka butuhkan berdasarkan lokasi terdekat, harga termurah, dan kesan pertama dari foto. Namun kebanyakan orang masih bingung dengan pilihannya. Apa yang menyebabkan orang memilih hotel berdasarkan lokasi terdekat dan harga termurah? Kebanyakan mereka ingin mendapatkan harga yang terjangkau, namun terkadang mereka mengesampingkan kualitasnya. Jadi Peneliti memecahkan masalah ini dengan memberikan rekomendasi hotel yang sesuai dengan perilaku pengguna. Peneliti ingin membantu mereka memilih hotel terbaik karena Peneliti tahu harga bukan satu-satunya penentu. Peneliti tidak ingin orang memberikan ulasan/umpan balik negatif untuk hotel karena kesalahan mereka yang memilih hotel hanya berdasarkan harga termurah. Penawaran hotel dengan segala fasilitasnya sedikit banyak membuat wisatawan kesulitan menentukan hotel pilihannya, dimana terdapat berbagai pertimbangan dalam memilih, diantaranya: harga sewa, fasilitas yang disediakan, lokasi, akses ke tempat wisata, dan lain sebagainya[1].

Sistem rekomendasi adalah perangkat lunak bantu dan teknik dalam menyediakan beberapa saran yang akan dimanfaatkan oleh seorang pengguna[2]. Saran-saran tersebut berkaitan dengan proses pengambilan keputusan[3].

Sistem rekomendasi dan tradisional informasi retrieval menggunakan faktor laten linear seperti faktorisasi matriks. Metode Deep Learning sangat berhasil untuk masalah dimensi tinggi lainnya. Seperti NLP, Visi dan Visi Komputer seperti Dacrema Cremonesi, dan Jannach[4].

I. Metodologi

Penelitian ini menggunakan *Retrieval model* dengan menggunakan library Tensorflow. Tahap pertama yaitu menyiapkan dataset. Namun karena sulitnya mendapat dataset yang cocok untuk model ini maka Peneliti membuat dataset dari *scratch*. Selanjutnya dataset yang sudah dimuat kemudian dikonversi menjadi tensor. Kemudian mengacak data dengan `random.set_seed()` agar data yang diacak saat dipanggil akan tetap bernilai sama pada saat diacak. Data yang diacak kemudian dibagi menjadi `user_model` dan `hotel_model`, Selanjutnya train model, fit model dan evaluasi model. Setelah di evaluasi model dapat dilakukan proses testing. Kemudian model sudah dapat di save untuk keperluan deploying menggunakan web service atau langsung ke mobile device.



Gambar 1. Retrieval Model

Dataset

Data yang digunakan pada model ini merupakan data tabular. Data user preferences dan data hotel. Data user yaitu berupa user id, hotel id, review rating user terhadap hotel. Sedangkan data hotel yaitu hotel id, dan data nama hotel. Data hotel yang Peneliti gunakan adalah data hotel yang berada di Singapura.

| user_id | rating | hotelld |
|---------|--------|---------|
| 26 | 5 | 49 |
| 2 | 3 | 82 |
| 15 | 3 | 100 |
| 21 | 5 | 22 |
| 31 | 5 | 88 |

Gambar 2. Dataset user preferences

| hotelld | name |
|---------|-----------------------------|
| 1 | Orchard Hotel Singapore |
| 2 | Sheraton Tower Singapore |
| 3 | M Social Singapore |
| 4 | Swissotel The Stamford |
| 5 | Crowne Plaza CHANGI AIRPORT |

Gambar 3. Dataset hotel

Dummy Dataset

Peneliti cukup kesulitan untuk mendapatkan dataset publik yang cocok untuk sistem rekomendasi ranking hotel ini sehingga Peneliti mencoba untuk membuat dataset dummy untuk model ini. Pada dataset hotel peneliti melakukan *scraping* dataset nama hotel di Singapura. Dataset yang terkumpul menjadi 2 bagian dataset yaitu dataset user *preferences* berjumlah 100 baris dan dataset hotel berjumlah 100 baris.

Konversi Data ke Data Tensor

Konversi dataset menjadi data tensor ini merupakan tahapan awal dari pre-processing data. Tensorflow hanya mendukung tipe data tensor untuk ranking dan retrieval model sehingga sebelum Peneliti dapat menggunakan model perlu mengubah tipe data yang sebelumnya number dan string menjadi tipe data tensor. Method yang digunakan untuk konversi data ini adalah `tf.data.Dataset.from_tensor_slices()`.

Selanjutnya setelah data dikonversi menjadi tensor dilakukan fungsi lambda untuk melakukan *mapping* data

```

ratings = tf.data.Dataset.from_tensor_slices({
    'user_id': df_merged.user_id.to_list(),
    'name': df_merged.name.to_list(),
    'rating': df_merged.rating.to_list()
})
  
```

```

hotels = tf.data.Dataset.from_tensor_slices({
    'name': df_merged.name.to_list()
})
  
```

Gambar 4. Konversi data ke data tensor

```

ratings = ratings.map(lambda x: {
    'user_id': x['user_id'],
    'rating' : x['rating'],
    'name': x['name'],
})
list(ratings.take(5).as_numpy_iterator())
  
```

```

hotels = hotels.map(lambda x: x['name'])
  
```

Gambar 5. Fungsi Lambda

Random Seed

Setelah dataset sudah dikonversi menjadi data tensor tahap selanjutnya adalah mengacak dataset dan ketika dipanggil mendapatkan nilai yang selalu sama saat diacak. Pengacakan dataset ini dilakukan sebelum membuat model. Pada proses ini juga selanjutnya membagi dataset menjadi data train dan data test. Proses pengacakan dan pembagian dataset menggunakan tensorflow

```
tf.random.set_seed(42)
shuffled = ratings.shuffle(100, seed=42, reshuffle_each_iteration=False)

train = shuffled.take(80)
test = shuffled.skip(80).take(20)
```

Gambar 6. Random set seed

Pada proses ini juga selanjutnya menginisiasi `unique_hotels_name` variabel dan `unique_user_ids` variabel sebagai variabel unik yang akan digunakan oleh model untuk merekomendasikan user.

Model

Pada proses ini Peneliti membuat dua model yaitu `user_model` dan `hotel_model` kemudian proses selanjutnya menginisiasikan metrics dan task. Metrics yang Peneliti gunakan adalah `FactorizedTopK`. Metrics ini merupakan metrics yang umum digunakan pada kasus rekomendasi. Proses selanjutnya adalah menginisiasi kelas model `HotelsModel` yang menggunakan `user_model` dan `hotel_model` dengan metrics dan task yang sudah diinisiasi sebelumnya.

```
user_model = tf.keras.Sequential([
    tf.keras.layers.StringLookup(
        vocabulary=unique_user_ids, mask_token=None),
    # We add an additional embedding to account for unknown tokens.
    tf.keras.layers.Embedding(len(unique_user_ids) + 1, embedding_dimension)
])
```

```
hotel_model = tf.keras.Sequential([
    tf.keras.layers.StringLookup(
        vocabulary=unique_hotels_name, mask_token=None),
    tf.keras.layers.Embedding(len(unique_hotels_name) + 1, embedding_dimension)
])
```

```
metrics = tfrs.metrics.FactorizedTopK(
    candidates=hotels.batch(128).map(hotel_model)
)
```

```
task = tfrs.tasks.Retrieval(
    metrics=metrics
)
```

```
class HotelsModel(tfrs.Model):
    def __init__(self, user_model, hotel_model):
        super().__init__()
        self.user_model = tf.keras.Model = user_model
        self.task = tf.keras.layers.Layer = task

    def compute_loss(self, features: Dict[Text, tf.Tensor], training=False) -> tf.Tensor:
        # We pick out the user features and pass them into the user model.
        user_embeddings = self.user_model(features["user_id"])
        # And pick out the movie features and pass them into the movie model,
        # getting embeddings back.
        positive_hotel_embeddings = self.hotel_model(features["name"])

        # The task computes the loss and the metrics.
        return self.task(user_embeddings, positive_hotel_embeddings)
```

```
model = HotelsModel(user_model, hotel_model)
model.compile(optimizer=tf.keras.optimizers.Adagrad(learning_rate=0.1))
```

```
cached_train = train.shuffle(100).batch(64).cache()
cached_test = test.batch(32).cache()
```

Gambar 7. Model

Evaluasi Model

Pada tahap evaluasi model hasil dari proses training model dievaluasi menggunakan metrics `FactorizedTopK` menghitung nilai akurasi dan nilai loss dari data testing.

II. Hasil dan Pembahasan

Data yang digunakan pada penelitian ini menggunakan dua dataset yaitu dataset user *preferences* dan data hotel yang masing-masing berjumlah 100 baris data. Data berjumlah 100 baris tersebut dibagi menjadi 64 dataset training dan 32 dataset testing. Model yang digunakan adalah retrieval model dengan tensorflow. Metrics yang digunakan yaitu FactorizedTopK. Berikut ini hasil prediksi hotel untuk beberapa user.

Gambar 7. Rekomendasi user id 25

Gambar 8. Rekomendasi user id 39

Gambar 9. Rekomendasi user id 105

```
# Get recommendations.
_, titles = index(tf.constant(["25"]))
print(f"Recommendations for user 25: {titles[0, :10]}")

Recommendations for user 25: [b'Hotel 81 Osaka' b'Capri by Fraser China Square'

# Get recommendations.
_, titles = index(tf.constant(["39"]))
print(f"Recommendations for user 39: {titles[0, :10]}")

Recommendations for user 39: [b'The Westin Singapore ' b'ibis budget Singapore Pearl'
b'Vintage Inn Boutique Capsule ' b'Four Chain View Hotel '
b'K Hotel Alival (Premier)' b'Hilton Garden Inn Singapore Serangoon'
b'Changi Cove' b'Wanderlust The Unlimited Collection by Oakwood'
b'Arianna Hotel' b'Hotel 81 Balestier ']
```

III. Kesimpulan

Hasil dari penelitian didapatkan rekomendasi berupa 10 baris hotel untuk user yang berbeda setiap user nya berdasarkan *preferences* user pada review rating hotel masing-masing. User baru juga akan mendapatkan rekomendasi berdasarkan rating dari user-user sebelumnya terhadap suatu hotel. Penelitian lebih lanjut dibutuhkan untuk mendapatkan rekomendasi optimasi hotel ranking lebih baik lagi seperti lokasi user terhadap hotel terdekat dan fitur lainnya.

IV. Daftar Pustaka

- [1] Y. Pristanto, S. Wahyuni, and Y. D. Handini, “Analisis Faktor-faktor Yang Mempengaruhi Keputusan Menginap Pada Hotel Bintang Mulia Jember,” Universitas Jember, 2013.
- [2] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*. New York: Springer, 2010.
- [3] L. Chen, M. D. E. Gemmis, A. Felfernig, P. Lops, F. Ricci, and G. Semeraro, “Human Decision Making and Recommender Systems,” *ACM Trans. Interact. Intell. Syst.*, vol. 3, no. 3, pp. 1–7, 2013.
- [4] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. “Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches”. In: *Proceedings of the 13th ACM Conference on Recommender Systems. RecSys '19*. Copenhagen, Denmark: ACM, 2019, pp. 101–109. ISBN: 978-1-4503-6243-6.