

Implementasi *Convolutional Neural Network* dan *Probabilistic Matrix Factorization* pada Sistem Rekomendasi Buku

Zaki Mudzakir Hidyatullah¹, Dade Nurjanah², Rita Rismala³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹zakimudzakir@students.telkomuniversity.ac.id, ²dadenurjanah@gmail.com, ³rismala.rita@gmail.com

Abstrak

Sistem Rekomendasi dapat merekomendasikan buku pada *user* tertentu berdasarkan prediksi *rating*, isi konten buku, ataupun metode lainnya. Banyak metode *recommendation system* yang digunakan seperti *Probabilistic Matrix Factorization*, dimana konten yang sudah diberi *rating* akan sering direkomendasikan. Namun pada *Probabilistic Matrix Factorization* memiliki kekurangan yaitu dalam mengatasi data yang memiliki nilai *rating* yang jarang. Maka diperlukan suatu metode yang digunakan untuk memahami konteks isi dari buku sehingga tidak hanya melihat dari *rating* saja namun dilihat juga dari *review* suatu buku. Untuk mempelajari *review* maka digunakan suatu metode yaitu *Convolutional Neural Network* dengan cara memberikan suatu nilai vektor yang mengarah terhadap konteks buku kepada *Probabilistic Matrix Factorization* suatu *recommender system*. Berdasarkan hasil pengujiannya, metode tersebut dapat meningkatkan keakuratan data dengan MAE = 3,0114707. Sedangkan untuk *Probabilistic Matrix Factorization* nilai MAE = 4,0185377. Dari nilai tersebut dapat dijelaskan bahwa metode *Convolutional Neural Network* dan *Probabilistic Matrix Factorization* bekerja cukup baik untuk data yang jarang memiliki *rating*.

Kata kunci : *recommender system, Convolutional Neural Network, Probabilistic Matrix Factorization*

Abstract

The Recommendation System can recommend books to certain users based on rating predictions, book content, or other methods. Many system recommendation methods are used such as *Probabilistic Matrix Factorization*, where content that has been rated will often be recommended. However, the *Probabilistic Matrix Factorization* has the disadvantage of overcoming data that has a rare rating value. So we need a method used to understand the context of the contents of the book so that it is not only seen from the rating but also seen from a book review. To study the review, a method called *Convolutional Neural Network* is used by giving a vector value that leads to the context of the book to the *Probabilistic Matrix Factorization* of a recommender system. Based on the test results, this method can improve the accuracy of the data with MAE = 3.0114707. As for the *Probabilistic Matrix Factorization* the MAE= 4.0185377. From these values it can be explained that the *Convolutional Neural Network* and *Probabilistic Matrix Factorization* methods work well enough for data that rarely has a rating.

Keywords: Recommender system, Probabilistic Matrix Factorization, Convolutional Neural Network

1. Pendahuluan

1.1. Latar Belakang

Sistem rekomendasi merupakan suatu sistem yang memudahkan penggunaannya dalam menentukan suatu produk atau barang yang akan dipilih atau dibeli [1]. Karena semakin banyaknya produk-produk yang tersedia di dunia ini, dengan menggunakan sistem rekomendasi maka pengguna akan lebih mudah memilih produk yang disukai. Sistem rekomendasi berperan dalam memprediksi keinginan pengguna terhadap suatu produk [1].

Salah satu metode yang sering digunakan dalam membuat suatu rekomendasi adalah *Probabilistic Matrix Factorization*. Pada *Probabilistic Matrix Factorization* biasanya menggunakan data dari *rating* atau ulasan yang sudah diberikan pengguna lain [3]. Kemudian pengguna baru akan ditentukan kemiripannya dari pengguna lain. Setelah itu sistem dapat memberikan rekomendasi buku kepada pengguna baru tersebut. Namun, *Probabilistic Matrix Factorization* memiliki kekurangannya yaitu metode *Probabilistic Matrix Factorization* akan mengalami kesulitan apabila dataset yang diambil memiliki skalabilitas yang tinggi dan apabila dataset yang diambil memiliki data yang jarang muncul [2] [4] [5].

Untuk mengatasi kekurangan dalam dataset yang jarang muncul, maka digunakan model lain dengan menggabungkan metode *Convolutional Neural Network* dan *Probabilistic Matrix Factorization* [2]. *Convolutional Neural Network* berperan untuk memahami makna suatu *review text* yang diberikan nilai vektor yang digunakan untuk dapat memahami suatu isi konteks dalam suatu buku. Setelah dilalui proses *Convolutional Neural Network*, maka selanjutnya melakukan proses *Probabilistic Matrix Factorization*.

Probabilistic Matrix Factorization berperan dalam merepresentasikan suatu matriks yang dimana mewakili setiap *user* dan buku dan memprediksi nilai *rating* dari karakteristik *user* dan buku [3].

1.2. Topik dan Batasannya

Sistem rekomendasi dengan menggunakan *Probabilistic Matrix Factorization* sering digunakan namun terkadang ketika data memiliki *sparsity* yang tinggi [1] [4]. Sehingga metode *Probabilistic Matrix Factorization* menjadi sulit memberikan rekomendasi maksimal sehingga dibutuhkan suatu metode yang dapat menangani *sparsity* tersebut dengan cara mengambil -- dari deskripsi buku tersebut kemudian menganalisa deskripsi menggunakan CNN untuk menemukan nilai-nilai vektor yang akan mempengaruhi suatu *rating* pada suatu buku [2]. Setelah hasil dari *review* buku tersebut kemudian nilai vektor tersebut digunakan dalam uji kesamaan buku untuk mendapat nilai *rating* dari buku yang memiliki konteks yang sama dan dilanjutkan dengan metode PMF yang akan memberikan hasil prediksi *rating*. Maka dengan demikian akan membantu menyelesaikan masalah data *sparsity* pada suatu sistem rekomendasi. Dengan menggabungkan metode CNN dan PMF maka metode ini dinamakan sebagai *Convolutional Matrix Factorization* [2]. Kemudian setelah itu akan dilakukn pengujian dengan menggunakan *Mean Absolute Error* (MAE) untuk mengukur kesalahan tersebut.

Terdapat batasan yang ada pada tugas akhir ini yang diantaranya dataset yang digunakan menggunakan dataset dari *amazon*. Pada dataset tersebut tidak seluruhnya diambil, hanya diambil sebagian saja yaitu diambil 320.000 *rating*. Dan user yang memberikan *rating* minimal 30 buku. Kemudian kata *review* yang diambil setiap buku paling maksimal 8000 kata saja.

1.3. Tujuan

Tujuan dari tugas akhir ini adalah untuk mengimplementasikan metode *Convolutional Neural Network* dan *Probabilistic Matrix Factorization* pada sistem rekomendasi buku dan kemudian membandingkannya dengan *Probabilistic Matrix Factorization* dengan menggunakan *Mean Absolute Error* terhadap dataset yang digunakan.

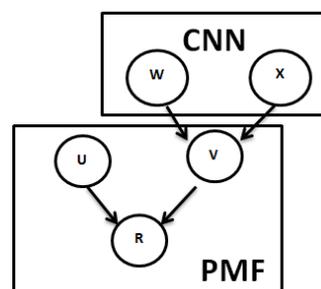
1.4. Organisasi Tulisan

Pada organisasi tulisan tugas akhir ini dibagi kedalam beberapa bagian, dimulai dari studi-studi terkait yang menjelaskan teori-teori pada tugas akhir ini, tinjauan pustaka, serta referensi-referensi yang mendukung, kemudian menjelaskan rancangan sistem yang dibangun. Dan dilakukan evaluasi dari hasil pengujian. Kemudian diberikan kesimpulan dan saran.

2. Studi Terkait

Terdapat banyak metode yang sudah pernah digunakan terkait sistem rekomendasi yang diantaranya ada metode Collaborative Filtering yang dimana Collaborative Filtering merupakan bagian dari Probabilistic Matrix Factorization [4]. Dan dijelaskan juga bahwa metode Collaborative Filtering memiliki kekurangan dalam *sparsity* data atau data yang jarang jadi terkadang menyulitkan juga dalam memprediksi nilai *rating* pada sistem rekomendasi [3]. Untuk mencegah terjadinya *sparsity* data maka dilakukanlah dengan menganalisa *review* buku menggunakan metode Convolutional Neural Network yang dimana dilakukan data word embeddingnya merupakan hasil training dari GloVe Word Representation [6][7][8][9]. Dari hasil Convolutional Neural Network tersebut menghasilkan nilai vektor yang merupakan suatu konteks dari buku kemudian setelah itu digunakan untuk menguji kemiripan setiap buku. Jika buku itu memiliki kemiripan yang sama, maka diberi *rating* yang sama dengan buku yang sama dan sudah diberi *rating*. Setelah itu dilakukanlah metode Probabilistic Matrix Factorization untuk memprediksi *rating*-*rating* tersebut [2][7].

Dengan menggabungkan Convolutional Neural Network dan Probabilistic Matrix Factrozation dapat mengurangi terjadinya kejarangan data. Dapat dilihat seperti pada gambar 1.



Gambar 1 Convolutional Neural network dan Probabilistic Matrix Factrization [7].

Pada Gambar 1 tertulis nilai W merupakan nilai *rating* buku, nilai X merupakan informasi vektor buku, nilai V merupakan variabel buku, nilai U merupakan variabel *user* nilai R merupakan kolaborasi variabel *user*

dan variabel buku. Pada proses *Convolutional Neural Network* dilakukan untuk memperoleh vektor buku dari *review text*. Setelah itu dilakukan *cosine similarity* untuk mendapatkan kemiripan setiap buku agar *rating* pada buku disamakan apabila nilai kemiripan buku dari vektor buku lebih besar dari 80%. Sehingga diperoleh variabel buku yang berupa nilai *rating* yang baru dari buku terhadap *user*. Setelah itu dilakukan *Probabilistic Matrix Factorization* untuk memperoleh rekomendasi buku dari variabel *user* dan variabel buku.

2.1. GloVe Word Representation

GloVe Word Representation merupakan representasi kata untuk menghasilkan *word embeddings* yang dibuat oleh Stanford University. *GloVe Word Representation* menggunakan data korpus dari Wikipedia yang kemudian dikumpulkan teks yang hasilnya dibangun *vocabulary* setiap kata dan setiap katanya menghasilkan vektor yang nilainya berjumlah lebih dari 50 dimensi [6]. Yang dimana nilainya adalah berupa peluang kemunculan kata terhadap seputar kata-kata.

Pada tugas akhir ini dimensi GloVe Word Representation yang digunakan adalah sebesar 300 dimensi. Yang dimana 300 dimensi ini lebih akurat dibandingkan dimensi yang lainnya [6].

2.2. Convolutional Neural Network

Tujuan dari Convolutional Neural Network pada sistem rekomendasi buku disini adalah untuk menemukan nilai vektor dari suatu buku. metode dalam membuat Convolutional Neural Network dalam menemukan nilai vektor dari suatu buku terdiri 4 lapisan yaitu *Embedding Layer*, *Convolution Layer*, *Pooling Layer*, *Output Layer* [7].

2.2.1. Embedding Layer

Pada lapisan *Embedding Layer* bertujuan untuk mengubah dokumen *review text* buku menjadi nilai matriks numerik yang mewakili dokumen untuk *Convolution Layer* berikutnya.

Dokumen kata secara acak telah diinisialisasi dari model *embedding* yang sudah dilatih sebelumnya oleh *GloVe Word Representation* [6]. Kemudian vektor-vektor kata tersebut di optimasi berdasarkan dokumen dengan $D \in R^{p \times l}$ dokumen menjadi

$$D = [\dots | w_{i-1} | | w_i | | w_{i+1} | \dots]$$

yang dimana l merupakan panjang dari dokumen dan p adalah ukuran matriks embedding untuk setiap kata w_i [7].

2.2.2. Convolution Layer

Pada tugas akhir ini menggunakan Arsitektur *Convolution* untuk menganalisa kalimat pada setiap dokumennya. Sehingga setelah didapat *output* dari *Embedding Layer* berupa matriks nilai ukuran setiap dokumen. Langkah selanjutnya melakukan analisa setiap kalimat pada dokumen untuk mendapatkan ekstraksi dari dokumen tersebut. Untuk menganalisa dokumen dengan benar, maka suatu kontekstual ekstraksi [9].

Model Convolution layer memberikan fitur konteks dari ekstraksi proses *embedding layer* [7] [8] [9] yang ditentukan dari:

$$c_c^j = f(W_c^j * D_{(c,i:(i+ws-1))} + b_c^j)$$

Yang dimana:

c_c^j : convolution layer

* : convolution operator

D: dokumen buku

W_c^j : hasil dari word embedding

b_c^j : merupakan nilai bias

f : fungsi aktivasi non linear.

Pada tugas akhir ini fungsi aktivasi non linear yang digunakan yaitu ReLU (*Retrieved Linear Unit*) untuk mencegah terjadinya *vanishing gradient* [11]. Hasil dari *Convolution layer* berupa nilai matriks ekstraksi dari proses *embedding layer* [10].

2.2.3. Pooling Layer

Pooling Layer mengekstraksi fitur-fitur representatif dari *Convolution layer*, dan juga berkaitan dengan

panjang variabel dokumen via operasi penyatuan yang membangun vektor fitur dengan panjang tetap. *Pooling layer* pada tugas akhir ini menggunakan *max pooling* [7]. Yang dimana *max pooling* dilakukan untuk mendapatkan nilai maksimal dari matriks *convolution layer*. Dokumen direpresentasikan sebagai vektor fitur kontekstual, dimana setiap vektor fitur kontekstual memiliki panjang variabel. Dan tentu mengalami 2 masalah yaitu panjang vektor kontekstual fitur yang bervariasi dan jika vektor memiliki nilai variasi yang banyak akan mengurangi performansi sistem [7] [8] [9].

2.2.4, Output Layer

Output Layer merupakan fitur yang diperoleh dari lapisan sebelumnya yang harus dikonversi untuk tugas tertentu. Jadi, pada hasil dari *Pooling Layer* memproyeksikan d_f pada ruang k -dimensional dan buku untuk tugas rekomendas, yang akhirnya menghasilkan dokumen laten vektor dengan menggunakan proyeksi nonlinier konvensional.

2.3. Cosine Similarity

Cosine Similarity merupakan perhitungan untuk mengecek seberapa besar kemiripan dari 2 buku melalui nilai vektor yang dimiliki 2 buku tersebut dengan perhitungan sebagai berikut [4].

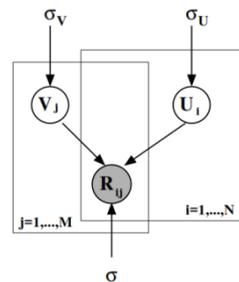
$$\text{Similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Yang dimana A merupakan nilai matriks A dan B merupakan nilai matriks B.

2.4. Probabilistic Matrix Factorization

Probabilistic Matrix Factorization (PMF) adalah suatu model probabilistik linear dengan model observasi *Gaussian* [4]. PMF bekerja cukup baik pada dataset menghitung *rating* dan seimbang.

Pada PMF, bentuk rangkaiannya adalah dari kedua fitur-fitur vektor yaitu *user* dan *item* digabungkan untuk menentukan nilai *rating*. Cara menggabungkannya dapat dilihat di gambar 2 berikut:



Gambar 2. Probabilistic Matrix Factorization [4].

Yang dimana nilai U_i merupakan hasil *given* dari nilai σ_U yang digabung dari data *rating user*. dan nilai V_j merupakan hasil *given* dari nilai σ_V yang digabung dari data *rating item* [4].

Probabilistic Matrix Factorization memprediksikan evaluasi buku kepada pengguna khusus berdasarkan *rating* sebelumnya dari pengguna yang mempunyai ketertarikan yang mirip [3]. Yang dimana apabila pengguna *user1* untuk buku *book1* mirip dengan pengguna lain *user2* apabila *user1* dan *user2* telah memberikan *rating* yang mirip untuk buku-buku yang lain. Untuk menemukan untuk menghitung nilai prediksi adalah

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) x P_{a,u}}{\sum_{u=1}^n P_{a,i}}$$

Yang dimana nilai $p_{a,i}$ merupakan nilai prediksi *user* a terhadap buku i. $P_{a,u}$ merupakan nilai *similarity* antara *user* a terhadap buku u. Sedangkan nilai n merupakan jumlah *user*.

2.5. Mean Absolute Error

Mean Absolte Error digunakan pada tugas akhir ini untuk mengukur keakuratan data prediksi dengan data yang sebenarnya. Kemudian mengubah preferensi yang diprediksi menjadi skor prediksi. Mean Absolute error dihitung dengan [12]

$$MAE = \frac{\sum_{i=1}^N \sum_{j=1}^M \sum_{r_{ij} \in T_i} |r_{ij} - \hat{r}_{ij}|}{\sum_{i=1}^N |T_i|}$$

yang dimana nilai \hat{r}_{ij} dihitung dengan

$$\hat{r}_{ij} = r_{a,i} + \frac{\sum_{k=1}^N \sum_{k \neq j} [Sim(u_i, u_k) \times (r_{kj} - r_{a,k})]}{\sum_{k=1}^N \sum_{k \neq j} Sim(u_i, u_k)}$$

yang dimana \hat{r}_{ij} merupakan skor prediksi buku b_j dari user u_i , $r_{a,i}$ adalah nilai rata-rata rating pada himpunan yang user u_i beri rating pada buku, r_{kj} merupakan nilai rating user u_i terhadap buku b_j pada himpunan dan $Sim(u_i, u_k)$ merupakan nilai cosine similarity yang merupakan nilai kemiripan user u_i terhadap user u_k .

3. Sistem yang Dibangun

3.1 Dataset

Dataset yang digunakan pada tugas akhir ini adalah dataset dari Amazon khusus untuk review buku saja. Terdapat berbagai rating pada suatu buku dan juga review text dari para user.

Tabel 1. Jumlah Data Amazon

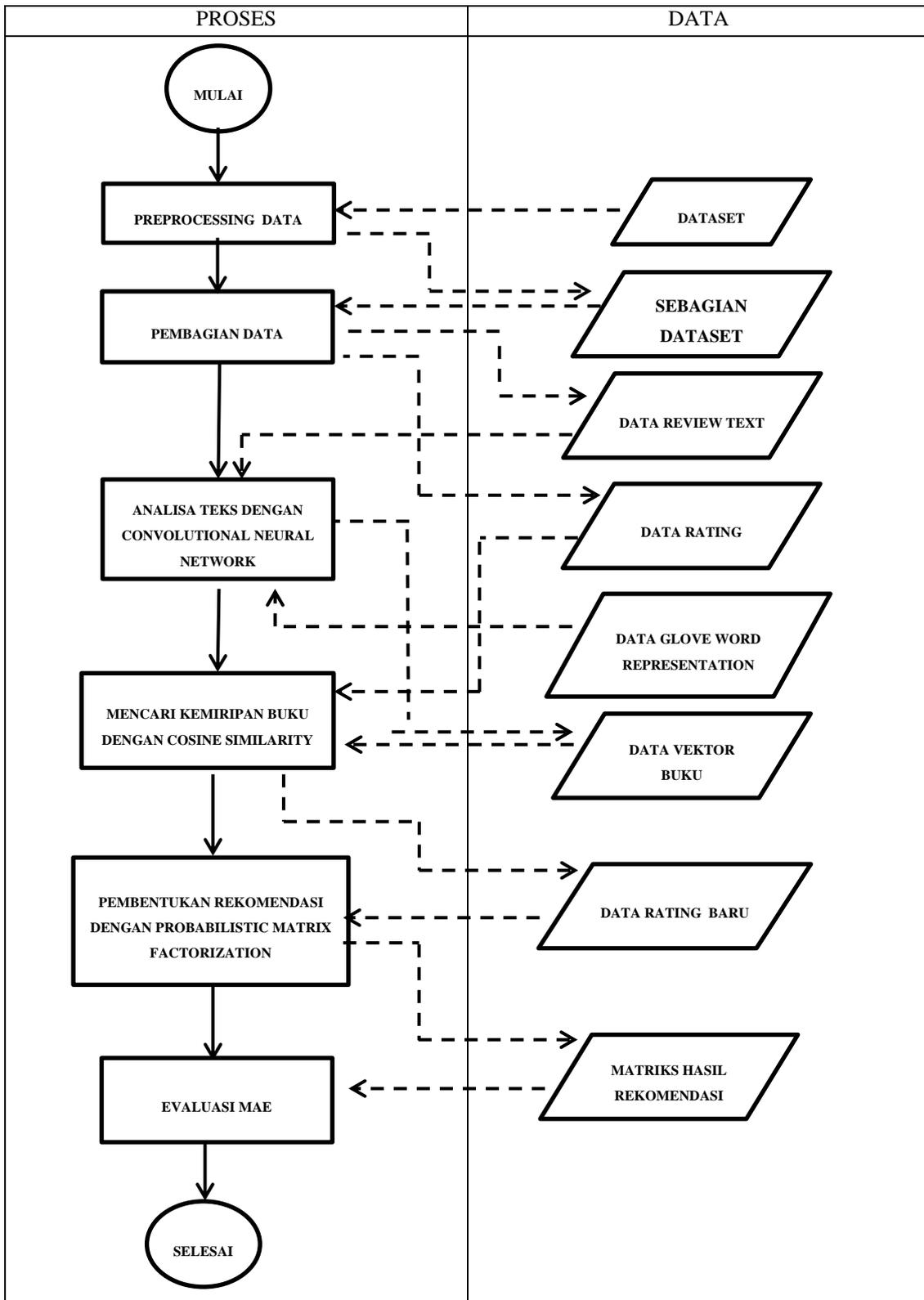
Total data rating	22.507.155
Total data buku	2.330.066
Total data user	8.026.324

Jumlah data yang terlalu besar sehingga penulis membatasinya dengan mengambil 320.000 data rating kemudian setelah itu dilakukan pemngambilan data dari user yang mengambil lebih dari 30 buku.

Tabel 2. Jumlah Data yang telah disaring dari Amazon

Total data rating	27484
Total data buku	6593
Total data user	493

3.2 Deskripsi dan Analisis Sistem



Gambar 3. Gambaran umum sistem

3.2.1. Preprocessing Data

Pada *Preprocessing Data* dilakukan pengambilan sebagian data dari dataset yaitu berjumlah 320.000 data *rating* dengan bentuk data sebagai berikut:

Tabel 3. Bentuk data

Bentuk data
{ "reviewerID": "A10000012B7CGYKOMPQ4L", "asin": "000100039X", "reviewerName": "Adam", "helpful": [0, 0], "reviewText": "Spiritually and mentally inspiring! A book that allows you to question your morals and will help you discover who you really are!", "overall": 5.0, "summary": "Wonderful!", "unixReviewTime": 1355616000, "reviewTime": "12 16, 2012" }

Pada tabel 3, bentuk data tersebut merupakan data dengan format JSON yang diantaranya ada reviewerID merupakan id *user*, asin merupakan id buku, reviewerName merupakan nama *user*, *reviewText* merupakan teks *review* buku, overall merupakan nilai *rating* dengan *range* 1-5. *Preprocessing Data* menghasilkan sebagian dataset dari dataset yang sebelumnya dengan jumlah 320.000 data *rating*.

3.2.2. Pembagian Data

Setelah didapatkan sebagian dataset tersebut kemudian data tersebut diambil dengan dipilih berdasarkan jumlah buku yang diberi *rating* oleh *user* harus lebih besar atau sama dengan 30 [7].

Setelah itu dilakukan pembagian data menjadi 2 data yaitu data *rating* seperti pada tabel 4 dan data *review text* seperti pada tabel 5.

Tabel 4. Data *rating*

Data <i>rating</i> (userId::bookId::rating)
A10000012B7CGYKOMPQ4L::000100039X::5.0
A2S166WSCFIFP5::000100039X::5.0
A1BM81XB4QHOA3::000100039X::5.0
A1MOSTXNIO5MPJ::000100039X::5.0
...
...
A38VFYFUTYJ0T5::006178320X::2.0
A3MT311233OHMQ::006178320X::5.0
A3MOP2UZZ84GWE::006178320X::5.0
A358DZNR8H86MM::006178320X::4.0
A3ETEBL6SRZYOZ::006178320X::5.0
A21RBPA7OMEHQ5::006178320X::5.0

Tabel 5. Data *review text*

Data <i>review text</i> (bookId::reviewText)
000100039X::Spiritually and mentally inspiring! A book that allows you to question your morals and will help you discover who you really are! This is one my must have books. It is a masterpiece of spirituality. I'll be the first to admit, its literary quality isn't much....
...
...
0061780286::I absolutley LOVED all these short stories, which all tie in together,, but my absolute favorite was the 1st one which was oh so very HOT!! I have finally found an author who writes erotic romance exactly as I have been looking for...

Pada bagian Data *review text* dilakukan analisa kata dengan mengambil paling maksimal 8000 kata dari setiap buku dengan menggunakan *Convolutional Neural Network*. Sehingga pada Data *review text* terdapat id buku dan *review text* buku tersebut.

3.2.3. Analisa teks dengan Convolutional Neural Network

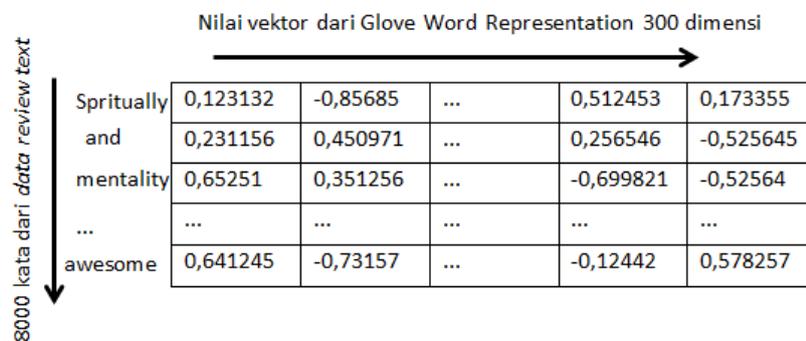
Pada proses *Convolutional Neural Network* terdapat proses *embedding layer* pada bab 2.2.3 [7]. Proses tersebut memerlukan *word embedding* dari *GloVe Word Representation* yang bentuk datanya seperti tabel 6.

Tabel 6. Data GloVe Word Representation

Data GloVe Word Representation	
the	0.04656 0.21318 -0.0074364 -0.45854 -0.035639 0.23643 -0.28836 0.21521 -0.13486 -1.6413 ... -0.33427 0.011807 0.059703
and	0.038466 -0.039792 0.082747 -0.38923 -0.21431 0.1702 -0.025657 0.09578 0.2386 -1.6342 ... -0.53427 0.02756807 0.023702
...	
...	
nightmask	0.60137 0.11643 0.25629 -0.069297 -0.18242 -0.20173 0.24848 -0.37567 0.28478 ... 0.1599 0.62007 0.22581

Dimana setiap katanya memiliki skor berupa vektor yang sudah dilatih oleh Stanford University [6]. Pada tugas akhir ini data *GloVe Word Representation* yang digunakan adalah 300 dimensi yang artinya setiap kata memiliki 300 nilai vektor.

Setelah diambil data *Glove Word Representation* tersebut dipilih setiap kata yang ada pada data *review text* untuk membangun suatu metriks dengan panjang 300x8000. Contohnya seperti pada gambar 4.



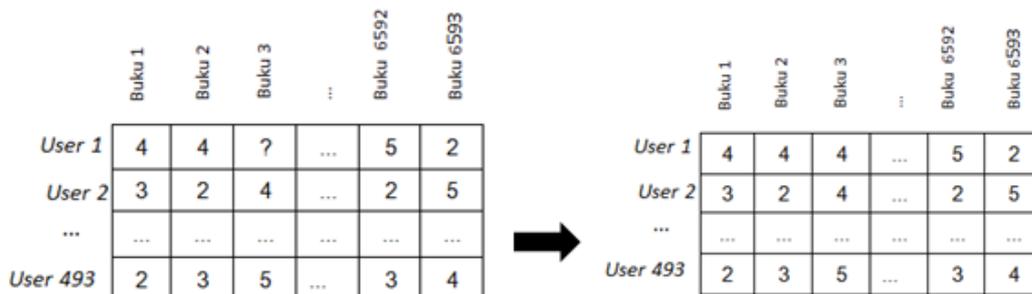
Gambar 4. Metriks embedding layer

Metriks *embedding layer* tersebut diolah kedalam *convolution layer*. *Convolution layer* ini untuk mengekstraksi data *embedding layer*. Pada tugas akhir ini menggunakan *Convolution layer* 2D untuk mengekstraksinya.

Hasil dari *Convolutional Neural Network* tersebut didapatkan nilai vektor setiap buku yang dimana nilai vektor buku tersebut dilakukan pencocokan kemiripannya dengan *Cosine Similarity*. Jika nilai *Cosine Similarity* mendapatkan nilai lebih besar dari 0,8. Maka buku tersebut dinyatakan memiliki kemiripan dan diberi rating sama dengan buku tersebut.

3.2.4. Mencari kemiripan buku dengan *cosine similarity*

Data vektor buku tersebut kemudian dihitung kemiripannya dengan *cosine similarity* pada setiap buku, jika buku tersebut memiliki nilai lebih dari 0,8. Maka, buku yang tidak memiliki *rating* pada *user* tertentu diberi *rating* sesuai buku yang sudah memiliki *rating* pada *user* tersebut. Hasilnya mendapatkan data *rating* yang baru untuk buku yang memiliki kesamaan pada buku tersebut. Sebagai contoh dapat dilihat pada gambar 5. Jika buku 1 sama dengan buku 3 dengan *cosine similarity* lebih besar 0,8 dan buku 3 tersebut belum memiliki *rating* oleh *user* 1, maka buku 3 pada *user* 1 tersebut diberi *rating* yang sama dengan buku 1 pada *user* 1.



Gambar 5. Ilustrasi mencari kemiripan buku dengan *cosine similarity*

Hasil dari *cosine similarity* tersebut menghasilkan nilai matriks yang memiliki data *rating* baru.

3.2.5. Pembentukan Rekomendasi Dengan Probabilistic Matrix Factorization

Pada data *rating* baru kemudian dilakukan proses *Probabilistic Matrix Factorization* untuk menemukan skor setiap buku pada masing-masing *user*. Proses ini lah yang digunakan untuk memberikan hasil rekomendasi buku untuk setiap *user*.

3.2.6. Evaluasi MAE

Pengujian ini dilakukan untuk mengetahui baiknya akurasi yang diberikan dengan membandingkan *Probabilistic Matrix Factorization* dengan *Convolutional Neural Network* dan *Probabilistic Matrix Factorization*. Untuk membandingkannya dibuat 2 sistem yang berbeda yaitu sistem untuk *Probabilistic Matrix Factorization* dan sistem untuk *Convolutional Matrix Factorization*. Pengukurannya dengan menggunakan *Mean Absolute Error* yang perhitungannya seperti pada bab 2.6

3.3. Lingkungan Operasional

Lingkungan operasional ini merupakan suatu lingkungan perangkat lunak dan perangkat keras yang digunakan dalam penelitian sistem ini.

3.3.1. Preprocessing Data

Spesifikasi perangkat keras pada penelitian sistem ini:

1. *PC Processor Core-i5 4500k*
2. *Random Access Memory (RAM) 8 GB*
3. *Solid State Drive (SSD) 256 GB*

3.3.1. Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak pada penelitian sistem ini:

1. Sistem Operasi: *macOS 10.13.16*
2. Bahasa Pemrograman *Python 3.7*
3. *Library* pada pemrograman *Python 3.7*:
 - a) *Pandas*: Struktur data tabel
 - b) *Scipy*: *Scientific Computing*
 - c) *Numpy*: Struktur data *array* banyak dimensi dengan fungsi numerik matematika
 - d) *Tensorflow*

4. Evaluasi

4.1. Proses Pengujian

Pada Proses pengujian dilakukan dengan membagi sebagian data yang besar menjadi 320.000 data *rating* kemudian dari data tersebut disaring dengan mengambil jumlah *user* yang memberikan *rating* lebih dari 30 buku. Setelah itu data tersebut dibagi menjadi 2 yaitu data *review text* dan data *rating*. Pada data *rating* tersebut untuk dilakukan pengujian maka dibagi menjadi 2 bagian yaitu data training dan data testing. Untuk pembagian datanya menjadi 80% data training dan 20 data testing dari setiap *user*.

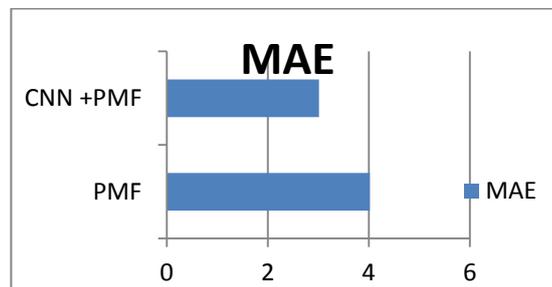
Pada data *review text* dilakukan training data dengan menggunakan *Convolutional Neural Network* dengan dibantu pada proses *word embedding* menggunakan *GloVe Word Representation*. Dan kemudian *Convolutional Neural Network* menghasilkan vektor setiap buku. Pada vektor tersebut dicari kemiripan setiap buku dengan menggunakan *Cosine Similarity*. Hasil dari *cosine similarity* tersebut diambil dengan nilai *rating* yang sama apabila kemiripan buku tersebut bernilai lebih dari 0,8. Setelah menemukan nilai yang sama, dilakukan proses *Probabilistic Matrix Factorization* yang digunakan untuk memprediksi nilai *rating* buku terhadap *user*. Hasilnya dianalisa dengan menggunakan *Mean Absolute Error* yang dibandingkan dengan hanya menggunakan *Probabilistic Matrix Factorization* biasa.

4.2. Tujuan Pengujian

Tujuan pengujian ini adalah untuk menerapkan metode *Convolutional Neural Network* yang digabungkan dengan metode *Probabilistic Matrix Factorization* pada sistem rekomendasi buku. Hasil dari metode tersebut dibandingkan dengan *Probabilistic Matrix Factorization* dengan menggunakan *Mean Absolute Error*.

4.3. Hasil Pengujian

Hasil pengujian dari penelitian ini adalah hasil data yang didapatkan lalu diuji dengan *Mean Absolute Error* untuk melihat besar presentasi kesalahannya dan melihat presentase kebenaran data. Hasil dari *Mean Absolute Error* dapat dilihat sebagai berikut:



Gambar 6. Hasil Pengujian

4.4. Analisis Hasil Pengujian

Pada tahapan *Convolutional Neural Network* didapatkan nilai vektor dengan 50 dimensi yang kemudian diambil kemiripan vektor tersebut dengan menggunakan *Cosine Similarity*. Dan pada data *Amazon* yang diambil pada tugas akhir ini terdapat beberapa buku yang memiliki kemiripan dengan buku yang lainnya.

Pencarian kemiripan ini untuk memberikan *rating* yang sama apabila *user* tidak memberikan *rating* pada buku tersebut dan apabila buku tersebut memiliki kemiripan dengan buku yang diberi *rating* oleh *user*.

Hasil nilai MAE yang diperoleh dapat diketahui bahwa ada beberapa hal yang mempengaruhi nilai MAE yaitu banyaknya jumlah *rating* pada masing-masing *user*. Jika jumlah *rating* buku semakin besar, maka nilai MAE semakin kecil dan ini menandakan tingkat keakuratan data semakin akurat.

5. Kesimpulan

Metode *Convolutional Neural Network* yang digabungkan dengan *Probabilistic Matrix Factorization* berhasil diterapkan pada sistem rekomendasi buku. Dari perbandingan yang didapatkan dengan menggabungkan *Convolutional Neural Network* dan *Probabilistic Matrix Factorization* lebih baik dibanding hanya dengan *Probabilistic Matrix Factorization* saja. Data *Mean Absolute Error* yang ditunjukkan untuk *Convolutional Neural Network* dan *Probabilistic Matrix Factorization* adalah 4,0185378 sedangkan *Probabilistic Matrix Factorization* 3,0114707.

Daftar Pustaka

- [1] Khalid, Haruna et. all. 2017. *A collaborative approach for research paper recommender system*. USA: PLOS One.
- [2] Salakhutdinov, R., & Mnih, A. 2007. *Probabilistic Matrix Factorization*. Canada: Dept. of Computer Science, University of Toronto.
- [3] Fusi, Nicolo, Sheth, Rishit, Elibol, Huseyn Melih. 2017. *Probabilistic Matrix Factorization for Automated Machine Learning*. USA: Dept. of Computer Science, , Tufts University.
- [4] Salehi, M., Kamalabadi, I. N., Ghouschi, M. B. G. 2012. *Personalized recommendation of learning material using sequential pattern mining and attribute based collaborative filtering*. New York: Springer Science.
- [5] Alluhaidan, Ala. 2013. *Recommender System Using Collaborative Filtering Algorithm*. Grand Rapids: Grand Valley State University.
- [6] J. Pennington, R. Socher, and C. D. Manning. 2014. *Glove: Global vectors for word representation*. In *Empirical Methods in Natural Language Processing (EMNLP)*. Stanford: Computer Science Department, Stanford University.
- [7] Kim, Donghyun et. all. 2016. *Convolutional Matrix Factorization for Document Context-Aware Recommendation*. Pohang: Pohang University of Science and Technology.
- [8] L. Quoc, M. Tomas. 2014. *Distributed Representations of Sentences and Documents*. Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043.
- [9] Kim, Yoon. 2014. *Convolutional Neural Networks for Sentence Classification*. New York: New York University.
- [10] J. Alon, S. Oren Sar, G. Yoav. 2018. *Understanding Convolutional Neural Networks for Text Classification*. Israel: Computer Science Department, Bar Ilan Univesity.
- [11] A. L. Maas, A. Y. Hannun, A. Y. Ng. 2013. *Rectifier Nonlinearities Improve Neural Network Acoustic Models*. Stanford: Computer Science Department, Stanford University.
- [12] Chai, T., Draxler, R. R. 2014. *Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literatur*. USA: NOAA Air Resources Laboratory (ARL), NOAA Center for Weather and Climate Prediction, 5830 University Research Court, College Park, MD 20740.