ANALISIS SEMANTIC SIMILARITY PADA ITEM BASED RECOMMENDER SYSTEM

Warih Maharani;Yanuar Firdaus AW

Departemen Teknik Informatika
Institut Teknologi Telkom
Bandung, Indonesia
E-mail: {rani, yanuar}@stttelkom.ac.id

Abstraksi

Recommender system *merupakan sebuah program yang dapat digunakan untuk memprediksi sebuah item berdasarkan informasi yang diperoleh dari user[1].Collaborative filtering merupakan allgoritma yang telah banyak digunakan dalam melakukan proses filtering. Paper ini menjelaskan tentang analisis akurasi prediksi yang diperoleh dari recommender system berdasarkan perbandingan training set dengan test set, ukuran neighborhood, ukuran model, serta nilai variabel α sebagai parameter dalam menghitung similarity. Hasil pengujian menunjukkan bahwa akurasi prediksi yang dihasilkan oleh algoritma* ***adjusted cosine similarity*** *dan* ***semantic similarity*** *relatif lebih lebih rendah jika dibandingkan dengan* ***adjusted cosine similarity*** *tanpa* ***semantic similarity*** *.*

Keywords: *Recommender system, semantic similarity, adjusted cosine similarity.*

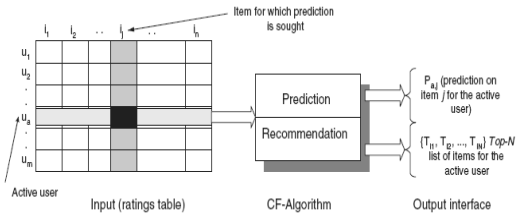
1. PENDAHULUAN

Recommender System adalah sebuah sistem yang menyediakan rekomendasi-rekomendasi mengenai hal-hal yang diinginkan dan sesuai dengan profil penggunanya. Informasi yang diberikan oleh user dapat diperoleh secara eksplisit maupun implisit. Untuk melakukan proses filtering, digunakan algoritma *Collaborative Filtering* (CF) yang telah banyak digunakan dalam proses filtering. Berdasarkan pendekatannya, proses CF dibagi 2, yaitu *model-based CF* dan *memory-based CF*. Dalam *memory-based CF,* prediksi dan rekomendasi dihasilkan dengan mencari kesamaan pada matriks user-item[10]. Sedangkan pada *model-based CF,* prediksi dan rekomendasi dihasilkan dengan membuat sebuah model yang berisi *rating* dari user. Dari model tersebut, akan dihitung *similarity* dari tiap-tiap elemennya. Salah satu contoh *model-based CF* adalah *item-based CF.*

Paper ini membahas tentang analisis *semantic similarity* dengan menggunakan i*tem-based CF*. Tujuannya adalah untuk menganalisis pengaruh penggunaan algoritma *semantic similarity* dan *adjusted cosine similarity* terhadap kualitas rekomendasi yang dihasilkan berdasarkan pengaruh ukuran model, algoritma *similarity* serta ukuran *training set* terhadap hasil rekomendasi.

2. GAMBARAN SISTEM

Pada bagian ini akan dijelaskan tentang i*tem-based CF* , algoritma *adjusted cosine similarity* serta algoritma *semantic similarity*. Proses CF yang dilakukan, digambarkan sebagai berikut :



Gambar 0.1 Proses collaborative Filtering

Proses collaborative filtering terdiri dari prediksi dan rekomendasi yang dihasilkan dengan membuat sebuah model yang berisi *rating* dari user. Dari model tersebut, akan dihitung *similarity* dari tiap-tiap elemennya. Dengan kata lain, item yang akan diprediksi / direkomendasikan memiliki kesamaan dengan item-item yang telah di-*rating* sebelumnya oleh *active user.*

2.1 Item-based Collaborative Filtering

Pada *item-based CF*, model yang dibangun berupa matriks berukuran *m x m*, dimana *m* adalah jumlah item yang terdapat pada sistem. Pada tahap pembangunan model, untuk setiap item *j*, akan dihitung *"k-most similar items"*, yaitu beberapa item yang memiliki kesamaan yang paling tinggi *{ j_1, j_2, j_3,.., j_k }*. Kemudian, kesamaan diantara item-item tersebut akan disimpan dalam matriks tersebut. Informasi kesamaan tersebut akan digunakan untuk menghitung prediksi *rating* dan memberikan rekomendasi kepada *active user*[4].

Untuk menghitung *similarity* antara dua item, digunakan *adjusted cosine similarity* yang menghasilkan MAE (*mean absolute error*) yang paling rendah berdasarkan penelitian[9].

2.2 Adjusted-cosine Similarity

Adjusted cosine similarity akan digunakan pada item-item yang memiliki user yang sama atau di-*rating* oleh user yang sama (*co-rated items*).

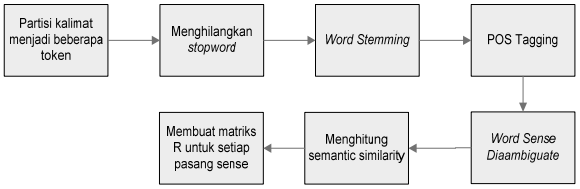
Jika *m* adalah jumlah user $U = \{u_1, u_2, u_3, \ldots, u_m\}$ dan *n* adalah jumlah item $I = \{ i_1, i_2, i_3, \ldots, i_n \}$ dan $R_{m,n}$ adalah matriks rating. Nilai $R_{p,q}$ merupakan rating *user* u_p terhadap item i_q. Maka, algoritma *adjusted cosine similarity* dapat dinyatakan sebagai berikut.

$$sim(i_p, i_q) = \frac{\sum_{k=1}^{m} (R_{k,p} - \overline{R_k}) \cdot (R_{k,q} - \overline{R_k})}{\sqrt{\sum_{k=1}^{m} (R_{k,p} - \overline{R_k})^2 \cdot (R_{k,q} - \overline{R_k})^2}}$$

Dengan $\overline{R_k}$ adalah nilai *rating* rata-rata (*average rating*) dari user *k.*

2.3 Semantic Similarity

Langkah-langkah yang dilakukan dalam menghitung *semantic similarity* antara dua kalimat adalah sebagai berikut [3].



Gambar 0.2 Proses penghitungan *semantic similarity* antara dua kalimat

2.4 Penggabungan Adjusted Cosine Similarity dengan Semantic Similarity

Berdasarkan [10], penggabungan antara *adjusted-cosine similarity* dan *semantic similarity* didefinisikan dengan kombinasi linier sebagai berikut.

$$TotalSim(i_p, i_q) = \alpha * SemSim(i_p, i_q) + (1 - \alpha) * RateSim(i_p, i_q)$$

RateSim(i_p,i_q) adalah kesamaan *rating* antara dua buah item p dan q dengan menggunakan *adjusted cosine similarity*. *SemSim*(i_p,i_q) adalah kesamaan semantic antara dua buah item p dan q dengan menggunakan algoritma Lin (2.1.5). Parameter α adalah parameter kombinasi linier [8]. Jika α = 0, maka *TotalSim*(i_p,i_q) akan sama dengan *RateSim*(i_p,i_q).

2.5 WordNet

Wordnet adalah sebuah basis data yang berisi jaringan semantik untuk bahasa Inggris yang dikembangkan oleh Princeton University[10].

Komponen utama dari wordnet berupa *synset*, yaitu sekumpulan sinonim yang dari suatu konsep (kata), beserta deskripsi makna dari konsep tersebut. *Synset* berbeda dengan "kata"(*words*), tapi merupakan sekumpulan makna kata yang bersinonim.

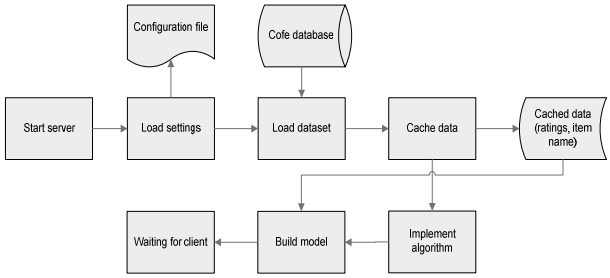
Paper ini menggunakan Wordnet sebagai "kamus" untuk mencari makna semantik dan sebagai *taxonomy* yang akan dijadikan acuan untuk menghitung kesamaan semantik.

3. IMPLEMENTASI SISTEM

Sistem terdiri dari dua buah aplikasi, yaitu aplikasi *client* dan *server.* Aplikasi *client* diimplementasi berdasarkan perancangan yang telah disebutkan sebelumnya. Sedangkan pada aplikasi *server* hanya diimplementasikan algoritma *semantic enhanced item.*

3.1 Collaborative Filtering Engine (CoFE)

Server menggunakan Collaborative Filtering Engine (CoFE) 0.4 sebagai *filtering engine*. CoFE merupakan sebuah aplikasi *open source* yang digunakan dalam riset yang berhubungan dengan *recommender system*. Proses yang dilakukan pada saat CoFE dijalankan dapat dilihat pada diagram dibawah.



Gambar 3.2 Proses *startup* pada CoFE 0.4

3.2 Semantic Enhanced Item Algorithm

Algoritma *semantic enhanced item similarity* merupakan algoritma *item-item similarity* yang telah ditambahkan proses penghitungan *semantic similarity*. Implementasi *semantic similarity* membutuhkan sebuah kamus yang memiliki perbendaharaan kata dalam bahasa Inggris, maka digunakan Wordnet 2.0 sebagai kamus.

4. PENGUJIAN SISTEM

Pada bagian ini diujikan akurasi prediksi dari algoritma *item-item similarity* yang telah ditambahkan algoritma *semantic similarity* (*semantic enhanced item-item similarity*) dan algoritma *item-item similarity* digunakan sebagai pembanding. Parameter-parameter diatas akan diukur besar nilai error nya dengan menggunakan MAE *(Mean Absolute Error).*

4.1 Perbandingan training dan test set

Pengujian dilakukan terhadap algoritma *item-item similarity* dengan *semantic enhanced item similarity,* dengan ukuran model = 50, jumlah neighborhood=50 dan nilai variable α = 0,3. Hasil pengujiannya adalah sebagai berikut :

Tabel 0-1 Hasil pengujian

Test Set	Training Set	Ratio (x)	MAE	
			Item-Item	Semantic
90000	10000	0,1	0,8925579	0,9175423
70000	30000	0,3	0,8109611	0,86699253
50000	50000	0,5	0,7892607	0,822979
30000	70000	0,7	0,762999	0,7780593
10000	90000	0,9	0,7631041	0,76351684

Terlihat bahwa nilai MAE semakin berkurang seiring dengan bertambahnya jumlah *training set*. Pengujian menunjukkan algoritma *item-item similarity* pada nilai *ratio x* yang lebih rendah, tetapi algoritma *semantic* menunjukkan penurunan nilai MAE seiring dengan bertambahnya nilai x. Artinya semakin banyak jumlah data yang digunakan maka semakin baik prediksi yang dihasilkan

4.2 Jumlah neighborhood

Hasil pengujian berdasarkan jumalh neighborhood dapat dilihat pada tabel di bawah ini.

Tabel 0-2 MAE terhadap jumlah *neighborhood*

Jumlah *Neighborhood*	Mean Absolute Error	
	Item	Semantic
20	0,771847	0,7791873
30	0,7696042	0,7760672
40	0,7689988	0,775105
50	0,7689714	0,77512646
60	0,7689714	0,77512646
70	0,7689714	0,77512646
80	0,7689714	0,77512646
90	0,7689714	0,77512646
100	0,7689714	0,77512646
200	0,7689714	0,77512646

Jumlah *neighborhood* memiliki dampak yang signifikan terhadap kualitas prediksi. Jumlah *neighborhood* digunakan untuk menentukan berapa jumlah *item* yang memiliki *similarity* paling tinggi terhadap suatu *item.*

4.3 Ukuran model

Pada *item-based recommender system*, ukuran model memiliki pengaruh dalam menghasilkan prediksi. Ukuran model digunakan pada saat pembangunan model yang dilakukan saat *server* pertama kali diaktifkan.

Tabel 0-3 MAE pada pengujian ukuran model

Ukuran Model	Mean Absolute Error	
	Item	Semantic
25	0,77198726	0,785463
50	0,76448154	0,77512646
75	0,76357096	0,7717011
100	0,76290137	0,77060264
125	0,76271677	0,7704616
150	0,76283383	0,7705824
175	0,7627337	0,7708121
200	0,7632248	0,7709456

Kedua algoritma menunjukkan kurva yang identik. Penurunan *error* terjadi pada *model size* = 25 dan mencapai titik terendah pada *model size* = 125. Dapat disimpulkan bahwa dengan hanya menggunakan sebagian *item* (dari total 1682 *item*), nilai *error* terendah dapat dihasilkan dengan hanya menggunakan 125 *item* yang digunakan sebagai *model.*

4.4 Nilai parameter α

Pengujian yang dilakukan berdasarkan parameter α menghasilkan seperti terlihat pada tabel di bawah ini :

Tabel 0-4 Pengujian terhadap nilai parameter α

alpha	MAE	alpha	MAE
0,1	0,7831899	0,6	0,7901422
0,2	0,7845154	0,7	0,7927745
0,3	0,785463	0,8	0,7943637
0,4	0,78750575	0,9	0,7959526
0,5	0,789382		

Dari tabel diatas, semakin besar nilai variabel α, nilai MAE semakin tinggi. Kenaikan nilai error disebabkan oleh adanya *string* judul film (*item name*) yang tidak terdapat dalam kamus (WordNet).

5. KESIMPULAN

Berdasarkan analisis pengujian yang dilakukan, diperoleh kesimpulan bahwa algoritma *similarity* berpengaruh terhadap prediksi. Kualitas prediksi algoritma *semantic enhanced item similarity* lebih rendah jika dibandingkan dengan algoritma *item item similarity*, hal ini disebabkan oleh adanya *string* dari dataset yang tidak terdapat dalam WordNet. Selain itu semakin besar ukuran training set dan jumlah *neighborhood*, nilai MAE semakin kecil. Sehingga kualitas prediksi menjadi lebih baik.

REFERENSI

[1] Billsus, D. Pazzani, Michael J. *Learning Collaborative Information Filters.* 1998. http://www.ics.uci.edu/~pazzani/Publications/MLC98.pdf. Diakses : 10 Maret 2007
[2] CoFE, http://eecs.oregonstate.edu/iis/CoFE/
[3] Deshpande, M., Karypis, G., *Item-based Top-N Recommendation Algorithms.* http://glaros.dtc.umn.edu/gkhome/fetch/papers/itemrsTOIS04.pdf, diakses : 21 Desember 2007
[4] Herlocker, J.L., Konstan, J., Borchers, A. Riedl, J., *An Algoritmic Framework for Performing Collaborative Filtering*, http://www.grouplens.org/papers/pdf/algs.pdf, diakses pada tanggal 16 Juli 2007
[5] Ioannis, Varelas. *Semantic Similarity Methods in WordNet and Their Application to Information Retrieval on the Web*, 2005, http://nike.psu.edu/widm05/p/p10-varelas.pdf, diakses pada tanggal 24 Oktober 2007
[6] Java WordNet Library (JWNL), http://sourceforge.net/projects/jwordnet
[7] Java WordNet Similarity, http://nlp.shef.ac.uk/result/software/JWordNetSim.zip
[8] Jin, X. Mobasher, B. *Using Semantic Similarity to Enhance Item-based Collaborative Filtering.* http://maya.cs.depaul.edu/~mobasher/papers/ewmf04.pdf, diakses pada tanggal 22 September 2007
[9] McRae, J., Piatek, A., Langley, A., *Collaborative Filtering*. 2004. http://www.imperialviolet.org/suprema.pdf, diakses pada tanggal 21 November 2007
[10] Sarwar, B., Karypis,G., Konstan, J., Riedl, J. *Item-based Collaborative Filtering Recommendation Algorithm*. 2001. http://www.inf.ed.ac.uk/teaching/courses/tts/papers/sarwar.pdf , diakses pada tanggal 16 Juli 2007