Jurnal Integrasi, vol. 6, no. 1, 2014, 21-25
ISSN: 2085-3858 (print version)

Article History
Received 10 February 2014
Accepted 11 March 2014

Analisis dan Pengujian Kinerja Korelasi Dokumen Pada Sistem Temu Kembali Informasi

Ari Wibowo
Program Studi Teknik Multimedia dan Jaringan, Politeknik Negeri Batam
E-mail : wibowo@polibatam.ac.id

Abstrak - *Sistem Temu Balik Informasi adalah ilmu mencari informasi dalam suatu dokumen. Proses pencocokan dilakukan secara parsial dan hanya mencari hasil temu balik yang terbaik. Query diberikan dalam bahasa alami dan dalam bentuk yang tidak lengkap. Sistem Temu Balik Informasi terdiri dari tiga komponen utama, yaitu masukan, pemroses dan keluaran. Penghitungan similaritas akan menghasilkan bobot pada tiap dokumen yang menentukan seberapa relevan dokumen tersebut terhadap query. Metode pembobotan yang digunakan dalam implementasi dapat berupa kombinasi dari TF (Term Frequency), IDF (Inverse Document Frequency), dan normalisasi sesuai input dari user. Pada pengujian terdapat tiga besaran performansi yang dihitung, yaitu Recall, Precision, dan NIAP.*
Kata Kunci : Sistem Temu Balik Informasi, query, performansi, term

1 PENDAHULUAN

Segala jenis informasi terdapat di internet, di samping lengkap, informasi di internet sangat banyak sekali jumlahnya. Hal ini tentunya menimbulkan permasalahan baru, yaitu bagaimana menemukan informasi yang kita inginkan dari sekian banyak informasi yang terdapat di internet. Untuk itu, diperlukan suatu mekanisme pencarian, Information Retrieval System (Sistem Temu Balik Informasi) sebagai sebuah sistem yang mampu mencari informasi yang relevan.

Pemahaman akan suatu ilmu tentunya tidak akan cukup jika ilmu itu tidak diterapkan dalam lingkungan sebenarnya. Untuk tujuan itulah, perangkat lunak untuk pengujian ini dikembangkan. Selain menerapkan ilmu yang didapat, dengan mengembangkan aplikasi ini juga bisa berlatih membangun aplikasi perangkat lunak dengan baik.

2 METODE PENELITIAN

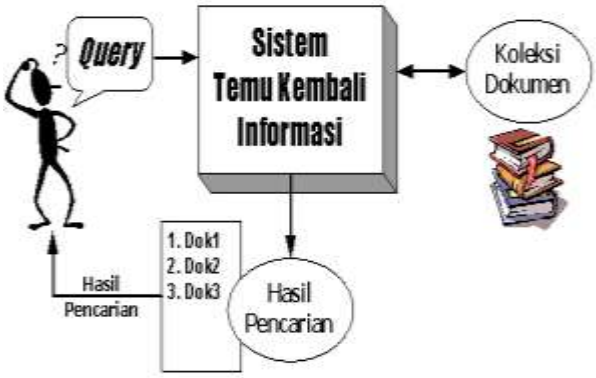
2.1 Sistem Temu Balik Informasi

Sistem Temu Balik Informasi (STBI) adalah ilmu mencari informasi dalam suatu dokumen, mencari dokumen itu sendiri dan mencari metadata yang menggambarkan suatu dokumen. Sistem Temu Balik Informasi merupakan cabang dari ilmu komputer terapan (*applied computer science*) yang berkonsentrasi pada representasi, penyimpanan, pengorganisasian, akses dan distribusi informasi. Dalam sudut pandang pengguna, Sistem Temu Balik Informasi membantu pencarian informasi dengan memberikan koleksi informasi yang sesuai dengan kebutuhan pengguna.

Dalam berbagai hal, Sistem Temu Balik Informasi seringkali disalah artikan menjadi Sistem Basis Data. Kenyataannya, Sistem Temu Balik Informasi memiliki perbedaan mendasar dengan Sistem Basis Data dalam berbagai hal. Karakteristik Sistem Temu Balik Informasi antara lain:

1. Proses pencocokan dilakukan secara parsial (*Partial Match*) dan hanya mencari hasil temu balik yang terbaik (*Best Match*).
2. Proses inferensi dilakukan menurut metode induksi.
3. Model yang diambil adalah model yang bersifat probabilistik.
4. Query diberikan dalam bahasa alami (*natural language*) dan dalam bentuk yang tidak lengkap (*incomplete query*)
5. Hasil temu balik yang diinginkan adalah hasil yang relevan (r*elevant matching*)

Sistem Temu Balik Informasi terdiri dari tiga komponen utama, yaitu masukan (*input*), pemroses (*processor*) dan keluaran (*output*). Komponen-komponen ini digambarkan pada Gambar 1.



Gambar 1 - Skema Umum Sistem

Inti dari sistem temu balik informasi adalah mencari dokumen-dokumen yang relevan sesuai dengan masukan (query) dari pengguna. Oleh karena itu, perlu dihitung similaritas dari tiap dokumen terhadap query yang diberikan. Penghitungan similaritas akan menghasilkan bobot pada tiap dokumen yang menentukan seberapa relevan dokumen tersebut terhadap query, sehingga dapat ditampilkan dokumen-dokumen yang relevan saja, secara terurut mulai dari yang paling relevan (bobot tertinggi).

Dokumen-dokumen yang ditampilkan oleh sistem temu balik informasi harus memenuhi persyaratan berikut:

- Recall : menemukan seluruh dokumen yang relevan dalam koleksi. Recall dapat dihitung dengan rumus:

$$\frac{jumlah\ dokumen\ relevan\ ditemukan}{jumlah\ dokumen\ relevan\ dalam\ koleksi}$$

  Nilai recall tertinggi adalah 1, yang berarti seluruh dokumen dalam koleksi berhasil ditemukan
- Precision : menemukan hanya dokumen yang relevan saja dalam koleksi. Precision dapat dihitung dengan rumus:

$$\frac{jumlah\ dokumen\ relevan\ ditemukan}{jumlah\ dokumen\ ditemukan}$$

  Nilai precision tertinggi adalah 1, yang berarti seluruh dokumen yang ditemukan adalah relevan
- NIAP : (Non Interpolated Average Precision) adalah penggabungan dari recall dan precision, yang dapat dihitung dengan rumus:

$$\sum_{i=1}^{n} \frac{precision\ pada\ dokumen\ ke-i}{jumlah\ dokumen\ relevan\ dalam\ koleksi}$$

Di mana *n* menunjukkan jumlah dokumen yang dicari hingga seluruh dokumen relevan ditemukan.
Nilai NIAP tertinggi adalah 1, yang berarti seluruh dokumen relevan berhasil ditemukan dengan seluruh dokumen relevan tersebut ditempatkan pada urutan teratas dalam hasil pencarian

Nilai NIAP akan digunakan untuk mengecek kebenaran hasil pencarian dari perangkat lunak yang dibangun.

2.2 Metode Pembobotan

Metode pembobotan yang digunakan dalam implementasi Giggle dapat berupa kombinasi dari TF (Term Frequency), IDF (Inverse Document Frequency), dan Normalisasi sesuai input dari user.

2.2.1 Term Frequency

Term Frequency (TF) adalah algoritma pembobotan heuristik yang menentukan bobot dokumen berdasarkan kemunculan term (istilah). Semakin sering sebuah istilah muncul, semakin tinggi bobot dokumen untuk istilah tersebut, dan sebaliknya. Hasil pembobotan ini selanjutnya akan diginakan oleh fungsi perbandingan untuk menentukan dokumen-dokumen yang relevan.

Terdapat empat buah algoritma TF yang digunakan:

- **Raw TF**

Raw TF menentukan bobot suatu dokumen terhadap istilah dengan menghitung frekuensi kemunculan suatu istilah tersebut pada dokumen. Raw TF selanjutnya akan dituliskan sebagak *tf*

- **Logarithmic TF**

  Logarithmic TF mengurangi tingkat kepentingan kemunculan kata dalam menghitung bobot dokumen terhadap suatu istilah dengan melakukan log terhadap TF. Log TF dapat dihitung dengan rumus:

$$ltf = 1 + \log(tf)$$

- **Binary TF**

  Binary TF menyeragamkan bobot dokumen terhadap istilah dengan memberi nilai 0 dan 1. Nilai 1 menyatakan suatu istilah muncul minimal satu kali dalam suatu dokumen, sementara 0 menyatakan sebaliknya.

$$btf = \begin{cases} 1, istilah\ muncul\ dalam\ dokumen \\ 0, istilah\ tidak\ muncul\ dalam\ dokumen \end{cases}$$

- **Augmented TF**

  Augmented TF menyeragamkan bobot dokumen terhadap istilah dengan memberikan range antara 0.5 hingga 1 sebagai bobot dokumen. Augmented TF dapat dihitung dengan rumus:

$$atf = 0.5 + 0.5 \times \left(\frac{tf}{\max tf\ dari\ seluruh\ dokumen}\right)$$

2.2.2 Inverted Term Frequency

Inverse Term Frequency (IDF) meningkatkan nilai bobot dokumen terhadap suatu istiilah dengan rumus heuristik : "semakin banyak dokumen yang mengandung sebuah istilah, maka semakin kecil bobot istilah tersebut (karena tidak dapat digunakan untuk membedakan relevansi dokumen satu dengan yang lain)"

IDF menentukan bobot suatu dokumen terhadap istilah dengan rumus:

$$idf = \log\left(\frac{jumlah\ seluruh\ dokumen\ dalam\ koleksi}{jumlah\ dokumen\ yang\ mengandung\ istilah}\right)$$

2.2.3 Normalisasi

Pembobotan term dengan menggunakan tf dan idf masih belum cukup dan memadai, ini dikarenakan ada faktor penting yang dilupakan yaitu panjang suatu dokumen dalam koleksi. Setiap dokumen yang terdapat dalam koleksi memiliki panjang yang berbeda-beda. Variasi panjang dokumen dalam koleksi akan menyebabkan :

1. Besarnya frekuensi term

Pada dokumen yang panjang, term yang sama cendrung muncul berulang kali sehingga menyebabkan term frequency cendrung besar. Besarnya term frequency mengakibatkan rata-rata bobot term menjadi tinggi dan meningkatkan nilai relevansi dokumen terhadap query pula.

2. Banyaknya term

Dalam dokumen yang panjang, sering ditemukan sejumlah term yang berbeda. Hal ini mengakibatkan meningkatnya sejumlah relevansi antara dokumen dan query.

Normalisasi panjang dokumen dimaksudkan untuk mengurangi hal tersebut diatas. Dengan adanya normalisasi panjang dokumen memungkinkan dokumen yang pendek ikut diperhitungkan dalam pencocokan dokumen (document similarity).

Korelasi Cosine antara vektor query dan vektor dokumen adalah :

$$Cos\left(\vec{Q}, \vec{D}\right) = \frac{\sum_{i=1}^{T} W_{qi}\ X\ W_{di}}{\sqrt[2]{W_{q1}^2 + W_{q2}^2 + \ldots + W_{qr}^2}\ X\ \sqrt[2]{W_{d1}^2 + W_{d2}^2 + \ldots + W_{dr}^2}}$$

dimana :

- wq = bobot tf x idf dari term i dalam query
- wd = bobot tf x idf dalam dokumen

Korelasi dibatasi antara 0 dan 1 dengan menggunakan panjang euclidean dari vektor individu dalam suatu persamaan. Korelasi

cosine dapat juga ditulis dalam bentuk persamaan :

$$Cos\,(\vec{Q},\vec{D}) = \sum_{i=1}^{T}\left(\frac{W_{qi}}{\sqrt[2]{W_{q1}^2 + W_{q2}^2 + \ldots + W_{qr}^2}}\ x\ \frac{W_{di}}{\sqrt[2]{W_{d1}^2 + W_{d2}^2 + \ldots + W_{dr}^2}}\right)$$

2.3 Metode Perbandingan

Metode perbandingan yang digunakan untuk membandingkan tingkat relevansi sebuah dokumen terhadap dokumen yang lain untuk query tertentu adalah metode ruang vektor.

Metode ruang vektor secara sederhana melakukan penghitungan similaritas dari dokumen terhadap query, dengan cara mengalikan semua istilah yang muncul pada query dan istilah pada dokumen dengan menggunakan fungsi similaritas.

Fungsi similaritas berfungsi untuk menghitung similaritas dari dokumen dan *query*. Fungsi ini memanfaatkan hasil dari fungsi pembobotan untuk menentukan similaritas antara dokumen dan *query*. Perhitungan dilakukan dengan rumus:

$$sim = \sum_{i=1}^{T} Wqi\ \times Wdi$$

Dimana T mewakili jumlah kata dalam suatu bahasa, Wqi mewakili bobot istilah-i dalam *query* dan Wdi mewakili bobot istilah-i dalam dokumen.

3 HASIL DAN PEMBAHASAN

3.1 Langkah Pengujian

Pada pengujian kali ini, skenario pengujian yang dilakukan terhadap sistem temu-balik informasinadalah sebagai berikut.

- Melakukan parsing terhadap dataset ADI dan CISI serta memilah-milah dataset tersebut menjadi beberapa dokumen yang terkompresi ke dalam format zip. Tujuan mekanisme ini adalah agar dataset ADI dan CISI menjadi masukan yang dapat diterima oleh aplikasi.
- Melakukan indexing atau pembentukan *inverted table* dengan berbagai kombinasi mode pemobobotan. Untuk setiap percobaan terhadap mode pembototan tertentu, proses indexing disertai dengan proses penghilangan *stop words* berbahasa Inggris, namun tidak melakukan proses *stemming*. Adapun kombinasi mode pembobotan yang digunakan dalam pengujian ini meliputi :
    - Raw Term Frequency
    - Binary Term Frequency
    - Logarithmic Term Frequency
    - Augmented Term Frequency
    - Inverted Term Frequency
    - Normalisasi
- Menghitung nilai *Recall*, *Precision*, dan NIAP untuk setiap percobaan retrieval terhadap tiap query. Dalam pengujian dengan dataset ADI dan CISI, terdapat sebuah file yang terdiri dari query-query dan juga file yang mengambarkan keterhubungan antara query dengan dokumen yang relevan dengannya.

3.2 Hasil Pengujian

Berikut ini didapatkan hasil pengujian yang dilakukan dengan menggunakan data ADI dan CISI.

Tabel 1 – Hasil Pengujian Dataset ADI

Metode	Precision	Recall	NIAP
Raw Term Frequency	0,053	0,800	0,035
Binary Term Frequency	0,048	0,750	0,130
Logarithmic Term Frequency	0,023	0,670	0,270
Augmented Term Frequency	0,038	0,810	0,320
Inverted Term Frequency	0,075	0,761	0,361
Normalisasi	0,089	0,846	0,382

Tabel 2 – Hasil Pengujian Dataset CISI

Metode	Precision	Recall	NIAP
Raw Term Frequency	0,137	0,930	0,479
Binary Term Frequency	0,031	0,930	0,064
Logarithmic Term Frequency	0,002	0,330	0,033
Augmented Term Frequency	0,075	0,800	0,140
Inverted Term Frequency	0,065	0,831	0,282
Normalisasi	0,081	0,867	0,411

3.3 Analisis Hasil

Pada pengujian kali ini, terdapat 3 besaran performansi yang dihitung, yaitu *Recall*, *Precision*, dan NIAP. Mekanisme perhitungan NIAP secara semantik sudah mencakup *Recall* dan *Precision* serta mempertimbangkan peringkat / ranking dari kumpulan dokumen yang terambil oleh sistem, maka baik atau tidaknya sistem temu-balik informasi ini cukup hanya melihat nilai rata-rata NIAP.

Dari data pengujian yang ada di atas dapat dilihat bahwa untuk koleksi dokumen ADI opsi indexing dengan menggunakan Normalisasi memiliki nilai performansi NIAP yang paling tinggi. Hal ini disebabkan karena pada koleksi dokumen ADI yang jumlah dokumennya sedikit, akan didapat jumlah dokumen relevan dan total keseluruhan dokumen yang berbanding lurus. Dengan begitu performansi yang diciptakakn oleh mode ini menjadi paling tinggi.

Sedangkan untuk koleksi dokumen CISI, nilai performansi NIAP tertinggi ditunjukan oleh metode indexing dengan menggunakan *Raw Term Frequency*. Hasil tersebut muncul karena pada mode *Raw Term Frequency* pembobotan dihitung hanya berdasar pada jumlah kemunculan term pada dokumen. Dengan begitu dataset dengan jumlah koleksi dokumen yang banyak seperti pada CISI akan memiliki performansi yang lebih besar.

4 KESIMPULAN

a. Sistem temu balik informasi melakukan penentuan kerelevanan dokumen berdasarkan term yang terdapat di dalam query dan dokumen.

b. Untuk koleksi dokumen yang besar mode yang memiliki performansi paling tinggi adalah Raw Term Frequency.

c. Untuk koleksi dokumen yang kecil, metode Normalisasi menghasilkan nilai performansi paling tinggi

5 SARAN

a. Testing dapat dilakukan pada koleksi dokumen yang lebih banyak.

b. Koleksi dokumen tidak hanya dokumen teks.

6 DAFTAR PUSTAKA

1. Kaniawati, Nia, 2005. *Phrase Indexing* Dalam Sistem Temu Balik Informasi. Program Studi Informatika, Fakultas Teknologi Industri, Institut Teknologi Bandung

2. Lavrenko, Victor., and Bruce Croft, W., 2001. *Relevance-Based Language Models*. Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, United States

3. Robertson, S.E., van Rijsbergen, C.J., and Porter, M.F., 1981. *Probabilistic Model of Indexing And Searching*. Oddy Etal(eds), Information Retrieval Research, Butterworths

4. Singhal, Amit., 2000. *Modern Information Retrieval: A Brief Overview*. Google, Inc., Sillicon Valley, California