ISBN : 978 - 602 - 8467 - 12 - 4

PROSIDING SEMINAR NASIONAL MATEMATIKA V

"Matematika dan Pendidikan Matematika, serta Pengembangan dan Aplikasinya"

Semarang, 24 Oktober 2009

Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Semarang
2009

Kombinatorik

Komputer

IMPLEMENTASI COSINE COEFFICIENT UNTUK PENGUKURAN KEMIRIPAN ANTAR DOKUMEN TEKS BERBAHASA INDONESIA PADA APLIKASI BERBASIS WEB

Mardi Siswo Utomo
Fakultas Teknologi Informasi Universitas Stikubank Semarang

Abstrak Jarak antar dokumen atau biasa disebut dengan Kemiripan dokumen (Document Similarity) biasanya digunakan pada sistem temu kembali informasi. Kemiripan antar dokumen digunakan sebagai acuan pencarian informasi lainya yang sejenis, sehingga dapat mengurangi waktu temu-kembali informasi untuk dokumen berikutnya yang sejenis. Fungsi ini sangat berguna pada korpus dokumen yang besar, sehingga memudahkan pengguna dalam pencarian dokumen-dokumen yang dimaksud.
Salah satu cara untuk mengukur jarak antar dokumen adalah menggunakan Cosine Coefficient. Cosine merupakan pendekatan vektor dalam mengukur sudut relevansi antar dokumen.
Dokumen harus melalui pemrosesan awal (preprocessing) untuk dapat diukur dengan cosine. Pemrosesan dokumen awal dimulai dari analisa token, kemudian dilanjutkan dengan filtering dan terakhir dilakukan proses indek sehingga dihasilkan proximity matrik.
Kemudian juga digunakan teknik ekseskusi parsial pada implementasi aplikasinya untuk dapat menangani dokumen-dokumen yang besar. Aplikasi yang dibangun adalah aplikasi berbasis web sehingga mempunyai fleksibilitas tinggi untuk terminal-terminal aksesnya. Aplikasi berbasis web mempunyai waktu ekseskusi yang terbatas, sehingga dibutuhkan ekseskusi parsial untuk menangani dokumen-dokumen yang banyak.

Kata kunci : Kemiripan dokumen, Cosine

Semakin banyaknya pilihan informasi yang tersedia, maka semakin sulit pula menemukan informasi yang sesuai dengan kebutuhan. Hanya sebagian kecil informasi yang sesuai dengan keinginan pengguna, selebihnya adalah informasi sampah. Sistem pencarian dan penelusuran informasi yang sesuai menjadi hal penting karena dapat menghemat waktu temu-kembali informasi.

Pada kebanyakan aplikasi mesin pencari berbasis web, pencarian dilakukan dengan mencari kata kunci tertentu. Pengguna melihat satu persatu informasi dari hasil mesin pencari, sampai ditemukan dokumen yang dimaksud. Dan akan melakukan hal yang serupa untuk mencari dokumen kedua yang mirip dengan dokumen yang ditemukan di awal, apabila dokumen awal dirasa belum cukup.

Aplikasi untuk melakukan indek pada dokumen dengan kuantitas besar membutuhkan waktu eksekusi yang lama, kelemahan utama aplikasi berbasis web adalah terbatasnya waktu ekseskusi, biasanya web server hanya memperbo-lehkan 30 s/d 60 detik waktu ekseskusi untuk aplikasi berbasis web. Walau dapat diperpanjang waktu eksekusinya, tetapi membuat aplikasi tidak terpantau capaian prosesnya.

Dengan membagi proses yang akan dijalankan menjadi bagian kecil, dapat mengurangi waktu eksekusi. Tetapi dengan resiko aplikasi harus dilakukan berulang-ulang sejumlah bagian yang terbagi. Pointer proses berikutnya disimpan pada variabel session yang akan digunakan lagi

pada eksekusi berikutnya. Bahasa pemrograman digunakan PHP dan manajemen basis data digunakan MySQL, dengan alasan layanan PHP-MySQL mudah dijumpai pada web hosting. Selain itu PHP mendukung perintah ekspresi regular yang sangat berguna untuk proses Tokenizing dan filtering.

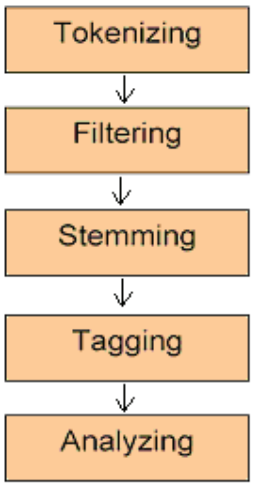
Text Mining

Data mining sendiri digunakan pada proses pengindeksan pada Sistem temu kembali informasi. Pada proses pengindeksan informasi-informasi penting di ekstrak dari dokumen-dokumen yang ada.

Text Mining merupakan bagian dari data mining dimana data mining sendiri mempunyai banyak arti diantaranya adalah : Proses pencarian informasi yang berharga dari data dengan ukuran besar. Data mining juga di definisikan sebagai Ekplorasi dan analisa data ukuran besar untuk menemukan pola-pola dan aturan-aturan yang bermanfaat. Tetapi datamining dapat didefinisikan dengan sederhana yaitu : mengekstrak atau menambang pengetahuan yang bermanfaat dari data berukuran besar (Kamber dan Han 2000:6)

Menurut Salton tipe informasi dapat dikategorikan menjadi 3 macam yaitu informasi berformat teks, informasi berformat suara dan informasi berformat grafik ataupun gambar (Salton 1989:4). Text mining atau sering disebut text data mining dalam bahasa Indonesia disebut dengan penambangan data teks merupakan proses penambangan data berformat teks dari suatu dokumen. Dengan penambangan teks, dapat dicari kata-kata yang dapat mewakili isi dari suatu dokumen. Suatu artikel berita dapat dianalisis apakah artikel berita tersebut tersebut termasuk ke dalam kategori olah raga, kesehatan, selebriti, kriminal, ekonomi, politik atau yang lain, dicocokkan dengan database kata kunci yang sebelumnya telah dibuat. Sehingga diharapkan dapat membantu sistem redaksi elektronik untuk dapat memilah atau mengetahui kategori dari sebuah artikel berita tanpa memerlukan seorang editor. Hal ini akan menghemat waktu dan biaya dalam menjalankan bisnis pada model kantor berita elektronik on-line berbasis internet (Adrifina dkk 2008).

Pada gambar 1 diperlihatkan tahapan-tahapan yang umum dilakukan pada saat melakukan penambangan teks. Proses penambangan melibatkan 5 proses yaitu : a) Tokenizing; b) Filtering; c) Stemming; d) Tagging; e) Analyzing.



Gambar 1 Tahapan penambangan teks

Tokenizing

Proses tokenizing adalah proses pemotongan string masukan berdasarkan tiap kata yang menyusunnya. Pada prinsipnya proses ini adalah memisahkan setiap kata yang menyusun suatu dokumen. Pada umumnya setiap kata terindentifikasi atau terpisahkan dengan kata yang lain oleh karakter spasi, sehingga proses tokenizing mengandalkan karakter spasi pada dokumen untuk melakukan pemisahan kata.

Filtering

Proses Filtering adalah proses pengambilan kata-kata yang dianggap penting atau mempunyai makna saja. Pada proses ini kata-kata yang dianggap tidak mempunyai makna seperti kata sambung akan dihilangkan. Pada proses ini biasanya digunakan Stop Word List untuk melakukan penghilangan kata. Dimana kata-kata yang terdapat dalam stop word list akan dihilangkan. Stop word list berbeda untuk setiap bahasanya.

Stemming

Proses stemming adalah proses untuk mencari root dari kata hasil dari proses filtering. Pencarian root sebuah kata atau biasa disebut degan kata dasar dapat memperkecil hasil indeks tanpa harus menghilangkan makna. Filtering adalah proses pengambilan kata-kata yang dianggap penting atau mempunyai makna. Ada dua pendekatan pada proses stemming yaitu pendekatan kamus dan pendekatan aturan. Beberapa penelitian juga telah dilakukan untuk stemmer bahasa Indonesia baik untuk pendekatan kamus ataupun pendekatan aturan. Vega, Jelita dan Tala mereka masing-masing mempunyai algoritma yang berbeda dalam melakukan proses stemmer pada dokumen bernahasa Indonesia.

Tagging

Proses tagging adalah mencari bentuk utama/root dari suatu kata lampau. Proses tagging tidak digunakan pada dokumen berbahasa Indonesia dikarenakan bahasa Indonesia tidak mengenal kata bentuk lampau.

Analyzing

Proses analyzing adalah proses analisa dari hasil proses tagging sehingga diketahui seberapa jauh tingkat keterhubungan antar kata-kata dan antar dokumen yang ada. Ada 3 pendekatan untuk melakukan pembobotan hubungan antar dokumen yaitu (Baesa dkk, 1998:25)

a) Model Boolean

Model ini merepresentasikan dokumen-dokumen dengan himpunan dari isitilah-isitilah dokumen, dan sebuah query dengan ekspresi boolean dari istilah-isitilah query. Banyak mesin pencari informasi didasarkan pada model ini. Suatu kecocokan diantara sebuah dokumen dan sebuah query biasanya diturunkan dengan menggunakan operasi teori himpunan boolean. Pada himpunan istilah-istilah dokumen dan istilah-istilah query.

b) Model Vektor

Vector Space Model merepre-sentasikan dokumen dan query dengan vektor-vektor bobot istilah dalam sebuah ruang multidimensi. Dalam VSM, sebuah istilah direpresentasikan dengan sebuah dimensi dari ruang vektor. Jadi, relevansi sebuah dokumen ke sebuah query didasarkan pada similaritas diantara vektor dokumen dan vektor query. Similaritas dari dua vektor biasanya dihitung dengan sudut, yakni, **cosine measure**, diantara dua vektor. Pendekatan tree distance (Lakkaraju dkk, 2007) merupakan contoh pendekatan dengan model vektor space.

c) Model Probabilistic

Ide dasar dari model probabilistic adalah bahwa jika diketahui beberapa dokumen relevan ke sebuah query, maka bobot yang lebih tinggi diberikan ke isitilah-isitilah yang mana muncul dalam dokumen-dokumen tersebut untuk mencari dokumen-dokumen lain yang relevan. Dalam model probabilistik, probabilitas munculnya setiap istilah dilatih dengan sebuah himpunan dokumen, himpunan query dan himpunan penentuan similaritas diantara tiap dokumen dan tiap query. Teorema Bayes sering digunakan dalam model ini untuk memberitahu bagaimana memperbaharui atau merevisi kepercayaan berkaitan dengan query yang baru dan dokumen baru. Dokumen yang relevan ke query yang baru dapat diperoleh didasarkan pada probabilitas kemunculan istilah query dalam himpunan dokumen training.

Cosine Coefficient

Cosine Coefficient atau cosine measure merupakan salah satu cara untuk mengukur tingkat kemiripan 2 dokumen. Cosine adalah ukuran kesamaan antara dua dari vektor n dimensi dengan mencari kosinus antar dimensi. Sebagai contoh diberikan dua vektor dari atribut X dan Y, dengan similaritas θ dilambangkan dengan menggunakan titik produk dan besarnya sebagai (Salton 1989:318). Rumus perhitungan nilai similaritas antari 2 buah dokumen diperlihatkan pada algoritma 1. Pada algoritma 2 diperlihatkan rumus similaritas suatu himpunan

Algoritma 1 *Rumus perhitungan nilai similaritas dari 2 buah dokumen*

$$Similarity\ (X,Y) = \frac{\sum_{i=1}^{t} XiYi}{\sqrt{\sum_{i=1}^{t} Xi^2 . \sum_{i=1}^{t} Yi^2}}$$

Algoritma 2 *Rumus perhitungan nilai similaritas himpunan*

$$Similarity\ (X,Y) = \frac{X \bigcap Y}{|X|^{\frac{1}{2}} . |Y|^{\frac{1}{2}}}$$

Dimana :

$X \bigcap Y$ Jumlah terms yang ada di X dan Y

|X| adalah jumlah term yang ada di X

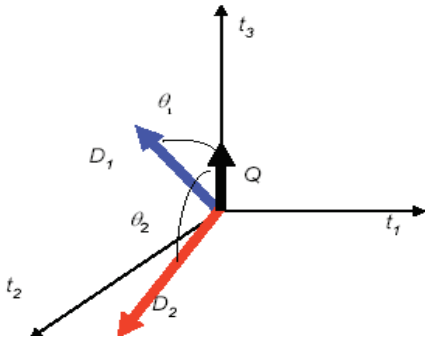
Sedang pada proses pencarian nilai cosinus berdasarkan kata kunci pada setiap dokumen dapat diperlihatkan pada gambar 2.

Dimana:

D Adalah dokumen

Q Adalah kata kunci

t Adalah kata dalam basisdata



Gambar 2 Vektor similaritas kata kunci dengan dokumen

Pada proses pencarian kemiripan antar dokumen yang bukan berdasarkan kata kunci maka variable Q diganti dengan kumpulan kata pada suatu dokumen yang akan dicari dokumen lain yang mirip dengan dokumen tersebut. Tidak ada proses query yang digenerate langsung oleh pengguna. Query dihasilkan dari kata-kata yang muncul di dokumen yang terpilih

Tabel indek

Tabel indek tidak dapat berdiri sendiri, karena membutuhkan data transaksi kata setiap dokumennya, sehingga dibutuhkan pula tabel master kata untuk mempercepat proses query.

Tabel 1 memperlihatkan struktur tabel master kata dan pada tabel 1 merupakan struktur dari tabel transaksi kata dokumen. Struktur tabel similaritas ditunjukan pada tabel 2

Tabel 1 Struktur tabel master kata

Nama Field	Tipe
id	Int(11)
teks	Varchar(30)
jumlah	Int(11)

Tabel 2 Struktur tabel transaksi kata dokumen

Nama Field	Tipe
idkata	int(11)
iddokumen	varchar(4)
jumlah	int(11)

Tabel 3 Struktur tabel similaritas.

Nama Field	Tipe
iddokumen	int(11)
iddokumen2	Varchar(4)
nilai	Decimal(8,5)

Text preprocessing

Dalam text preprocessing ada beberapa langkah yang perlu dilakukan untuk mendapatkan teks yang bebas derau (noise) atau bebas kata-kata yang tidak bermakna. Selain membebaskan dari derau, text preprocessing juga mengembalikan kata menjadi kata dasar atau root word.

Langkah-langkah dalam Text preprocessing dalam bahasa Indonesia adalah :

a)Proses Tokenizing.
b) Proses Filtering.
c)Proses Stemming.

a) tokenizing

Deteksi Token dapat dilakukan dengan perintah untuk mengubah teks menjadi array dengan pemisah karakter ' '. Setelah dlakukan token kata-kata akan terpisah dari kalimatnya dan disusun dalam suatu array.

Setelah token telah dideteksi maka array hasil dari deteksi tersebut diolah oleh proses berikutnya. Pemrosesan pada proses berikutnya dilakukan kata-perkata untuk meringankan proses.

Proses tokenizing dilakukan pada semua dokumen yang ada, untuk data yang besar maka dibutuhkan waktu yang cukup lama. Masalah yang timbul pada aplikasi berbasis web adalah waktu eksekusi web server dibatasi 30 s/d 60 detik tergantung konfigurasinya. Walaupun waktu eksekusi dapat diperlama tetapi hal tersebut akan menggangu kompaktibiltas dengan web-web hosting yang tersedia dan capaian proses tidak terpantau.

Untuk mengatasi hal tersebut maka proses tokenizing dilakukan satu persatu, setelah data record pertama diproses eksekusi oleh web server sampai selesai kemudian digunakan javascript autoreload untuk memproses dokumen berikutnya untuk memanggil fungsi yang sama tetapi untuk dokumen berikutnya.

Proses ini menggunakan variabel session cookies untuk menyimpan pointer data terakhir dan maksimal / jumlah data yang akan diproses. Keuntungan dari cara ini sistem dapat memproses dokumen dengan jumlah yang sangat banyak, sekalipun kemampuan terbatas pada kemampuan menyimpan dari layanan basisdata. Kekurangan dari cara ini adalah waktu eksekusi menjadi lebih lama karena hanya 1 dokumen saja yang diproses setiap kali program dipanggil.

Untuk pemrosesan 1 dokumen data tidak perlu melakukan mekanisme autoreload, seperti pada saat pemrosesan 1 record pada saat setelah data disimpan waktu dilakukan perubahan ataupun penambahan data baru.

b) Filtering

Proses filtering meliputi 2 proses yaitu :

Proses pertama menghilangkan angkadan tanda baca dan tag-tag html.

Proses kedua menghilangkan kata-kata yang tidak bermakna (tidak, ke, yang , dsb) biasa disebut dengan stopword atau stoplist.

Proses ketiga adalah mengubah semua kata yang tersisa menjadi kata dasar (root) dengan menggunakan algoritma stemming tertentu.

c) Proses Indek

Setelah kata telah dikembalikan dalam bentuk asal (kata dasar), proses selanjutnya adalah pengindekan kata-kata dasar tersebut. Tujuan akhir dari proses indek ini adalah proximity matrik. Pada proximity matrik diketahui jarak antar dokumen berdasarkan dari jumlah kata yang berpotongan dihitung dengan rumus cosine coeficient.

Tabel dokumen berelasi dengan tabel master kata menghasilkan tabel transaksi katadokumen. Berikut ilustrasi tabel master dokumen pada tabel 4, tabel master kata pada tabel 5 dan tabel transaksi katadokumen pada tabel 6.

Tabel 4 Tabel master dokumen

ID Dokumen	Judul
1	Sistem Informasi Akademik Berbasis WEB
2	Disain dan Implementasi Sistem Pakar Berbasis WEB
3	Implementasi Kunci Publik Pada Koneksi Jaringan Bluetooth

Tabel 5 Tabel master kata

IDKata	Kata
1	Sistem
2	Informasi
3	akademik
4	Basis
5	WEB
6	Disain
7	Implan
8	Pakar
9	Kunci
10	Publik
11	Konek
12	Jaring
13	Bluetooth

Tabel 6 Tabel transaksi katadokumen

IDDokumen	IDKata	Jumlah
1	1	1
1	2	1
1	3	1
1	4	1
1	5	1
2	6	1
2	7	1
2	1	1
2	8	1
2	4	1
2	5	1
3	7	1
3	9	1
3	10	1
3	11	1
3	12	1
3	13	1

Implementasi algoritma cosine coefficient

Setelah kata dan dokumen terindek dalam tabel transaksi katadokumen langkah selanjutnya adalah melakukan pengkuran jarak antar dokumen dengan Cosine Coefficient. Hasil pengukuran disimpan dalam tabel cosine (proximity matrix). Tabel 7 adalah ilustrasi nilai dari tabel cosine. Nilai cosine dihasilkan dari rumus pada algoritma 3.

Proses perhitungan nilai similaritas juga digunakan teknik pemrograman yang sama seperti pada proses filtering. Proses perhitungan dilakukan satu-persatu pada setiap dokumen. Setelah memproses 1 dokumen proses akan berhenti dan javascript akan memuat ulang halaman web, sehingga dokumen berikutnya akan diproses.

Tabel 7 Ilustrasi Tabel Cosine

IDDokumen1	**IDDokumen2**	Nilai
1	2	0,6
1	3	0,4
2	1	0,6
2	3	0,5
3	1	0,4
3	2	0,5

Algoritma 3 *Rumus nilai cosine*

```
Nilai =

(Cosine(Judul dokumen1,Judul Dokumen2) +

Cosine(Isi dokumen1, Isi Dokumen2)) / 2
```

Proses perhitungan nilai cosine coefficient dapat dilakukan langsung menggunakan stored procedure Mysql, sehingga menghemat waktu pemrosesan dan pemrograman.

Implementasi Visualisasi Hasil

Visualisasi similaritas dilakukan saat sebuah dokumen di baca, pada bagian bawah dokumen ditampilkan daftar judul, hyperlink dan nilai kemiripan dokumen lain yang mempunyai nilai kemiripan tertinggi dengan dokumen yang sedang dibaca. Daftar kemiripan dokumen ditampilkan terurut berdasarkan nilai similaritas tertinggi, dengan nilai similaritas minimal 20% atau 0,2 diperlihatkan pada gambar 2. Hanya dokumen dengan nilai similaritas <= 0,2.

Uji Coba

Korpus menggunakan data judul dan abstrak skripsi mahasiswa. Total Dokumen yang digunakan dalam penelitian ini adalah 500 dokumen sehingga didapatkan proximity matrik sebesar 500 x 500 = 250.000 elemen, dengan distribusi nilai sbb :

Tabel 8 Hasil pengukuran proses similaritas

Similaritas	Jumlah sel
100%	500
61% - 70%	20
51% - 60%	46
41% - 50%	157
31% - 40%	649
21% - 30%	4.442
11% - 20%	28.596
1% - 10%	146.395
0	69.195
Total :	250.000

Daftar Pustaka

Adrfina A; Putri JU dan Simri IW, 2008, *Pemilahan Artikel Berita dengan Text Mining, KOMMIT 2008,* Jakarta Indonesia

Baesa R; Ribeiro B, 1998, *Modern Information Retrieval*, ACM Press New York USA

Klose A; N¨urnberger; Krusel; Hartmann dan Richards, 2000, *Interactive Text Retrieval Based on Document Similarities*, Institute for Knowledge and Language Processing, University of Magdeburg, Jerman.

Lakkaraju P; Gauch S dan Speretta M, 2007, *Document Similarity Based on Concept Tree Distance,* University of Kansas, USA

Lee M; Pincombe B dan Welsh M, 2005, *An Empirical Evaluation of Models of Text Document Similarity,* University of Adelaide, Australia

Liu Y; Hui C; Hang M dan Ma S, 2004 *Finding abstract field of web pages or query specific retrieval*, Text Retrieval Conference. http://trec.nist.gov (diakses tanggal 24 Maret 2009)

Sihombing P; Embong A dan Sumari P, 2006, *Comparison of Document Similarity in Information Retrieval System by Different Formulation*, Universiti Sains Malaysia, Penang, Malaysia.

Wijaya S; Nugroho B; Khoerniawan T dan Mirna A, 2007, *Analisis struktur dokumen pada perolehan informasi dokumen web*, Faculty of computer science University of Indonesia, Indonesia