Model Ruang Vektor pada *Information Retrieval System*

Rama Febriyan 13511067
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13511067@std.stei.itb.ac.id

***Abstract*—Sebuah *Information Retrieval System* merupakan sistem yang digunakan untuk mencari informasi berdasarkan *query* yang diberikan oleh pengguna. Informasi yang dicari berasasal dari dokumen yang tidak terstruktur. Dalam melakukan *retrieval*, sebuah sistem IR dapat menggunakan model ruang vektor untuk menentukan seberapa mirip suatu dokumen dengan query yang ada. Pada makalah ini, dibahas bagaimana suatu sistem IR memodelkan kecocokan dengan dokumen menggunakan vektor.**

***Keywords*—Vektor, mesin pencari, *Information Retrieval System*, model ruang vektor.**

I. PENDAHULUAN

Mesin pencari atau *search engine* merupakan hal yang pada zaman sekarang tidak lepas dari kehidupan manusia. Google, Bing, Yahoo merupakan beberapa contoh mesin pencari yangumum digunakan.

Mesin pencari tidak bekerja hanya dengan mencocokkan teks yang ada pada kata kunci dengan teks pada dokumen begitu saja. Akan tetapi, mesin pencari yang telah dikenal juga melakukan penyusunan dokumen berdasarkan seberapa cocok dokumen dengan kata kunci.

Mesin pencari merupakan salah satu penerapan dari *information retrieval system*, sebuah aktivitas untuk mendapatkan informasi yangrelevan dari sekumpulan sumber informasi. Sebuah system IR dapat dimodelkan dengan menggunakan ruang vektor. Misalkan dalam suatu dokumen terdapat *n* kata yang berbeda. Kata-kata ini akan membentuk ruang vektor yang memiliki dimensi sebesar *n*. Selanjutnya, model ruang vektor ini akan digunakan untuk memodelkan seberapa relevan kata kunci atau *query* yang diberikan dengan kata pada dokumen yang ada.

II. VEKTOR

Vektor merupakan besaran yang memiliki besar dana rah. Sebuah vektor $\bar{u}$ merupakan sebuah list dari angka-angka

$$u = (u_1, u_2, \dots, u_n)$$

di mana u_i disebut sebagai komponen dari $\bar{u}$. Besar dari vektor $\vec{u}$ dituliskan dengan $\|\vec{u}\| \equiv u$. Posisi, perpindahan, kecepatan dan percepatan merupakan contoh dari vektor.

Vektor memiliki sifat-sifat sebagai berikut [1]:

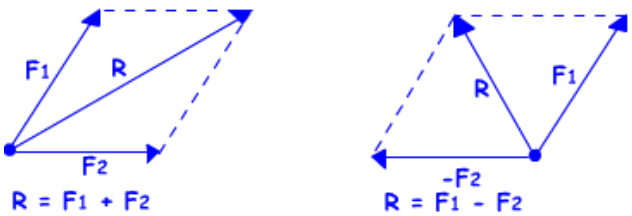
1. Vektor dikatakan sama jika memiliki besar dan arah yang sama
2. Vektor harus memiliki unut yang sama agar dapat dijumlahkan atau dikurangkan
3. Negatif dari suatu vektor memiliki besar yang sama namun berlawanan arah
4. Pengurangan vektor dapat dilakukan dengan menjumlahkan dengan vektor negatif
5. Perkalian atau pembagian vektor dengan skalar akan menghasilkan vektor
6. Proyeksi dari suatu vektor di sepanjang sumbu koordinat disebut sebagai komponen vektor
7. Menjumlahkan vektor delakukan dengan menjumlahkan komponen-komponen yang bersesuaian

Misalkan terdapat vektor **u** dengan komponen (u_1,u_2,u_3), maka panjang vektor **u** dapat dihitung dengan

$$\|\vec{u}\| = \sqrt{{u_1}^2 + {u_2}^2 + {u_3}^2}$$

Jika terdapat vektor **v**=(v_1,v_2,v_3) maka

$$\boldsymbol{u} + \boldsymbol{v} = (u_1 + v_1, u_2 + v_2, u_3 + v_3)$$



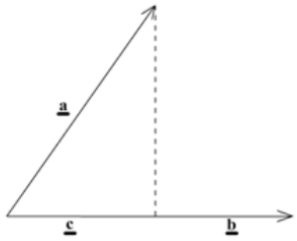
Gambar 1 Penjumlahan dan pengurangan vektor (sumber: google)

Perkalian skalar suatu vektor dapat dituliskan sebagai

$$\boldsymbol{u} \cdot \boldsymbol{v} = \|u\|\|\boldsymbol{v}\| \cos \alpha$$

dengan α merupakan sudut antara vektor **u** dan **v**.

Sebuah vektor dapat di proyeksikan ke sumbu pada koordinat kartesian atau pada vektor lainnya.



Gambar 2 Proyeksi vektor a terhadap vektor b

III. INFORMATION RETRIEVAL SYSTEM

Information retrieval system atau sistem temu balik informasi merupakan sebuah aktifitas untuk mendapatkan informasi yang relevan dari sumber-sumber yang ada. Contoh IR yang palin umum adalah mesin pencari. Pada mesin pencari dimasukkan kata kunci atau *query* yang ingin ditemukan. Kata kunci ini kemudian akan di cocokkan dengan dokumen yang ada kemudian dikembalikan kepada *user* dengan urutan berdasarkan tingkat kerelevanan dokumen tersebut dengan *query*.

Gambar 3 Ilustrasi dari sebuah IR system (sumber: google)

IR berbeda dengan pancarian pada basis data biasa. IR lebih digunakan untuk melakukan pencarian dari informasi yang tidak terstruktur [2].

Tujuan dari sebuah sistem IR yang ideal adalah:

1. Menemukan seluruh dokumen yang relevan
2. Menemukan dokumen yang relevan saja

Terdapat dua istilah yang berakaitan dengan tujuan tersebut, yaitu *recall* dan *precission* [3]. *Recall* adalah perbandingan dari dokumen relevan yang ditemukan dengan jumlah dokumen yang relevan dalam koleksi dokumen.

$$recall = \frac{jumlah\ dokumen\ relevan\ ditemukan}{jumlah\ seluruh\ dokumen\ relevan}$$

Sedangkan precission merupakan perbandingan antara jumlah dokumen relevan yang ditemkukan dengan jumlah dokumen yang berhasil ditemukan.

$$precision = \frac{jumlah\ dokumen\ relevan\ ditemukan}{jumlah\ seluruh\ dokumen\ ditemukan}$$

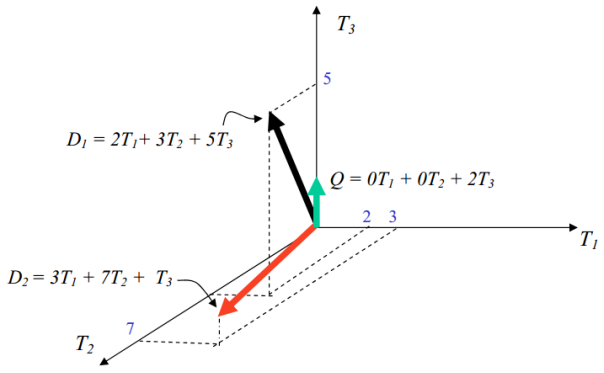
Recall dan *precision* merulakan perhitingan terhadap kumpulan dokumen hasil secara keseluruhan. Keduanya menggambarkan perfromansi dari sebuah sistem IR dalam menemukan dokumen yang relevan. Akan tetapi, pengukuran ini tidak akan menggambarkan performansi sistem mengenai urutan dari dokumen-dokumen yang relevan. Keadaan ini dapat dicontohkan jika terdapat dua hasil pencarian ketika *recall* pada hasil pencarian pertama sama dengan *precision* pada hasil pencarian kedua dan *precision* pada hasil pencarian pertama memiliki nilai yang sama dengan *recall* pada hasil kedua. Nilai-nilai ini belum menggambarkan sistem yang mana yang lebih baik. Keadaan lainnya adalah ketika dua sistem IR menghasilkan *recall* dan *precision* yang sama, namun menghasilkan urutan dokumen yang berbeda.

A.Model Ruang Vektor

Sebuah sistem IR dapat dimodelkan dengan model ruang vektor. Misalkan terdapat kata yang berbeda sebanyak *n* dalam dokumen. Kata-kata tersebut akan membentuk sebuah ruang vektor berdimensi *n*. Setiap kata atau *term* pada dokumen atau query diberikan bobot sebesar w_i. Misal, terdapat tiga *term* T_1,T_2, dan T_3 serta dua dokumen D_1 dan D_2 dan *query* Q. Masing-masing dituliskan dalam bentuk vektor:

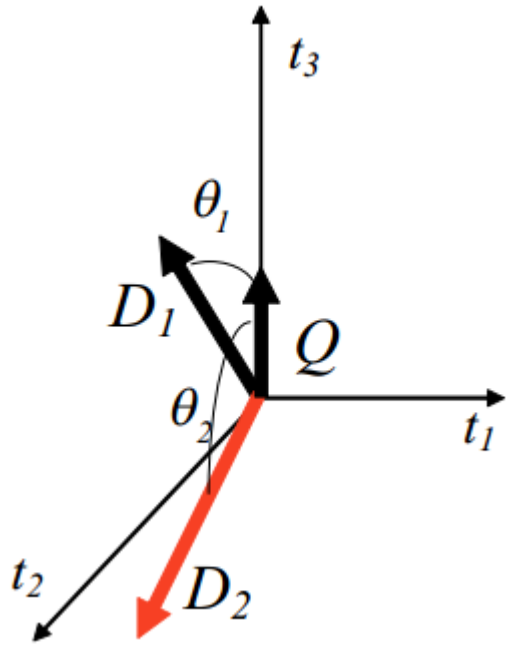
$$D_1 = (2, 3, 5)$$
$$D_2 = (3, 7, 0)$$
$$Q = (0, 0, 2)$$

Ketiga vektor tersebit dapat di representasikan dalam bentuk grafik seperti pada gambar 2.



Gambar 4 Representasi dokumen dan query dalam vektor [3]

Relevansi suatu dokumen dengan *query* dipandang sebagai pengukuran kesamaan atau pengukuran kemiripan (*similarity*) antara vektor dokumen dengan *query*. Semakin sama suatu vektor dokumen dengan *query* maka dapat dikatakan bahwa semakin relevan dokumen tersebut dengan *query*. Sudut yang dibentuk antara vektor dokumen dengan vektor *query* menggambarkan kesamaan dokumen dengan *query* tersebut. Representasi dari hubungan sudut vektor dokumen dengan *query* dapat dilihat pada gambar 3.



Gambar 5 Representasi sudut vektor dokumen dan query

Perhitungan kesamaan vektor **Q** dan **D** dapat dilakukan sebagai berikut:

$$Sim(Q,D) = \cos(Q,D) = \frac{Q \cdot D}{\|Q\|\|D\|} = \frac{1}{\|Q\|\|D\|}\sum_{i=1}^{n} Q_i \cdot D_i$$

Perhitungan kemiripan ini memiliki keuntungan dengan adanya normalisasi terhadap panjang dokumen. Hal ini akan memperkecil pengaruh panjang dokumen. Panjang kedua vektor digunakan sebagai faktor normalisasi [3].

Proses perangkingan dianggap sebagai proses pemilihan vektor dokumen yang dekat dengan vektor *query*.

Besar vektor dokumen berasal dari bobot kata pada dokumen. Pemberian bobot pada kata dapat dilakukan dengan menghitung jumlah kemunclan kata atau *term frequency*. Semakin besar kemunculan suatu kata, akan semakin besar pula bobot dari kata tersebut. Faktor yang patut di perhatikan dalam pemberian bobot adalah kejarangmunculan kata dan faktor normalisasi terhadap panjang dokumen [3]. Kata yang muncul pada sedikit dokumen harus dipandang sebagai kata penting. Dalam koleksi dokumen terdapat berbagai dokumen dengan panjang yang beragam. Keadaan seperti ini akan memicu adanya ketimpangan karena dokumen yang panjang cenderung memiliki frekuensi kemunculan kata yang besar.

Terdapat beberapa metode untuk menghitung *term frequency* suatu kata, yaitu:

1. *raw tf*, menghitung jumlah kemunculan *term* pada dokumen.
2. *logaritmic tf*, bobot dihitung dengan persamaan:

$$1 + \log(tf)$$

3. *binary tf*, bobot dihitung berdasarkan ada atau tidak adanya *term* pada suatu dokumen yang direpresentasikan dengan nilai 0 atau 1
4. *augmented tf* dilakukan dengan memperkecil jarak nilai *tf*. *Augmented tf* dihitung dengan cara

$$0.5 + 0.5 \times \frac{tf}{\max tf}$$

Untuk menentukan apakah suatu *term* merupakan *term* yang penting atau tidak, digunakan *inverse document frecuency (idf)*. Jika terdapat N total dokumen dan sebanyak df_i buah dokumen yang mengandung *term* t_i, nilai dari *inverse document frecuency* dihitung dengan

$$idf_i = \log\left[\frac{N}{df_i}\right]$$

Karena dokumen memiliki panjang yang berbeda, dilakukan normalisasi untuk menhilangkan kekurangan yang terjadi dalam pembobotan kata. Kekurangan tersebut adalah tingginya bobot kata karena dokumen yang panjang dan banyaknya term yang ditemukan pada dokumen yang panjang. Proses yang dilakukan adalah dengan normalisasi panjang dokumen. Normlisasi yang umum dilakukan adalah *cosine normaization* sesuai persamaan berikut:

$$\cos(\vec{Q},\vec{D}) = \frac{\sum_{i=1}^{T} w_{qi} \times w_{di}}{\sqrt{w_{q1}^2 + w_{q2}^2 + \ldots + w_{qr}^2} \times \sqrt{w_{d1}^2 + w_{d2}^2 + \ldots + w_{dr}^2}}$$

B.Generalized Vector Space Model

Generalized Vector Space Model merupakan sebuah bentuk hasil generalisasi dari model ruang vektor biasa. GVSM menggunakan korelasi antara *term* dengan *term* lainnya. Dalam GVSM, setiap *term* t_i diekspresikan sebagai kombinasi linear 2^n dari vektor m_r dengan $r = 1 \ldots 2^n$.

Untuk dokumen D_k *query* Q, dan t_i serta t_j merupakan vektor dari ruang dimensi 2^n, fungsi kesamaan berubah menjadi:

$$sim(Q,D_k) = \frac{\sum_{j=1}^{n}\sum_{i=1}^{n} w_{i,k} \times w_{j,q} \times t_i \cdot t_j}{\sqrt{\sum_{i=1}^{n} {w_{i,k}}^2} \times \sqrt{\sum_{i=1}^{n} {w_{i,q}}^2}}$$

Korelasi term dapat diimplementasikan dalam berbagai cara. Misalnya menggunakan frekuensi kemunculan term sebagai input dari algoritma [4].

Tsatsaronis [5] melakukan pengukuran *Semantic Relatedness* menggunakan thesaurus *O*. Perhitungan mempertimbangkan panjang yang diambil oleh *compactness* (SCM) dan kedalaman oleh *semantic path elaboration (SPE)*. Perkalian dalam t_i dengan t_j dihitung

dengan:

$$t_i \cdot t_j = SR((t_i, t_j), (s_i, s_j), O)$$

s merupakan *sense* dari *term t* dengan memaksimalkan nilai $SCM \cdot SPE$.

IV. KESIMPULAN

Vektor merupakan besaran yang umum muncul pada bidang matematika dan fisika. Akan tetapi, vektor juga dapat digunakan untuk memodelkan sebuah sistem temu balik informasi.

Dengan menggunakan model ruang vektor pada suatu sustem IR, perangkingan dokumen menjadi lebih mungkin dilakukan. Hal ini dapat membantu utntuk menemukan dokumen yang relevan sebagaimana yang dimaksud oleh pengguna.

Akan tetapi, penggunaan model ruang vektor akan terasa buruk jika digunakan pada dokumen yang sangat panjang karena akan menghasilkan vektor dengan dimensi yang besar.

REFERENCES

[1] Unwinnipeg, "Scalars and Vectors," 10 September 1997. [Online]. Available: http://theory.uwinnipeg.ca/physics/twodim/node2.html. [Diakses 11 December 2015].

[2] R. Munir, "Homepage Rinaldi Munir," 2015. [Online]. Available: http://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2015-2016/Aplikasi%20Aljabar%20Vektor%20pada%20IR.pptx. [Diakses 11 December 2015].

[3] R. Mandala, Pengujian dan Model Ruang vektor (PDF).

[4] S. Wong dan P. C. N. W. Wojciech Ziarko, Generalized vector spaces model in information retrieval, New York: SIGIR ACM, 1985.

[5] G. V. P. Tsatsaronis, A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness, New York: ACM, 2009.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 16 Desember 2015

Rama Febriyan 13511067