TEXT MINING UNTUK PENCARIAN DOKUMEN BAHASA INGGRIS MENGGUNAKAN SUFFIX TREE CLUSTERING

Tatas Wicaksono
Jurusan Teknik Informatika
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember
E-mail : tatas@student.eepis-its.edu

Abstrak

Sebuah pencarian terhadap kumpulan dokumen umumnya memberikan hasil berupa cuplikan dokumen-dokumen yang disusun berdasarkan peringkat kecocokan dalam daftar yang panjang. Tidak jarang suatu pencarian menghasilkan puluhan bahkan ratusan cuplikan dokumen yang menyebabkan seorang pengguna harus menggulung layar ke atas dan ke bawah (*scrolling*) untuk meneliti satu persatu cuplikan dokumen. Keadaan ini menyebabkan seorang pengguna mengalami kesulitan dalam hal menentukan dokumen yang relevan dengan topik yang ia inginkan.

Pada Proyek Akhir ini dikembangkan suatu aplikasi pengelompokan dokumen berbasis web dengan metode *suffix tree clustering*. Konsep dasar metode ini adalah dengan mengelompokkan dokumen hasil pencarian ke dalam bentuk grup-grup atau clusters berdasarkan kata atau frase yang terdapat di dalam dokumen-dokumen tersebut. Aplikasi membutuhkan input pencarian dan akan menghasilkan output berupa cluster yang di dalamnya terdapat dokumen yang bersesuaian. Cluster ini bisa bertingkat-tingkat tergantung dari kata atau frase yang mungkin bisa dibedakan lagi pada cluster induk yang sama. Cluster-cluster yang dihasilkan inilah yang ditampilkan kepada pengguna. Selanjutnya pada cluster terakhir yang dipilih akan menampilkan kumpulan dokumen yang masing-masing terdiri dari judul, cuplikan dan URL dokumen. Dengan metode ini diharapkan hasil pencarian akan lebih mudah untuk ditelusuri.

Kata kunci : *text mining, suffix tree, suffix tree clustering, pengelompokan dokumen*.

1. PENDAHULUAN

Perkembangan teknologi dewasa ini khususnya internet berkembang sangat pesat. Hal ini diiringi juga dengan semakin berkembangnya Teknologi Informasi yang dibutuhkan oleh pengguna sehingga mengakibatkan munculnya suatu cabang ilmu baru dalam Teknologi Informasi yaitu Pencarian Informasi (*Information Retrieval*).

Sistem pencarian dokumen pada umumnya menampilkan hasil pencarian dalam daftar yang panjang berdasarkan peringkatnya. Kemudian pengguna diharuskan memilah sendiri dokumen mana yang relevan dengan topik yang ia cari dalam daftar tersebut. Sayangnya sebagian besar *search engine* menggunakan paradigma tersebut. Selain itu *search engine* juga memiliki karakteristik yaitu memiliki presisi hasil pencarian yang rendah. Kedua kelemahan *search engine* tersebut membuat pengguna cukup kesulitan untuk menemukan informasi dari dokumen yang mereka cari.

Ada sebuah cara yang cukup membantu untuk mendapatkan hasil pencarian dokumen dengan tepat dan mudah. Kita bisa menggunakan model *clustering* untuk mengelompokkan hasil pencarian dokumen sesuai dengan topik yang terkait. Dalam proyek akhir ini digunakan metode *Suffix Tree Clustering (STC)* untuk mengelompokkan dokumen hasil pencarian. STC tidak memperlakukan dokumen sebagai suatu himpunan kata-kata tetapi lebih sebagai *string*, yaitu memanfaatkan kedekatan informasi antar kata. STC bergantung pada model pohon *suffix* untuk mengefisiensikan identifikasi set dokumen yang berbagi frase dan menggunakan informasi ini untuk membuat klaster.

2. DASAR TEORI

Text Mining

Definisi dari *text mining* sudah sering diberikan oleh banyak ahli riset dan praktisi. Seperti halnya *data mining*, *text mining* adalah proses penemuan akan informasi atau *trend* baru yang sebelumnya tidak terungkap dengan memproses dan menganalisa data dalam jumlah besar. Wikipedia mendefinisikan text mining sebagai berikut. *"Text mining, also known as intelligent text analysis, text data mining, unstruktur data management, or knowledge discovery in text…, refers generally to the process of extracting interesting and non-trivial information and*

knowledge (usually converted to metadata elements) from unstruktur text (i.e. free text) stored in electronic form." [Wikipedia]

Proses *text mining* meliputi proses *tokenizing, filtering* dan *stemming*. *Tokenizing* adalah proses penghilangan tanda baca pada kalimat yang ada dalam dokumen, sehingga menghasilkan kata-kata yang berdiri sendiri-sendiri.

Tahap *filtering* adalah tahap pengambilan kata-kata yang penting dari hasil *tokenizing*. Tahap *filtering* ini dapat menggunakan algoritma *stoplist* atau *wordlist*. *Stoplist* yaitu penyaringan (*filtering*) terhadap kata-kata yang tidak layak untuk dijadikan sebagai pembeda atau sebagai kata kunci dalam pencarian dokumen sehingga kata-kata tersebut dapat dihilangkan dari dokumen. Sedangkan *wordlist* adalah daftar kat-kata yang mungkin digunakan sebagai kata kunci dalam pencarian dokumen, dengan demikian maka tentu jumlah kata yang termasuk dalam *wordlist* akan lebih banyak daripada *stoplist*, sehingga dalam proyek akhir ini digunakan daftar *stoplist.*

Stemming adalah proses mengubah kata menjadi kata dasarnya dengan menghilangkan imbuhan-imbuhan pada kata dalam dokumen atau mengubah kata kerja menjadi kata benda.

Stemming dalam Bahasa Inggris

Algoritma ini didahului dengan pembacaan tiap kata dari *file* sampel. Sehingga input dari algoritma ini adalah sebuah kata yang kemudian dilakukan :

1. Pengecekan kata pada kamus *irregular verb* yang sudah disiapkan. Apakah kata tersebut berjenis *irregular verb* ataukah tidak. Bila kata tersebut berjenis *irregular verb* maka dikembalikan dalam bentuk kata dasarnya (*infinitive*). Bila tidak maka dilanjutkan pada proses selanjutnya.
2. Mengambil kata terakhir dari pasangan kata (*kata majemuk*), karena yang memiliki arti yang utama adalah pada kata terakhir. Sebagai contoh kata *"half-life",* maka yang diambil adalah kata "*life*".
3. Kata yang sifatnya *plural* diubah menjadi *singular*, sebagai contoh :
   - *classes* menjadi *class*
   - *studies* menjadi *study*
   - *cats* menjadi *cat*
4. Menghilangkan akhiran *–ed* dan *–ing* pada kata yang memiliki akhiran tersebut.
5. Mengubah kata-kata dengan akhiran seperti (*tional, izer, ization, ation, ator, alism, fulness, ousness, ical, dll*) ke bentuk dasarnya.
6. Menghilangkan akhiran seperti (*full, ness, ence, er, ic, able, ible, ment, ent, ion, ism, dll*).

Pengklasteran

Pengklasteran adalah proses membuat pengelompokan sehingga semua anggota dari setiap partisi mempunyai persamaan berdasarkan matrik tertentu. Sebuah *cluster* adalah sekumpulan obyek yang digabung bersama karena persamaan atau kedekatannya [Ali]. Pengklasteran atau klasterisasi merupakan sebuah teknik yang sangat berguna karena akan mentranslasi ukuran persamaan yang intuitif menjadi ukuran yang kuantitatif. Ada banyak pendekatan untuk membuat *cluster* diantaranya adalah membuat aturan yang mendikte keanggotaan dalam grup yang sama berdasarkan tingkat persamaan di antara anggotanya. Pendekatan lainnya adalah dengan membuat sekumpulan fungsi yang mengukur beberapa properti dari pengelompokan tersebut sebagai fungsi dari beberapa parameter dari sebuah pengklasteran.

Suffix Tree Clustering

Inti dari suatu hasil pencarian yang menerapkan *clustering* adalah penggunaan algoritma *clustering*. Algoritma *Suffix Tree Clustering* (STC) memiliki dua kunci utama, yaitu :

1. Menggunakan *phrase* sebagai dasar pembentukan *cluster*-nya.
2. Menggunakan suatu definisi *cluster* sederhana.

Suffix tree clustering memiliki dua langkah utama. Dalam langkah pertama, pencarian *shared phrase* untuk semua dokumen yang dikoleksi. Kita menyebut *shared phrase* sebagai *phrase cluster* atau *base cluster*, yang ditemukan dengan menggunakan suatu struktur data yang dinamakan *suffix tree* [Novan]. Dalam langkah kedua, kita mengkombinasikan *base cluster-base cluster* ke dalam suatu *cluster*. Panggabungan antar dua *base cluster* didasarkan pada jumlah dokumen yang melakukan overlap diantara kedua *base cluster* tersebut [Zamir]. Suatu *phrase* yang dimaksud dalam konteks algoritma ini adalah urutan satu atau lebih kata-kata.

STC memiliki tiga langkah utama, yaitu:

1. *Cleaning Dokumen*.

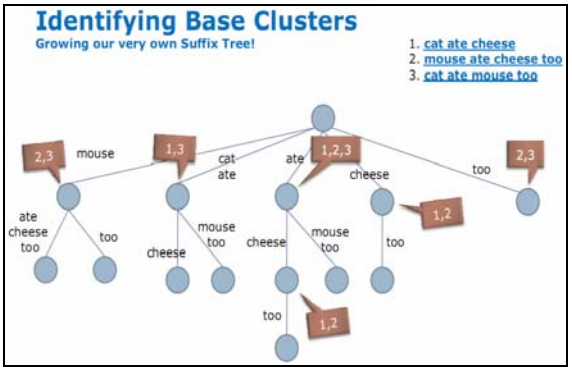
2. Identifikasi *Base Cluster* menggunakan *Suffix Tree*.
3. Mengkombinasikan *Base Cluster* ke dalam suatu *cluster*.

Document Cleaning

Document Cleaning adalah tahap awal dalam algoritma *Suffix Tree Clustering.* Pada tahap ini, dokumen yang telah didapat akan dibersihkan dan dipersiapkan untuk tahap selanjutnya. Proses untuk mempersiapkan dokumen meliputi proses pembersihan dokumen dari karakter-karakter yang tidak diperlukan, proses analisa leksikal teks, proses penghapusan kata *stopword*, dan proses *stemming*. Kesemua proses pembersihan tersebut biasa kita kenal dengan *text mining*.

Identifying Base Cluster

Tahap identifikasi *base cluster* merupakan tahap terpenting dalam algoritma *suffix tree clustering*, karena pada tahap ini akan menghasilkan *cluster-cluster* dasar [Zamir]. Pembentukan *base cluster* dilakukan dengan cara menemukan *share phrase* antar dokumen. Untuk menemukan *share phrase* digunakan struktur data *suffix tree*. Dengan menggunakan stuktur data ini, maka setiap dokumen akan direpresentasikan menjadi suatu kalimat. Untuk menemukan *base cluster* dapat dilakukan dengan cara membuat suatu *invert index* dari phrase untuk semua dokumen.



Setiap *base cluster* yang terbentuk memiliki suatu *score*. Penghitungan *score* merupakan sutu fungsi dari jumlah dokumen yang masuk anggota *base cluster* dan jumlah kata yang menyusun *phrase* dari *base cluster*. Fungsi untuk menghitung *score base cluster* ditunjukkan oleh persamaan (1)

$$S(B) = |B|.f(|P|) \qquad (1)$$

Dimana pada persamaan (1), |B| = jumlah dokumen di dalam base *cluster* B dan |P| = jumlah kata yang menyusun frase P.

Combining Base Cluster

Tahap ini digunakan untuk menangani ovelaping *cluster*. Dalam tahap ini, *phrase* tidak dipertimbangkan. Sebelum melakukan kombinasi antar *base cluster*, kita harus menghitung dulu nilai *similarity* antar *base cluster* yang didasarkan pada jumlah dokumen yang overlap. Adanya overlaping dokumen ini didasarkan karena dokumen memiliki lebih dari satu topik sehingga dokumen dapat memiliki lebih dari satu *phrase* yang di-*share*.

Ukuran nilai *similarity* menggunakan nilai biner. Rumus untuk menghitung nilai *similarity* antar *base cluster* ditunjukkan pada persamaan (2) dan (3).

$$|Bm \cap Bn| / |Bm| > 0{,}5 \qquad (2)$$
$$|Bm \cap Bn| / |Bn| > 0{,}5 \qquad (3)$$

Dimana pada persamaan (2) dan (3) : $|Bm \cap Bn|$ = jumlah dokumen yang overlap terhadap base cluster *Bm dan Bn.* $|Bm|$ *dan* $|Bn|$ = jumlah dokumen dalam base cluster *Bm* dan *Bn.*

Dalam persamaan di atas, menunjukkan penggunaan nilai *threshold* 0,5 karena nilai tersebut merupakan nilai tengah antara 0 sampai 1. Jika persamaan (2) dan (3) bernilai benar maka *similarity* akan bernilai 1 sehingga antara kedua *base cluster* tersebut akan terhubung. Jika salah satu dari persamaan (2) dan (3) bernilai benar atau keduanya bernilai salah maka *similarity* akan bernilai 0 sehingga antara kedua *base cluster* tersebut tidak terhubung.

3. PENGUJIAN DAN ANALISIS

Tokenizing

Tes menggunakan 3 dokumen berikut :

*) Anggodo investigation intensified Lock Snare Anggoro.txt
Corruption Eradication Commission KPK not want to waste Anggodo investigation in cases of alleged Widjojo hinder the investigation of corruption cases. At least two targets that can be expected the eradication of mob law and the entrance through the ...

*) Anggodo Sleeping Beauty in Prison.txt
Anggodo Widjoyo has been named as suspects Corruption Eradication Commission KPK . Its been two nights the brother of fugitive cases of corruption Anggoro Widjoyo it languished as a prisoner at Cipinang Penitentiary. Bonaran dismiss that his client ...

*) Ask the President Bonaran Waive Anggodo.txt
Bonaran Situmeang Widjojo Anggodo legal counsel asked President Susilo Bambang Yudhoyono to free his client from prosecution. Bonaran also considered there has been injustice against his client. He gave an example when the case against the ...

Hasil Tokenizing :

Daftar Kata dari File "Anggodo investigation intensified Lock Snare Anggoro.txt" :
{ 48 kata }

corruption
eradication
commission
kpk
not
want
to
waste
anggodo
investigation
in

Daftar Kata dari File "Anggodo Sleeping Beauty in Prison.txt" :
{ 54 kata }

anggodo
widjoyo
has
been
named
as
suspects
corruption
eradication
commission
kpk

Filtering

Hasil Filtering :

Daftar Kata dari File "Anggodo investigation intensified Lock Snare Anggoro.txt" :
{ 29 kata }

corruption
eradication
commission
kpk
want
waste
anggodo
investigation
cases
alleged
widjojo

Daftar Kata dari File "Anggodo Sleeping Beauty in Prison.txt" :
{ 30 kata }

anggodo
widjoyo
named
suspects
corruption
eradication
commission
kpk
nights
brother
fugitive

Stemming

Hasil Stemming :

Daftar Kata dari File "Anggodo investigation intensified Lock Snare Anggoro.txt" :
{ 29 kata }

corruption -> corrupt
eradication -> eradic
commission -> commiss
kpk -> kpk
want -> want
waste -> waste
anggodo -> anggodo
investigation -> investigate
cases -> case
alleged -> alleg
widjojo -> widjojo

Daftar Kata dari File "Anggodo Sleeping Beauty in Prison.txt" :
{ 30 kata }

anggodo -> anggodo
widjoyo -> widjoyo
named -> name
suspects -> suspect
corruption -> corrupt
eradication -> eradic
commission -> commiss
kpk -> kpk
nights -> night
brother -> brother
fugitive -> fugitive

Tahap Pembentukkan Suffix tree

Pada tahap ini akan dilakukan pengujian terhadap hasil pembentukan *suffix tree.* Pengujian dilakukan dengan memberikan inputan beberapa kalimat. Hasil pengujian dapat dilihat dari *base cluster* yang terbentuk. Beberapa kalimat yang digunakan sebagai uji coba pembentukkan *suffix tree* antara lain:

- Cat ate cheese
- Cat ate mouse too
- Mouse ate cheese too
- Mouse ate fish dead
- Cat ate fish dead too
- Cat play ball

No	Nomor Node	Label Node	Documents
1	1	cheese	file01.txt, file03.txt
2	2	eat	file01.txt, file03.txt, file04.txt, file03.txt, file05.txt
3	3	cat	file01.txt, file04.txt, file05.txt, file06.txt
4	4	too	file02.txt, file03.txt, file05.txt
5	5	mouse	file02.txt, file03.txt, file05.txt
6	6	eat mouse too	file02.txt, file03.txt, file04.txt
7	8	cat eat mouse too	file02.txt, file03.txt, file04.txt
8	9	mouse too cheese	file01.txt, file03.txt
9	11	eat cheese too	file01.txt, file03.txt
10	12	cheese too eat	file01.txt, file02.txt, file03.txt, file04.txt, file05.txt
11	13	mouse eat cheese too	file01.txt, file03.txt
12	15	dead	file04.txt, file05.txt
13	16	fish dead	file04.txt, file05.txt
14	17	eat fish dead	file04.txt, file05.txt
15	18	eat eat	file01.txt, file02.txt, file03.txt, file04.txt, file05.txt
16	19	mouse eat fish dead	file04.txt, file05.txt
17	20	mouse mouse	file02.txt, file03.txt, file04.txt
18	24	eat fish dead too	file04.txt, file05.txt
19	25	fish dead too cat eat	file01.txt, file02.txt, file05.txt
20	26	ball	file06.txt
21	27	play ball	file06.txt
22	29	cat eat	file01.txt, file02.txt, file03.txt, file04.txt, file05.txt

Scoring Cluster

No	Nomor Node	Label Node	|B|	f(|P|)	s(B)
1	1	cheese	2	1	2
2	2	eat	5	1	5
3	3	cat	4	1	4
4	4	too	3	1	3
5	5	mouse	3	1	3
6	6	eat mouse too	3	3	9
7	8	cat eat mouse too	3	4	12
8	9	mouse too cheese	2	3	6
9	11	eat cheese too	2	3	6
10	12	cheese too eat	5	3	15
11	13	mouse eat cheese too	2	4	8
12	15	dead	2	1	2
13	16	fish dead	2	2	4
14	17	eat fish dead	2	3	6
15	18	eat eat	5	2	10
16	19	mouse eat fish dead	2	4	8
17	20	mouse mouse	3	2	6
18	24	eat fish dead too	2	4	8
19	25	fish dead too cat eat	3	5	15
20	26	ball	1	1	1
21	27	play ball	1	2	2
22	29	cat eat	5	2	10

Similarity Base Cluster

Combining Base Cluster :
[View / Hide]

No	Node A	Node B	Status
1	cheese	eat	connected
2	cheese	cat eat	not connected
3	cheese	too	not connected
4	cheese	mouse	not connected
5	cheese	eat cheese	connected
6	eat	cheese	connected
7	eat	cat eat	connected
8	eat	too	connected
9	eat	mouse	connected
10	eat	eat cheese	connected
11	cat eat	cheese	not connected
12	cat eat	eat	connected
13	cat eat	too	not connected
14	cat eat	mouse	not connected
15	cat eat	eat cheese	not connected
16	too	cheese	not connected
17	too	eat	connected

Cluster	Base Cluster
2	1, 3, 4, 5, 6, 8, 9, 11, 12, 13, 18, 20, 29

Uji coba Waktu *Generate Suffix tree*

Pada skenario ini akan dilakukan pengujian terhadap waktu yang dibutuhkan untuk melakukan proses *generate suffix tree*. Pengujian ini didasarkan pada jumlah dokumen yang ingin di*generate suffix tree*nya..

Jml Dokumen	Jml Kata	Jml Base Cluster	Waktu (Detik)
3	30	25	2
5	50	40	13
10	100	72	110
3	90	42	13
5	150	76	129
10	300	169	308
3	150	72	85
5	250	113	309
10	500	262	587
3	300	127	303
5	500	219	524
10	1000	603	1236

4. KESIMPULAN

Setelah melalui tahap implementasi dan uji coba, maka dapat ditarik kesimpulan sebagai berikut:

1. Algoritma *Suffix tree Clustering* dapat diterapkan untuk *clustering* dokumen Berbahasa Inggris
2. Untuk melakukan *clustering* dokumen yang didasarkan pada multiword *phrase* digunakan struktur data *suffix tree*.
3. Untuk pembentukkan *suffix tree* membutuhkan waktu yang lama karena selain tergantung pada jumlah dokumen yang dikoleksi juga tergantung pada jumlah kata untuk setiap dokumen yang ingin diklasifikasikan.

DAFTAR PUSTAKA

Adiwijaya Igg Ph.D. 2006. ***Text Mining and Knowledge Discovery, Kolokium bersama komunitas datamining Indonesia & soft-computing Indonesia***.

Agusetia Usmaida. 2009. ***Web Mining Untuk Pencarian Berdasarkan Kata Kunci Dengan Automatic Clustering***, Tugas Akhir Jurusan Teknik Informatika Politeknik Elektronika Negeri Surabaya.

Agus Zainal Arifin, 2008. ***Klasifikasi Online Dokumen Berita dengan Menggunakan Algoritma Suffix Tree Clustering***. Sesindo 2008 -ITS

A.R. barakbah. 2006. ***Clustering, In. Workshop Data Mining***. Jurusan Teknik Informatika Politeknik Elektronika Negeri Surabaya.

Fast String Searching With Suffix Trees diambil dari http://marknelson.us/1996/08/01/suffix-trees/

Guihong Cao, Dawei Song dan Peter Bruza. 2003**.** ***Suffix Tree Clustering on Post Retrieval Documents***.

Hung Chim dan Xiaotie Deng. 2007. ***A New Suffix Tree Similarity Measure for Document Clustering***. Canada : IW3C2

Irregular Verb Dictionary diambil dari http://www.englishpage.com/irregularverbs/irregularverbs.html

Jon Atle Gulla. 2009. ***Contextualized Clustering in Exploratory Web Search.***

List Of English Stop Words diambil dari http://armandbrahaj.blog.al/2009/04/14/list-of-english-stop-words/

Novan S. 2001. ***Implementasi Aplikasi Information Retrieval Untuk Pendeteksian dan Klasifikasi Berita Kejadian Berbahasa Indonesia Berbasis Web***. Tugas Akhir, Jurusan Teknik Informatika Fakultas Teknologi Informasi ITS Surabaya.

Oren Zamir dan Oren Etzioni. 1998. ***Web Document Clustering A Feasibility Demonstration.***

Samue Sambasivam dan Nick Theodosopoulos. 2006. ***Advances Data Clustering Methods of Mining Web Documents***.

Suffix Trees diambil dari http://www.allisons.org/ll/AlgDS/Tree/Suffix/

Tries and Suffix Trees diambil dari http://www.cs.mcgill.ca/~cs251/OldCourses/1997/topic7/