IMPLEMENTASI DAN ANALISIS INCREMENTAL DOCUMENT CLUSTERING MENGGUNAKAN ALGORITMA COBWEB

Dyah Alifda Prihaningrum[1], Yanuar Firdaus A.w.[2], Shaufiah[3]

[1]Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Clustering dokumen merupakan cara yang tepat untuk mempermudah search engine dalam melakukan query terhadap kumpulan dokumen yang besar. Dokumen-dokumen yang memiliki kesamaan akan dikelompokkan sehingga membentuk topik-topik atau subtopik yang berbeda. Algoritma-algoritma clustering dokumen yang sering dipelajari adalah algoritma-algoritma batch clustering, di mana keseluruhan dokumen diperlukan sejak awal dan clustering dilakukan dengan banyak iterasi terhadap dokumen tersebut. Namun, dengan adanya publikasi online di web yang semakin berkembang seperti sekarang ini, terjadi ledakan jumlah informasi yang bertambah setiap harinya. Metode batch clustering dianggap tidak efisien untuk kasus semacam ini. Agar proses clustering dapat dilakukan segera setelah dokumen masuk, maka clustering perlu dilakukan secara incremental.

Terdapat beberapa algoritma incremental clustering yang populer. Salah satunya adalah algoritma Cobweb yang diimplementasikan pada Tugas Akhir ini. Tugas Akhir ini menggunakan Cobweb untuk mengelompokkan retrieved documents hasil pencarian search engine. Cobweb melakukan clustering data dengan membangun classification tree di mana tiap node dari tree tersebut menggambarkan cluster yang berisi objek-objek data. Dalam membangun tree, Cobweb menggunakan category utility (CU) untuk mengevaluasi tree dan mendapatkan pengelompokan data yang paling tepat. Dari pengujian yang dilakukan pada Tugas Akhir ini, hasil akhir menunjukkan bahwa clustering menggunakan algoritma Cobweb yang diterapkan pada retrieved documents memberikan solusi dengan kualitas yang baik, karena meskipun pada pohon klaster pasti terjadi overlapping, tetap terbukti memiliki sifat kohesif. Kohesif adalah keadaan di mana persamaan antardokumen dalam klaster yang sama lebih besar daripada persamaan antardokumen pada klaster yang berbeda.

Kata Kunci : dokumen, incremental clustering, Cobweb, search engine, retrieved documents , classification tree, category utility, persamaan, kohesif

Abstract

Document clustering is an appropriate way to simplify the search engine performing the query against a large collection of documents. Similar documents will be grouped to form different topics or subtopics. Document clustering algorithms that are often studied are the batch clustering ones, where the entire document is required from the beginning and the clustering is performed by many iterations of each document. However, with the current groeing online publishing on the web, explosion of information is increasing every day. Batch clustering methods are considered inefficient for such cases. In order for the clustering process can be performed immediately after the document signed in, it needs to be done incrementally.

There are several popular incremental clustering algorithms. One of them is the Cobweb algorithm implemented in this final project. Cobweb is used to classify retrieved documents from search results by search engine. Cobweb perform data clustering by building a classification tree where every node of the tree depicts the cluster that contains the data objects. In the tree building, Cobweb uses category utility (CU) to evaluate the tree and get the most appropriate grouping of data. From the testing performed on this final project, final result shows that the Cobweb clustering algorithm implemented on retrieved documents provides solutions with good quality, because despite the inevitable overlapping clusters of trees, still proved to have the cohesiveness characteristic. Cohesive is a state where the similarity between documents in the same cluster is greater than the similarity between documents in different clusters.

Keywords : document, incremental clustering, Cobweb, search engine, retrieved documents ,classification tree, category utility, similarity, cohesive

1. Pendahuluan

1.1 Latar belakang

Dengan berkembangnya teknologi yang ada, maka berkembang pula kebutuhan akan informasi. Begitu banyak dokumen informasi yang tersedia. Sistem pencarian yang konvensional akan memberikan hasil berupa daftar panjang dokumen yang memaksa user untuk menelusuri satu-persatu agar menemukan informasi yang relevan. Hal ini pun masih diterapkan oleh sebagian *search engine* yang ada.[5] Akibatnya, dibutuhkan waktu yang lama untuk menemukan informasi-informasi dengan topik tertentu dari sekian banyak kumpulan dokumen. Untuk itu, perlu dilakukan *clustering* atau pengelompokan dokumen. *Clustering* dokumen merupakan cara yang tepat untuk mempermudah search engine dalam melakukan query terhadap kumpulan dokumen yang besar. Dokumen-dokumen yang memiliki kesamaan akan dikelompokkan sehingga membentuk topik-topik atau subtopik yang berbeda.

Algoritma-algoritma *clustering* dokumen yang sering dipelajari adalah algoritma-algoritma *batch clustering*, di mana keseluruhan dokumen diperlukan sejak awal dan *clustering* dilakukan dengan banyak iterasi terhadap dokumen tersebut. Namun, dengan adanya publikasi *online* di web yang semakin berkembang seperti sekarang ini, terjadi ledakan jumlah informasi yang bertambah setiap harinya. Sebagai contoh, dokumen teks dalam situs-situs pemberitaan selalu bertambah setiap saat. Kemudian muncul ide untuk melakukan *batch clustering* terhadap dokumen yang dikumpulkan secara periodik, baru selanjutnya menggabungkan klaster-klaster yang sudah terbentuk. Metode seperti ini tentu menimbulkan adanya waktu tunda hingga suatu dokumen masuk ke dalam klaster tertentu. Pada kenyataannya, waktu tunda tidak dapat diterima di beberapa bidang tertentu, misalnya bisnis jasa finansial, di mana keputusan perdagangan sangat bergantung pada berita, sehingga akses berita yang cepat pun sangat diperlukan.[7] Agar proses *clustering* dapat dilakukan segera setelah dokumen masuk, maka *clustering* perlu dilakukan secara *incremental.*

Terdapat beberapa algoritma *incremental clustering* yang populer. Salah satunya adalah algoritma Cobweb. Algoritma ini menggunakan konsep hierarkhi dalam proses *clustering*nya. Konsep hierarkhi dalam *clustering* dokumen memberikan dua keuntungan. Pertama, dengan mendeskripsikan hubungan antar grup dokumen akan mempercepat pencarian topik yang spesifik. Kedua, dengan memberikan hierarkhi topik yang berisi dokumen kepada user, user dapat mencari informasi yang diperlukan sesuai level spesifikasi yang diinginkannya.[7] Cobweb melakukan *clustering* data dengan membangun pohon klasifikasi di mana tiap node dari pohon tersebut menggambarkan cluster yang berisi objek-objek data. Dalam membangun pohon, Cobweb menggunakan *category utility* (CU) untuk mengevaluasi tree dan mendapatkan pengelompokan data yang paling tepat.

Pada Tugas Akhir ini akan diteliti bagaimana proses *clustering* dokumen, khususnya dokumen berbentuk teks hasil pencarian *search engine*, menggunakan algoritma Cobweb sebagai algoritma *clustering* incremental dan hierarkhis. Dengan sistem clustering Cobweb yang kokoh, diharapkan dapat menghasilkan solusi klaster yang berkualitas baik dan menyajikan data sesuai kebutuhan user.

1.2 Perumusan masalah

Untuk analisis *incremental document clustering* menggunakan algoritma Cobweb, masalah yang akan diselesaikan dalam Tugas Akhir ini adalah sebagai berikut.

1. Bagaimana efektivitas metode *incremental hierarchical clustering* yang diterapkan algoritma Cobweb melakukan *clustering* dokumen teks?
2. Bagaimana pengaruh jumlah dokumen ter-*retrieve* yang muncul pada *hitlist* dan query inputan user terhadap kualitas klaster yang dihasilkan?

1.3 Batasan Masalah

Adapun batasan-batasan yang diberikan dalam penyelesaian masalah Tugas Akhir ini adalah sebagai berikut.

1. Penelitian pada Tugas Akhir ini hanya terfokus pada *clustering* dokumen menggunakan algoritma COBWEB.
2. Dokumen yang dijadikan objek *clustering* berupa dokumen berbentuk teks yang diambil dari database.
3. Dokumen teks yang digunakan adalah dokumen berbahasa Inggris.
4. Proses *clustering* menggunakan dataset yang berupa dokumen hasil pencarian oleh *search engine*.
5. Hanya membahas metode *clustering* menggunakan algoritma Cobweb, sedangkan text preprocessing menjadi batasan masalah.
6. Metode *clustering* yang digunakan tidak menangani pelabelan.

1.4 Tujuan

Tujuan Tugas Akhir ini adalah sebagai berikut.

1. Membangun aplikasi yang dapat melakukan *clustering* dokumen teks dengan tepat untuk mendapatkan hierarki hitlist hasil pencarian search engine menggunakan algoritma COBWEB.
2. Menguji kualitas klaster yang dibangun dalam memenuhi permintaan informasi dengan mengukur nilai *intrinsic similarity* serta melihat pengaruh jumlah dokumen ter-*retrieve* dan *query* inputan user terhadap perubahan nilainya.

1.5 Metodologi penyelesaian masalah

Metode yang digunakan untuk menyelesaikan permasalahan-permasalahan Tugas Akhir ini terdiri dari langkah-langkah sebagai berikut.

1. Studi Literatur
   Mempelajari dan memahami salah satu algoritma *incremental document clustering* yaitu algoritma COBWEB melalui literatur berupa buku, makalah, atau jurnal dari berbagai media terutama internet.
2. Pengumpulan Dataset
   Pengumpulan dataset yang akan dijadikan sebagai data latih untuk merancang model *clustering* menggunakan algoritma Cobweb.
3. Pembangunan Aplikasi
   Pembangunan aplikasi yang meliputi:
   a. Perancangan Sistem

Menyiapkan dokumen yang akan digunakan untuk *clustering* menggunakan algoritma Cobweb. Selanjutnya dilakukan perancangan sistem yang nantinya dapat mengelompokkan hasil pencarian search engine ke dalam *cluster-cluster* yang tepat menggunakan algoritma Cobweb.

b. Implementasi
Di tahap ini dilakukan implementasi algoritma Cobweb pada dokumen yang diperoleh, yang meliputi:
- Menggunakan search engine untuk mendapatkan *retrieved documents* sebagai inputan bagi *clustering system* yang akan dibangun.
- Memproses *retrieved documents* sehingga menjadi data terstruktur dengan nilai atribut numerik.
- Melakukan *clustering* data menggunakan algoritma Cobweb dengan mengimplementasikannya ke dalam kode program.

c. Pengujian
Sistem aplikasi yang sudah dibangun kemudian diuji untuk mengetahui apakah sistem sudah berjalan seperti yang diharapkan.
1) Untuk menganalisis efektivitas aplikasi, pengujian dilakukan dengan menjalankan aplikasi search engine, kemudian dilakukan beberapa kali percobaan dengan melakukan pencarian berita terkait kata kunci tertentu.
2) Untuk menganalisis pengaruh jumlah dokumen dan jumlah atribut terhadap efektivitas aplikasi, pencarian dilakukan beberapa kali dengan memilih query-query dengan kombinasi panjang query dan jumlah dokumen pada *hitlist* yang representatif.

4. Analisis Hasil
Pada tahap ini dilakukan analisis terhadap hasil *clustering* yang diperoleh dari perancangan sistem menggunakan algoritma Cobweb. Output pengujian aplikasi dianalisis berdasarkan parameter-parameter yang telah ditentukan yaitu *intrinsic similarity* menggunakan H-*score*. Dari hasil analisis tersebut akan diambil kesimpulan mengenai cara kerja algoritma Cobweb dalam *clustering* dokumen teks.

5. Penyusunan Laporan
Pada tahap ini, akan dilakukan penyusunan laporan akhir sekaligus dokumentasi dengan mengikuti kaidah penulisan yang benar dan sesuai dengan ketentuan yang ditetapkan oleh institusi.

5. Kesimpulan dan Saran

5.1 Kesimpulan

1. Perubahan panjang query, yang dalam hal ini term-term query berperan sebagai atribut dalam proses *clustering*, tidak mempengaruhi perubahan nilai kualitas klaster yang dihasilkan.
2. Pertambahan jumlah hitlist bukan penyebab peningkatan kualitas klaster.
3. Pengujian yang dilakukan menunjukkan bahwa solusi *clustering* menggunakan algoritma Cobweb memiliki kualitas yang baik, di mana klaster-klasternya bersifat kohesif dengan nilai ukuran kualitas intrinsik >1.00.
4. Kualitas hitlist pada pengujian ini tidak mempengaruhi kualitas klaster yang dihasilkan.

5.2 Saran

1. Menggunakan dokumen teks berbahasa selain bahasa Inggris dan/atau langsung diambil dari halaman web dalam mengimplementasikan metode *clustering* dengan algoritma Cobweb.
2. Melakukan pengujian dengan jumlah dataset yang lebih besar dan/atau jumlah term atribut yang lebih banyak.
3. Melakukan pengembangan terhadap algoritma agar dapat menangani pelabelan.

Daftar Pustaka

[1]	Agus Zainal Arifin, dan Ari Novan Setiono. Klasifikasi Dokumen Berita Kejadian Berbahasa Indonesia dengan Algoritma Single Pass Clustering. http://www.its.ac.id/personal/files/material/1260-agusza-SITIAKlasifikasiEvent.pdf. Diakses pada 3 Oktober 2009 pukul 15.02 WIB
[2]	Aprilian, Laura. *Term Weighting dengan Metode Savoy pada Information Retrieval*. 2010. Institut Teknologi Telkom Bandung. Bandung.
[3]	Berkhin, Pavel. Survey of Clustering Data Mining Techniques. http://www.ee.ucr.edu/~barth/EE242/clustering_survey.pdf. Diakses 26 November 2009 pukul 17.23 WIB
[4]	Fabian Moerchen, Klaus Brinker, dan Claus Neubauer. Any-time clustering of high frequency news streams. http://www.mybytes.de/papers/moerchen07anytime.pdf. Diakses pada 11 Oktober 2009 pukul 22.36 WIB
[5]	Fisher, Douglas. 1987. Knowledge Acquisition Via Incremental Conceptual Clustering. http://www.springerlink.com/content/qj16212n7537n6p3/fulltext.pdf. Diakses 29 November 2009 pukul 14.22 WIB
[6]	Karhendana, Arie. Pemanfaatan Document Clustering pada Agregator Berita. Institut Teknologi Bandung. http://digilib.itb.ac.id/files/disk1/598/jbptitbpp-gdl-ariekarhen-29899-2-2008ta-1.pdf. Diakses pada 10 Maret 2009 pkl 13.03 wib
[7]	Lintas. Clustering. http://ginageh.wordpress.com/2008/10/28/clustering/. Diakses pada 8 Oktober 2009 pukul 23.32 WIB
[8]	Michael Steinbach, George Karypis, dan Vipin Kumar. A Comparison of Document Clustering Techniques. http://cs.fit.edu/~pkc/classes/ml-internet/papers/steinbach00tr.pdf. Diakses pada 6 Oktober 2009 pukul 16.16 WIB
[9]	Miswan. Pemanfaatan Analisis Gugus (Cluster Analysis) Pada Sistem Temu Kembali Informasi Berbasis Internet.Pusat Pengembangan Teknologi Informasi dan Komputasi - BATAN. http://www.batan.go.id/ppin/lokakarya/LKSTN_13/Miswan.pdf. Diakses pd 10 Maret 2009 pkl 13.04 wib
[10]	Oren Zamir, dan Oren Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results. http://www.cs.washington.edu/research/projects/WebWare1/etzioni/www/papers/www8.pdf. Diakses pada 29 September 2009 pukul 15.48 WIB
[11]	Roberto Basili, Maria Teresa Pazienza, dan Paola Velardi. Hierarchical Clustering of Verbs. http://acl.ldc.upenn.edu/W/W93/W93-0107.pdf. Diakses pada 11 Oktober 2009 pukul

	22.36 WIB
[12]	Sahoo, Nachiketa. 2006. Incremental Hierarchical Clustering of Text Documents. http://www.andrew.cmu.edu/user/nsahoo/draft6.pdf. Diakses pada 11 Oktober 2009 pukul 21.41.
[13]	Wikipedia. Cobweb (clustering). http://en.wikipedia.org/wiki/Cobweb_(clustering). Diakses pada 4 Oktober 2009 pukul 22.59.
[14]	Wikipedia. Conceptual clustering. http://en.wikipedia.org/wiki/Conceptual_clustering. Diakses pada 6 Oktober 2009 pukul 21.57 WIB
[15]	Wikipedia. 2009. Information retrieval. http://en.wikipedia.org/wiki/Information_retrieval. Diakses pada 7 Oktober 2009 pukul 22.13 WIB.
[16]	Wikipedia. 2009. Precision and recall. http://en.wikipedia.org/wiki/Precision_and_recall. Diakses pada 7 Oktober 2009 pukul 22.14 WIB.