INTEGRASI PERINGKAS DOKUMEN OTOMATIS DENGAN ALGORITMA LATENT SEMANTIC ANALYSIS (LSA) PADA PERINGKAS DOKUMEN OTOMATIS UNTUK PROSES CLUSTERING DOKUMEN

Ardytha Luthfiarta[1], Junta Zeniarja[2], Abu Salam[3]
[1,2,3] Fakultas Ilmu Komputer, Teknik Informatika, Univ. Dian Nuswantoro
Email: ardytha.luthfiarta@dsn.dinus.ac.id[1], junta@dsn.dinus.ac.id[2], abu.salam@dsn.dinus.ac.id[3]

Abstrak

Teknologi pengklasteran dokumen memiliki peran yang signifkan dalam kemajuan teknologi informasi, diantaranya mempunyai peranan penting dalam pengembangan web di bidang akurasi kategorisasi keyword otomatis pada search engine, kategorisasi berita untuk surat kabar elektronik, peningkatan rating situs dengan teknologi Search Engine Optimization (SEO) dan sangat memungkinkan untuk diimplementasikan dalam berbagai teknologi informasi lainnya, oleh karena itu diperlukan penelitian untuk meningkatkan ketepatan akurasi dalam pengklasteran dokumen. Dalam penelitian ini Algoritma Latent Semantic Analysis (LSA) dapat melakukan proses reduksi kalimat dengan lebih baik dibandingkan algoritma Feature Based sehingga mendapatkan hasil akurasi proses clustering dokumen yang lebih akurat. Beberapa tahapan clustering dalam penelitian ini, yaitu preprocessing, peringkas dokumen otomatis dengan metode fitur, peringkas dokumen otomatis dengan LSA, pembobotan kata, dan algoritma clustering. Hasil penelitian menunjukkan tingkat akurasi menggunakan peringkas dokumen otomatis dengan LSA dalam proses clustering dokumen mencapai 71,04 % yang diperoleh pada tingkat peringkas dokumen otomatis dengan LSA 40% dibandingkan dengan hasil clustering tanpa peringkas dokumen otomatis yang hanya mencapai tingkat akurasi 65,97 %.

Kata kunci: *Text Mining, Clustering, Peringkas Dokumen Otomatis, LSA*

Abstract

Document clustering techonology has a significant role in the advancement of information technology, such as an important role for web development in the field of automatic keyword categorization accuracy on search engine, news classification for electronic newspaper, improvement of site rank using Search Engine Optimization (SEO) technology and enable to be implemented in various information technology, therefore is needed a research to improve accuracy in document clustering. In this research, Latent Semantic Analysis (LSA) algorithm can do sentence reduction process better than Feature Based algorithm so could be resulted in more accurate document clustering process. Several steps of clustering in this research are preprocessing, automatic document compression using feature method, automatic document compression using LSA, word weighting and clustering algorithm. The result of this research shows accuracy rating for automatic document compression using LSA in document clustering processing get the rating of 71,04% that was obtained on automatic document compression with LSA of 40% compared with clustering without automatic document compression that is only get the accuracy rating of 65,97%.

Keywords: *Text Mining, Clustering, Automatic Document Compression, LSA*

1. PENDAHULUAN

Proses peringkas dokumen adalah sebuah proses untuk melakukan pengurangan volume dokumen menjadi lebih ringkas, dengan cara mengambil inti dokumen dan membuang term yang dianggap tidak penting tanpa mengurangi makna sebuah dokumen.[1][2], terdapat dua tipe pembuatan suatu ringkasan yang mengambil bagian terpenting dari teks aslinya yaitu abstrak dan ekstrak. Abstrak menghasilkan sebuah interprestasi terhadap teks aslinya, dimana sebuah kalimat akan ditransformasikan menjadi kalimat yang lebih singkat[3], sedangkan ekstraksi merupakan ringkasan teks yang diperoleh dengan menyajikan kembali bagian tulisan yang dianggap topik utama tulisan dengan bentuk yang lebih disederhanakan [4][5], dalam penelitian ini akan digunakan fitur ringkasan ekstrak sebagai model peringkas dokumen otomatis.

Penerapan teknik peringkasan dokumen untuk clustering dokumen memiliki dampak yang signifikan, hal ini dikarenakan proses clustering dokumen seringkali terkendala oleh besarnya volume dokumen yang ada. Permasalahan itu muncul karena volume dokumen yang besar identik dengan besarnya matrik term-dokumen, padahal tidak semua term relevan dan terkadang muncul term-redundan dan hal inilah yang menyebabkan proses clustering menjadi tidak optimal [6]. Penelitian ini bertujuan untuk optimalisasi proses clustering dokumen dengan melakukan reduksi matrik term-dokumen.

Di dalam model peringkas dokumen otomatis dapat digunakan algoritma *Feaure Based dan Latent Semantic Analysis (LSA)* untuk proses reduksi kalimat[7]. Penelitian yang sudah pernah dilakukan dengan menggunakan algoritma *Feature Based* dalam proses peringkas dokumen otomatis sebagai *feature reduction* untuk proses *clustering* dokumen dihasilkan tingkat akurasi yang lebih baik dibandingkan dengan proses clustering menggunakan teknik *feature reduction* standar [8][9]. Peringkas Dokumen menggunakan Algoritma LSA diharapkan dapat melakukan proses reduksi kalimat dengan baik dibandingkan algoritma *Feature Based* sehingga dapat lebih meningkatkan akurasi proses *clustering* dokumen.

Clustering dokumen adalah proses pengelompokan dataset dokumen merujuk pada *similarity* (kemiripan) pola data dokumen ke dalam suatu cluster, sedangkan yang tidak memiliki kemiripan akan dikelompokkan ke dalam cluster yang lain.[9]. K-means merupakan salah satu algoritma klaster yang paling terkenal dan sering digunakan untuk menyelesaikan permasalahan clustering yaitu dengan mengelompokkan sejumalah k cluster (dimana jumlah k telah di definisikan sebelumnya) [10].

Langkah-langkah algoritma *K-means* adalah sebagai berikut:

1. Tentukan nilai k sebagai jumlah klaster yang ingin dibentuk
2. Bangkitkan k centroid (titik pusat klaster) awal secara random.
3. Hitung jarak setiap data ke masing-masing centroid menggunakan rumus korelasi antar dua objek yaitu

Euclidean Distance dan kesamaan Cosine.

4. Kelompokkan setiap data berdasarkan jarak terdekat antara data dengan centroidnya.
5. Tentukan posisi centroid baru (k C) dengan cara menghitung nilai rata-rata dari data-data yang ada pada centroid yang sama.

$$C_k = \left(\frac{1}{n_k}\right)\sum d_i$$

(1)

Dimana k n adalah jumlah dokumen dalam cluster k dan i d adalah dokumen dalam cluster k.

6. Kembali ke langkah 3 jika posisi centroid baru dengan centroid lama tidak sama.

$$Sim(d_x,d_y) = \frac{\sum_{k=1}^{n} x_k \times y_k}{\sqrt{\sum_{k=1}^{n} x_k{}^2} \times \sqrt{\sum_{k=1}^{n} y_k{}^2}}$$

(2)

2. METODE PENELITIAN

2.1 Tahap Preprocessing

Tahapan *preprocessing* adalah tahapan awal sebelum dilakukan proses clustering, tahapan ini diperlukan agar dokumen hasil crawling, yang akan diproses berada dalam bentuk yang tepat dan dapat diproses pada tahapan selanjutnya. Penelitian ini menggunakan tiga tahap untuk preprocessing, yaitu : *tokenization, stopword, dan stemming*.

2.2 Peringkasan Teks Dokumen Otomatis *(Automatic Text Summarization)*

Peringkas dokumen teks otomatis adalah bentuk ringkas dari dokumen, yang bertujuan untuk menghilangkan term yang dianggap tidak relevan atau redundan dengan menjaga inti makna dari dokumen, sehingga meskipun dokumen tadi memiliki volume yang besar akan tetapi para pengguna dokumen dapat memahami inti maknanya dengan cepat dan benar [11][12].

2.3 Metode Berbasis Fitur

Dalam penelitian ini ada beberapa tahapan metode berbasis fitur yang digunakan, yaitu sebagai berikut :

a. Fitur Judul
Skor(S_i) =
$$\frac{Jumlahkatapadajudul}{Jumlahkatayangsamadenganjudul}$$
(3)

b. Panjang Kalimat
Skor(S_i) =
$$\frac{Jumlahkatapadakalimat}{jumlahkatapadakalimatterpanjang}$$
(4)

c. Bobot Kata
Skor(S_i) =
$$\frac{Jumlah\ TF-IdF\ dalam\ kalimat}{Maksimal\ jumlah\ TF-IdF}$$
(5)

TF-IDF = *jumlah kata pada dokumen * idf*
= *jumlah kata pada dokumen * $log\left(\frac{df}{N}\right)$*
df = jumlah kalimat yang mengandung kata x
N = jumlah kalimat dalam pada dokumen

d. Posisi Kalimat
Skor(S_i) = 1 merepresentasikan kalimat pertama dan kalimat terakhir. 0 merepresentasikan kalimat lainnya.

e. Kesamaan Antar Kalimat

$$sim_{cos}(d_i,d_j) = \frac{d_i \,.\, d_j}{||d_i||.\ ||d_j||} \qquad (6)$$

$$= \frac{\sum_{k=1}^{n} w_{ik} \, X \, w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2} \, X \, \sqrt{\sum_{k=1}^{n} w_{jk}^2}} \qquad (7)$$

w_{ik} = Bobot kata pada dokumen
$_{wjk}$ = Bobot kata pada query

sedangkan untuk menghitung skor dari fitur ini adalah [4] :

$$\text{Skor}(S_i) = \frac{\text{jumlah cosine similarity}}{jumlah\ maksimal\ similarity} \qquad (8)$$

f. Kata Tematik

$$\text{Skor}(S_i) = \frac{jumlah\ kata\ tematik\ dalam\ kalimat}{panjang\ kalimat(jumlah\ kata\ pada\ kalimat)} \qquad (9)$$

g. Data Numerik

$$\text{Skor}(S_i) = \frac{jumlah\ data\ numerik}{panjang\ kalimat\ (jumlah\ kata\ pada\ kalimat)} \qquad (10)$$

2.4 Metode Berbasis LSA (Latent Semantic Analysis)

LSA *(Latent Semantic Analysis)* adalah metode statistik aljabar yang mengekstrak struktur semantik yang tersembunyi dari kata dan kalimat [7], untuk mencari interelasi diantara kalimat dan kata, digunakan metode aljabar Singular Value Decomposition (SVD). Disamping mempunyai kapasitas relasi model diantara kata dan kalimat, SVD ini mempunyai kapasitas reduksi noise yang membantu untuk meningkatkan akurasi [8][13].

2.5 Document Representation Vector Space Model

VSM mengubah koleksi dokumen kedalam matrik *term-document* [9]. Pada gambar 1. Dimana *d* adalah dokumen dan *w* adalah bobot atau nilai untuk setiap term.

$$A_{mxn} = \begin{matrix} d1 & d2 & \downarrow & dn & \downarrow \\ \downarrow & \downarrow & & \downarrow & \\ \end{matrix} \begin{bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2n} \\ \vdots & \cdot & & \vdots \\ \omega_{m1} & \omega_{m2} & \cdots & \omega_{mn} \end{bmatrix} \begin{matrix} \leftarrow t1 \\ \leftarrow t2 \\ \leftarrow \\ \leftarrow tm \end{matrix}$$

Gambar 1. Matrik Term-dokumen

2.6 TFIDF

Penelitian ini menggunakan TFIDF sebagai metode *term weighting*. TF adalah jumlah munculnya suatu *term* dalam suatu dokumen, *IDF* adalah perhitungan logaritma pembagian jumlah dokumen dengan frekuensi dokumen yang memuat suatu *term*, dan *TFIDF* adalah hasil perkalian nilai *TF* dengan *IDF* untuk sebuah term dalam dokumen. Persamaan IDF dan TFIDF dapat dilihat pada persamaan 10 dan 11 dibawah ini:

$$IDF = log \frac{D}{DF} \qquad (10)$$

$$TFIDF(t) = TF * log \frac{D}{DF} \qquad (11)$$

2.7 Similiarity Measure

Dalam penelitian ini untuk menghitung persamaan antar dokumen akan mengukur jarak antar 2 dokumen d_i dan d_j, dengan menggunakan rumus *cosines similiarity*. Pada Vector Space Model Dokumen direpresentasikan dalam bentuk $d = \{w_1, w_2, w_3, \ldots, w_n\}$ dimana *d* adalah dokumen dan *w* adalah nilai bobot setiap term dalam dokumen[14]. Persamaan similarity measure dapa dilihat pada persamaan 12 berikut ini :

$$similiarity\ (d_i, d_j) = cosines\ \theta = \frac{\vec{di}.\vec{dj}}{||di||.||dj||} \qquad (12)$$

2.8 Evaluation Measure

Ada beberapa teknik evaluation measure untuk mengukur kualitas

performa dari model clustering dokumen, diantaranya adalah information metrix, misclassification index, purity, F-Measure. Penelitian ini menggunakan teknik F-measure untuk mengukur kinerja model yang diusulkan. Pengukuran F-Measure berdasar pada nilai Precision dan Recall. Semakin tinggi nilai Precision dan Recall maka menunjukkan tingat akurasi tinggi hasil clustering dokumen..Recall dan precision kategori *i* dalam cluster *j* diperoleh dari persamaan 13 berikut :

$$Recall\ (i,j) = \frac{nij}{ni} \qquad Precision\ (i,j) = \frac{nij}{nj} \quad (13)$$

n_{ij} = jumlah dokumen kategori *i* dalam cluster *j*,
n_i = jumlah dokumen dalam kategori *i*
n_j = jumlah dokumen dalam cluster *j*

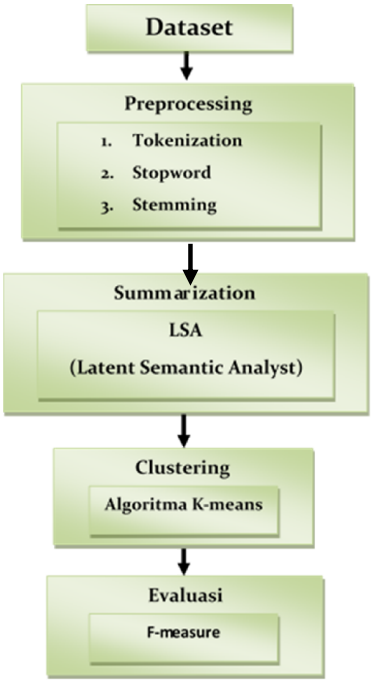
Perhitungan F-measure menggunakan persamaan sebagai berikut:

$$F(i,j) = \frac{2*(Precision*Recall)}{(Precision+Recall)} \quad (14)$$

Dan, rata-rata perhitungan F-Measure menggunakan persamaan berikut:

$$F = \sum_i \frac{ni}{n} \ max_{j=1,\ldots,k} \ F(i,j) \quad (15)$$

Max {F(i,j)} = nilai maksimum F-Measure dari kategori i dalam cluster j



Gambar 2. Model yang diusulkan

Algoritma yang diusulkan akan diimplementasikan secara umum dengan menggunakan pemrograman JAVA. Sistem akan dibangun menggunakan *Lucene3* sebagai *java library*. *Lucene* memiliki fungsi *stopword removal* dan *stemming* sebagai *preprocessing*, perhitungan pembobotan *Term Frequency Invers Document Frequency* (TFIDF) dan perhitungan *cosines similarity* untuk menghitung kemiripan antar dokumen, selain itu *lucene* secara luas sudah diakui dalam penggunaannya untuk mesin pencari dan situs pencarian. Keunggulan lainnya adalah lucene merupakan software library yang open source.

2.9 Dataset

Penelitian ini memakai data yang berasal dari situs portal berita yahoo news Indonesia, jumlah dataset test sebanyak 150 dokumen berita berbahasa indonesia dari 5 kategori berita yaitu: Sport, Ekonomi, Hukum, Kriminal, dan Politik. Dataset tersebut di-

transformasi untuk mendapatkan atribut yang relevan dan sesuai dengan format input algoritma clustering dokumen.

2.10 Preprocessing

Di dalam penelitian ini menggunakan 3 tahapan preprocessing yang akan di gunakan yaitu: Tokenization, Stopword, dan Stemming.

a. Tokenization

Tahap tokenizing adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya, contoh dari tahapan ini adalah sebagai berikut :

Teks Input : "Belajar membaca buku".

Hasil Token : Belajar
membaca
buku

b. Stopword

Dalam tahap stopword, kata-kata yang tidak relevan dalam penentuan topic sebuah dokumen akan dihilangkan, misal kata "adalah", "dari", "sebuah", "atau" dan lain-lain dalam dokumen bahasa Indonesia.

c. Stemming

Steming merupakan tahap mencari root kata / kata dasar dari tiap kata hasil filtering, contoh dari tahap ini adalah sebagai berikut :

Hasil Filter : Belajar
membaca
buku

Hasil Stemming : ajar
baca
buku

2.11 Evaluasi

Evaluasi dilakukan dengan mengamati hasil clustering dari pengujian metode yang diusulkan dengan algoritma LSA (Latent Semantic Analysis). Dalam penelitian ini, digunakan F-measure untuk mengukur kinerja clustering. F-measure diperoleh dari pengukuran recall dan precision. Recall adalah rasio dokumen yang relevan yang terambil dengan jumlah seluruh dokumen dalam koleksi dokumen, sedangkan precision adalah rasio jumlah dokumen relevan terambil dengan seluruh jumlah dokumen terambil. Validasi hasil dengan membandingkan hasil evaluasi metode yang diusulkan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil akurasi kinerja clustering

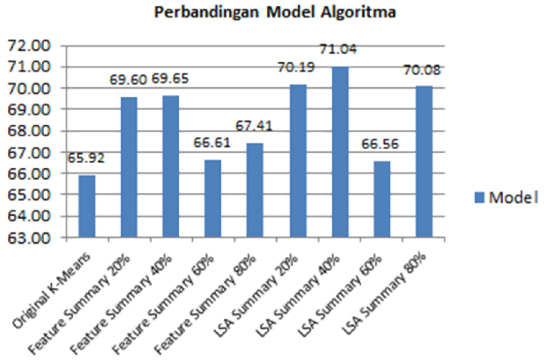
Tabel 1: Hasil penelitian

Metode	F-Measure 1	F-Measure 2	F-Measure 3	F-Measure 4	F-Measure 5	Rata-rata	%
Original K-Means	0.62400	0.68800	0.65333	0.60533	0.72533	0.65920	65.92
Feature base 20%	0.67733	0.81867	0.67200	0.66400	0.64800	0.69600	69.60
Feature base 40%	0.71733	0.79200	0.67467	0.61333	0.68533	0.69653	69.65
Feature base 60%	0.65867	0.74400	0.60800	0.65067	0.66933	0.66613	66.61
Feature base 80%	0.71200	0.67467	0.63733	0.65600	0.69067	0.67413	67.41
LSA 20%	0.71200	0.68000	0.68800	0.74933	0.68000	0.70187	70.19
LSA 40%	0.70400	0.66933	0.76267	0.71200	0.70400	0.71040	71.04
LSA 60%	0.67733	0.65333	0.63200	0.72800	0.63733	0.66560	66.56
LSA 80%	0.82667	0.66667	0.70667	0.63200	0.67200	0.70080	70.08

Tabel 1 diatas merupakan perbandingan hasil penelitian dari beberapa model yang diuji dan model yang diusulkan. Dari hasil penelitian yang dilakukan dapat dibuktikan bahwa rata-rata hasil proses clustering dokumen menggunakan model yang diusulkan yaitu peringkas dokumen otomatis dengan metode *Latent Semantic Analysis (LSA)* dapat meningkatkan akurasi hasil *clustering* pada dokumen teks berbahasa Indonesia.

Tingkat akurasi rata-rata tertinggi diperoleh menggunakan peringkas dokumen otomatis dengan metode LSA mencapai 71,04 % yang diperoleh pada tingkat peringkas dokumen otomatis LSA 40% dibandingkan dengan tanpa

menggunakan peringkas dokumen otomatis yang hanya mencapai rata-rata tingkat akurasi 65,92 %, dari gambar diatas juga dapat dilihat hasil rata-rata proses clustering dokumen dengan menggunakan teknik peringkas dokumen otomatis secara keseluruhan mengalami peningkatan kinerja dari pada kinerja clustering dokumen tanpa menggunakan teknik peringkas dokumen otomatis. Hasil penelitian lebih lengkap dapat dilihat pada Gambar 3 dibawah ini:



Gambar 3. Rata-rata hasil kinerja proses clustering dokumen

Grafik batang pada gambar 3 diatas menunjukkan bahwa secara keseluruhan untuk beberapa pengujian, menunjukkan model peringkas dokumen otomatis menggunakan algoritma LSA menghasilkan akurasi hasil clustering yang lebih baik dibandingakan dengan algoritma Feature based. Dan hasil terbaik ditunjukkan oleh model peringkas dokumen LSA dengan tingkat % summary 40% yaitu menghasilkan nilai F-measure rata-rata sebesar 71,04%.

4. KESIMPULAN

Berdasarkan percobaan-percobaan yang telah dilakukan dapat disimpulkan bahwa Peringkas Dokumen Otomatis dengan *Latent Semantic Analysis (LSA)* pada Proses *Clustering* Dokumen Teks Berbahasa Indonesia dapat meningkatkan kinerja *clustering* dokumen lebih baik dari pada Peringkas Dokumen Otomatis dengan Metode Fitur dan Proses *Clustering* Dokumen Standar, mengalami peningkatan dari tingkat akurasi 65,92 % untuk proses *clustering* standar menjadi 71,04% untuk proses *clustering* dokumen menggunakan peringkas dokumen otomatis dengan *Latent Semantic Analysis (LSA).*

DAFTAR PUSTAKA

[1] Mohammed Abdul Wajeed, & Adilakshmi, T., "Text Classification Using Machine Learning," *Journal of Theoretical and Applied Information Technology, 119-123. 2009.*

[2] S. Catur, S. Abu, and S. Abdul, "Integrating Feature-Based Document Summarization as Feature Reduction in Document Clustering, "*Proceedings of International Conference on Information Technology and Electrical Engineering,* July 2012, pp. 39-42.

[3] Changqiu Sun, Xiaolong Wang & Jun Xu, "Study on Feature Selection in Finance Text Categorization," *International Conference on Systems, Man, and Cybernetics Proceedings of the 2009 IEEE.*

[4] H. Al-mubaid and A.S. Umair, "A new text categorization technique using distributional clustering and learning logic," *IEEE Trans. Knowl. Data Eng*, vol. 18, 2006, pp. 1156-1165.

[5] Ladda Suanmali, Naomie Salim & M Salem Binwahlan, " Automatic text summarization using feature based fuzzy extraction," *Jurnal*

teknologi Maklumat jilid 20. Bil 2, 2008.

[6] Luying LIU, Jianchu KANG, Jing YU & Zhongliang WANG, "A Comparative Study on Unsupervised Feature Selection Methods for Text Clustering," *Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE '05. Proceedings of 2005 IEEE International Conference on.*

[7] R. Peter, S. G, D. G, & S. Kp, "Evaluation of SVD and NMF Methods for Latent Semantic Analysis," *InternationalJournal of Recent Trends in Engineering*, vol. 1, 2009, pp. 308-310.

[8] Tao Liu, Shengping Liu, Zheng Chen & Wei-Ying Ma, "An Evaluation on Feature Selection for Text Clustering," *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003),* Washington DC, 2003.

[9] L. Muflikhah & B. Baharudin, "Document Clustering using Concept Space and Cosine Similarity Measurement," *International Conference on Computer Technology and Development*, Kota Kinabalu: 2009, pp. 58 - 62.

[10] W. Song and S. C. Park, "A Novel Document Clustering Model Based on Latent Semantic Analysis," pp. 539–542, 2007.

[11] Krysta M. Svore, Lucy V., & Christopher J.C. Burges, "Enhancing Single-document Summarization by Combining RankNet and Third-party Sources," *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 448–457, Prague, June 2007.*

[12] JIANG Xiao-Yu, FAN Xiao-Zhong, Wang Zhi-Fei & Jia Ke-Liang, "Improving the Performance of Text Categorization using Automatic Summarization," *International Conference on Computer Modeling and Simulation IEEE 2009.*

[13] Rakesh Peter, Shivapratap G, Divya G & Soman KP, "Evaluation of SVD and NMF Methods for Latent Semantic Analysis," *International Journal of Recent Trends in Engineering ,Vol 1, No. 3, May 2009.*

[14] Anna Hung, "Similarity Measures for Text Document Clustering," *NZCSRSC 2008, April 2008, Christchurch, New Zealand.2008.*