Klasifikasi Teks Bahasa Indonesia Pada Corpus Tak Seimbang Menggunakan NWKNN

Achmad Ridok[1)], Retnani Latifah [2)]
Filkom Unibraw
Jl. MT. Haryono 169 Malang/
e-mail: [1)]acridokg@gmail.com, [2)]retnani.latifah@gmail.com

Abstrak

Pada umumnya pengklasifikasian teks mengasusumsikan data latih terdistribusikan sama rata pada setiap kategori. Padahal pada kenyataannya kumpulan data yang tidak seimbang sering kali muncul. Penelitian ini mencoba menerapkan algoritma Neighbor Weighted K-Nearest Neighbor (NWKNN) untuk mengklasifikasikan dokumen teks berbahasa Indonesia dengan corpus yang tidak seimbang. Algoritma ini memberikan bobot yang kecil untuk tetangga – tetangga yang berasal dari kategori yang besar dan memberikan bobot yang besar pada tetangga – tetangga yang berasal dari kategori yang kecil. Hasil uji coba sistem pada 785 dokumen dari 8 kategori dengan rasio dokumen latih 75% dan dokumen uji 25% serta exponent = 2 dan threshold document frequency = 5, menunjukkan nilai rata - rata precision, recall dan F1 measure masing-masing sebesar 84.2%, 86.7% dan 84.3%. Hasil perbandingan dengan algoritma KNN menunjukkan nilai recall dan *F1 measure NWKNN masing-masing lebih tinggi sebesar 6.9% dan 2.6% daripada nilai recall dan F1 measure KNN.*

Kata kunci: *KNN, NWKNN. Data tidak seimbang*

1. Pendahuluan

Klasifikasi teks atau kategorisasi teks merupakan proses yang secara otomatis menempatkan dokumen teks ke dalam suatu kategori berdasarkan isi dari teks tersebut [7]. Penelitian mengenai klasifikasi teks telah banyak dikembangkan dengan berbagai macam metode yang secara umum terbagi menjadi 3 kelompok yaitu yang pertama adalah klasifikasi teks berbasis statistic seperti *Naive Bayes*, *K-Nearest Neighbor*, *Category Center Vector*, *Support Vector Machine* dan *Maximum Entropy Model*. Kelompok yang kedua adalah klasifikasi teks berbasis koneksi, misalnya menggunakan *Artificial Neural Network*. Kelompok yang terakhir adalah klasifikasi teks berbasis aturan (*rule-based*), seperti *Decision Tree*. Berdasarkan penelitian oleh Yang Yiming dan Xin Liu pada tahun 1999, metode klasifikasi berbasis statistik, terutama KNN dan SVM terbukti memiliki kinerja yang lebih baik dibandingkan lainnya.

Kebanyakan penelitian dengan metode–metode klasifikasi standar mengasumsikan bahwa data latih terdistribusi merata pada setiap kategori. Padahal pada kenyataannya sering dijumpai jumlah data latih yang tidak seimbang. Pada kumpulan data yang tidak seimbang akan ada kelas mayoritas dan kelas minoritas. Ketika sebuah pengklasifikasi teks dihadapkan pada *corpus* yang seperti ini sering kali kinerja klasikasi dari pengklasifikasi tersebut mengalami penurunan [5].

Penyelesaian masalah distribusi data tidak simbang ini telah diusulkan penyelesainya oleh Tan (2005) yakni metode *Neighbor-weighted K-Nearest Neighbor* (NWKNN) yang diterapkan pada *corpus* berbahasa Inggris dan berbahasa Mandarin. Pada metode ini, *neighbors* (tetangga) yang berasal dari kelas mayoritas akan diberi bobot yang kecil sedangkan *neighbors* dari kategori minoritas akan diberi bobot yang besar. Hasil dari penelitian ini menyebutkan bahwa kinerja klasifikasi dengan metode NWKNN terjadi peningkatan yang signifikan daripada metode KNN [5].

Berawal dari fakta ini, pada penelitian metode NWKNN digunakan untuk mengklasifikasikan teks berbahasa Indonesia dengan *corpus* yang tidak seimbang serta menggunakan metode frekuensi dokumen untuk proses pemilihan fiturnya. Setiap *term* dihitung nilai *Document Frequency*-nya sebagai *threshold* dasar pemilihan fitur. Metode pemilihan fitur ini digunakan karena selain sederhana juga memiliki waktu komputasi yang rendah [6].

Sistematika paper ini disusun sebagai berikut. Pada bagian pendahuluan dipaparkan latar belakang mangapa penelitian ini dilakukan, selanjutnya pada bagian tinjuaun pustaka dibahas studi pustaka dan beberapa penelitian terkait. Paper akan ditutup dengan kesimpulan dan saran setelah dibahas metode penelitian dan uraian hasil penelitian.

2. Metodologi Penelitian

2.1. Langkah Penelitian

Langkah – langkah yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Mengumpulkan *corpus* berita berbahasa Indonesia dalam bentuk file **.txt* untuk digunakan sebagai data latih dan data uji.
2. Mengembangkan program pengklasifikasi menggunakan metode *Neighbor-Weighted K-Nearest Neigbor*.
3. Melakukan evaluasi hasil klasifikasi yang dihasilkan oleh sistem menggunakan *precision*, *recall* dan *F1 measure*.

2.2. Deskripsi Data

Data yang digunakan baik data latih maupun uji berasal dari dokumen berita yang yang diunduh dari www.kompas.com pada periode Oktober 2011. Kategori yang digunakan adalah kategori nasional, bisnis dan keuangan, olahraga, megapolitan, sains, travel, edukasi dan oase. Jumlah keseluruhan dokumen sebanyak 785 dokumen yang dibagi menjadi 75% dokumen latih dan 25% dokumen uji dengan jumlah dokumen pada tiap – tiap kategori tidak sama sebagaimana dapat dilihat pada tabel 1.

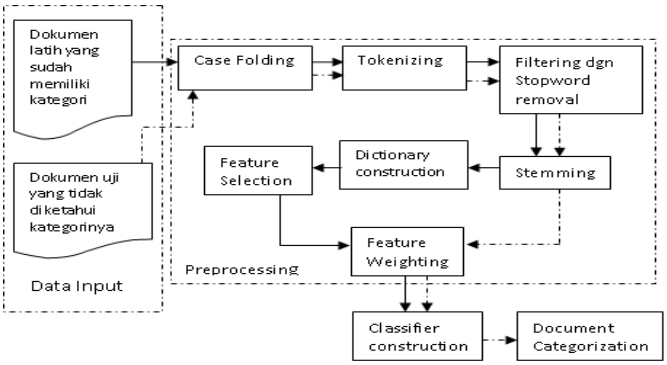
Tabel 1. Jumlah Dokumen

Kategori	**#Dokumen**
Nasional	291
Sains	14
Megapolitan	156
Edukasi	23
Bisnis dan keuangan	136
Travel	49
Oase	32
Olahraga	84
Total	**785**

2.3. Arsitektur Sistem

Klasifikasi teks merupakan proses penentuan kategori suatu dokumen teks sesuai dengan karakteristik dari teks tersebut. Dalam prosesnya klasifikasi teks terdiri dari 3 komponen yaitu praproses data, konstruksi pengklasifikasi dan pengkatagorian dokumen. Tahapan praproses terdiri dari *case folding*, *tokenizing*, *filtering* dan *stemming*. Selain itu pembentukan kamus, pemilihan fitur dan pembobotannya merupakan bagian dari tahap ini [2]. Tahap pembentukan pengklasifikasi adalah suatu tahap pembentukan model pengklasifikasi melalui proses pembelajaran terhadap data latih. Sedangkan pengkatagorian documen adalah suatu tahapan proses testing dari data uji atau data yang akan ditentukan katagorinya berdasarkan model pengklasifikasi yang telah diperoleh.

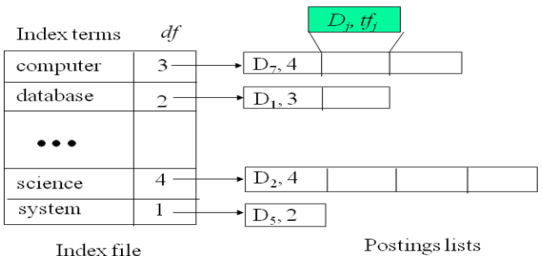
Secara umum arsitektur sistem dapat digambarkan sebagaimana gambar 1. Alur prosesnya dapat dijelaskan sebagai berikut :



Gambar 1. Arsitektur Sistem

Semua dokumen baik dokumen latih maupun dokumen uji dilakukan *preprocessing* yang meliputi case *folding*, *tokenizing*, *filtering*, *stemming*, *dictionary construction* / pembentukan kamus,

feature selection / pemilihan fitur dan *feature weighting* / pembobotan. Setelah proses tokenisasi dan filtering dari kata stoplist selanjutnya dialakukan proses stemming untuk mendapatkan kata dasarnya. Proses stemming pada penelitian ini menggunakan algoritma *Enhanced Confix Stripping Stemmer* yang merupakan pengembangan dari algoritma *Confix Stripping Stemmer* (CS *Stemmer*) yang dikembangkan oleh Arifin, Mahendra, dan Ciptaningtyas yang digunakan untuk melakukan *stemming* bahasa Indonesia. [1]. Kumpulan term hasil dari rangkaian proses ini selanjutnya akan disimpan dalam suatu struktur data *inverted index* sebagaimana gambar 2.



Gambar 2. Struktur data *inverted index*

Pembobotan masing-masing fitur dilakukan setelah pemilihan fitur berdasarkan batas minimal keberadaan term tersebut dalam korpus (DF). Metode yang digunakan untuk pembobotan masing-masing fitur adalah tf-idf sebagaimana persamaan 1. Dimana tfij frekuensi term ke-I pada dokumen latih ke j, sedangkan N dan dfi masing-masing menyatakan banyaknya dokumen latih dan banyaknya dokumen yang mengandung term ke-i.

$$w_{ij} = tf_{ij} \cdot log_2\left(\frac{N}{df_i}\right) \quad (1)$$

Semua fitur yang telah diberi bobot untuk dokumen latih dan dokumen uji direpresentasikan sebagai matrik term-dokumen. Selanjutnya untuk menentukan kesamaan antara suatu dokumen uji (q) dan dokumen latih ke j digunakan cossine similarity sebagaimana persamaan 2. Dalam hal ini t merupakan jumlah term, w_{ij} merupakan bobot term i pada dokumen latih j sedangkan w_{iq} merupakan bobot term i pada dokumen uji q.

$$CosSim(q, d_j) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t}(w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}} \quad (2)$$

2.4. Pembentukan Pengklasifikasi

Pembentukan pengklasifikasi dengan metode NWKNN diawali dengan pembentukan pengklasifikasi algoritma KNN. Mekanisme algoritma KNN penentukan klas dari suatu dokumen uji q dilakukan dengan mencari k tetangga terdekat dari dokumen latih berdasarkan peringkat kesamaannya. Selanjutnya akan dicari katagori dari dokumen uji dengan menghitung skore label tertinggi dari k dokumen yang dipilih menggunakan persamaan 3.

$$score(q, c_i) = \sum_{d_j \epsilon KNN(q)} Sim(q, d_j) \cdot \delta(d_j, c_i) \quad (3)$$

Dari persamaan diatas, $d_j \epsilon KNN(q)$ merupakan himpunan tetangga terdekat dari dokumen q, $Sim(q, d_j)$ merupakan kemiripan antara dokumen uji q dengan dokumen latih d_j sedangkan $\delta(d_j, c_i)$ merupakan klasifikasi dari dokumen d_j dengan mengacu pada

$$\delta(d_j, c_i) = \begin{cases} 1 & d_j \epsilon c_i \\ 0 & d_j \notin c_i \end{cases} \quad (4)$$

Dokumen uji q dimasukkan ke dalam kategori yang memiliki skor yang paling besar.

2.4.1. Algoritma NWKNN

Algoritma NWKNN mirip dengan algoritma KNN tradisional. Pembedanya terletak pada algoritma NWKNN dilakukan pemberian bobot. Pada tetangga – tetangga yang berasal dari kategori yang besar maka akan diberi bobot yang kecil sedangkan pada tetangga – tetangga yang berasal dari kategori yang kecil akan diberi bobot yang besar dengan menggunakan persamaan 5 [5]. Algoritma ini dapat dijelaskan sebagai berikut. Untuk setiap dokumen uji q, pilih K tetangga terdekat dari dokumen latih yang terdapat pada kategori $K^*\{C_1^d, C_2^d, \dots, C_{K^*}^d\}$. Kemudian bobot kategori dapat dihitung dengan persamaan berikut :

$$Weight_i = \frac{1}{\left(\frac{Num\left(C_i^d\right)}{Min\{Num\left(C_m^d\right) | \, m = 1, \dots, K^*\}}\right)^{1/exponent}} \quad (5)$$

2.5. Metode Evaluasi

Untuk mengevaluasi kesamaan diantara dokumen – dokumen dapat diukur menggunakan *recall*, *precision* dan *F1-measure*. *Recall* adalah tingkat keberhasilan mengenali suatu kelas yang harus dikenali. *Precision* adalah tingkat ketepatan hasil klasifikasi dari seluruh dokumen. *F1-measure* merupakan nilai yang mewakili keseluruhan kinerja sistem dan merupakan penggabungan nilai *recall* dan *precision* dalam sebuah nilai.

2.5.1. *Macroaveraged*

Macroaveraged adalah *precision* , *recall* dan *F1-measure* dievaluasi secara lokal pada tiap kategori dan kemudian dievaluasi secara global dengan cara menghitung rata-rata hasil *precision* , *recall* dan *F1-measure* dari kategori – kategori yang berbeda [3]. *Macroveraged F1* digunakan untuk mengevaluasi keseluruhan kinerja algoritma pada dataset yang diberikan. *Macroaveraged F1* menghitung nilai *F1* tiap kategori dan kemudian menghitung nilai rata – ratanya [2]. *Macroaveraged precision, recall* dan *F1* dapat didefinisikan sebagai berikut :

$$macroaveraged\ precision = \frac{\sum_{i=1}^{m} P(i)}{m} \quad (6)$$

$$macroaveraged\ recall = \frac{\sum_{i=1}^{m} R(i)}{m} \quad (7)$$

$$macroaveraged\ F1 = \frac{\sum_{i=1}^{m} F1(i)}{m} \quad (8)$$

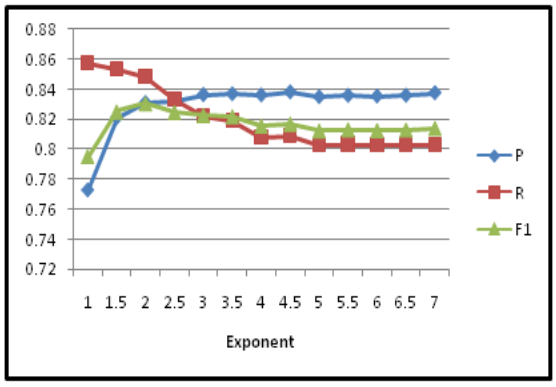
3. Hasil dan Pembahasan

3.1. Pengaruh Exponent Terhadap Kinerja NWKNN

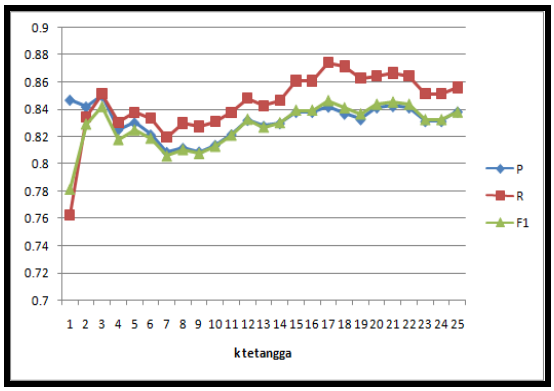
Hasil pengujian pengaruh nilai *exponent* terhadap kinerja NWKNN menunjukkan bahwa pada awalnya *F1 measure* mengalami kenaikan sampai *exponent* = 2 akan tetapi kemudian mengalami penurunan. Selain itu semakin besar nilai *exponent* nilai *macro precision* semakin naik dan nilai *macro recall* semakin turun. Hasil ini sesuai dengan apa yang dicapai oleh Tan pada penelitiannya. Grafik dari hasil pengujian ini dapat dilihat pada gambar 3.

3.2. Pengaruh Jumlah Tetangga (k) Terhadap Kinerja NWKNN

Hasil uji coba pengaruh k terhadap kinerja NWKNN dapat ditunjukkan pada gambar 4. Dari grafik menunjukkan bahwa pengaruh jumlah k tetangga terhadap hasil akurasi tidak stabil atau mengalami naik turun. Nilai *F1 measure* tertinggi diperoleh pada saat k = 17 yaitu sebesar 0.846 atau 84.6%. Sebelum k = 3 dapat dilihat bahwa kurva *precision* berbeda dengan kurva *recall* dan *F1 measure* tetapi mulai k=3 kurva *precision, recall* dan *F1 measure* memiliki kemiripan. Nilai k yang terlalu kecil menyebabkan akurasi klasifikasi tidak begitu baik karena terpengaruh adanya *noise*. Nilai k yang relatih lebih baik berada pada rentang k=15 sampai dengan k = 22.



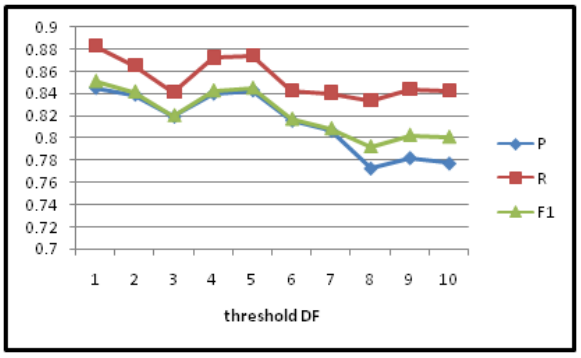
Gambar 3. Grafik Pengaruh Exponent Terhadap Kinerja NWKNN



Gambar 4. Grafik Pengaruh Jumlah Tetangga (k) Terhadap Kinerja NWKNN

3.3. Pengaruh Feature Selection Document Frequency Terhadap Kinerja NWKNN

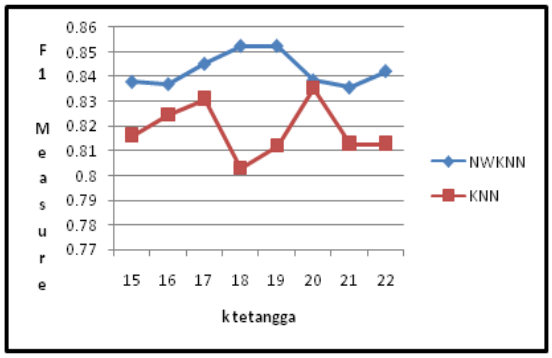
Dari gambar 5 dapat dilihat bahwa akurasi tertinggi dicapai pada saat *threshold* = 1. Nilai akurasi menurun setelah *threshold* lebih besar dari 1 hal ini dikarenakan *term* yang dibuang mencapai lebih dari 50% sehingga terdapat beberapa *term* yang seharusnya tidak perlu dibuang, ikut terbuang. Namun demikian proses *feature selection* dengan DF *thresholding* cukup mampu untuk meningkatkan kembali akurasi sistem sampai mendekati nilai akurasi pada saat *threshold* = 1. Hal ini dapat dilihat bahwa akurasi mengalami kenaikan kembali pada saat *threshold* = 4 dan *threshold* = 5. Dimana selisih nilai *precision, recall* dan *F1 measure* pada saat *threshold* = 1 dan *threshold* = 5 adalah 0.2%, 0.9% dan 0.6%. Apabila tidak mengikutsertakan *threshold* = 1, maka akurasi tertinggi diperoleh pada saat *threshold* = 5 yang mana nilai *precision*, *recall* dan *F1 measure* adalah *precision* = 84.3%, *recall* = 87.4% dan *F1 measure* = 84.5%.



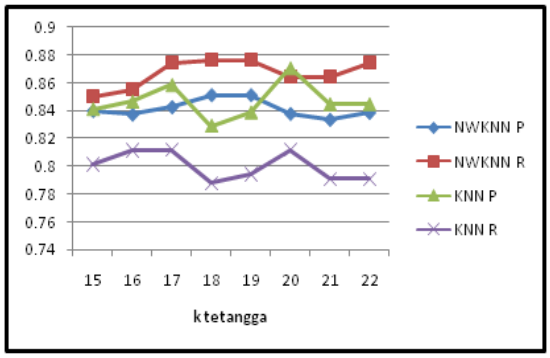
Gambar 5. Grafik Pengaruh Feature Selction DF Terhadap KinerjaNWKNN

3.4. Perbandingan Kinerja NWKNN dan KNN

Grafik pada gambar 6 dan 7 menunjukkan hasil pengujian pada parameter terbaik, yakni *threshold* = 5, *exponent* = 2, dan nilai k antara 15-22. Hasil uji coba menunjukkan bahwa NWKNN memiliki akurasi yang lebih baik daripada KNN dengan nilai rata – rata *precision*, *recall* dan *F1 measure* masing-masing 84.2%, 86.7% dan 84.3% dan 84.6%, 79.8% dan 81.7%. Dari gambar 6 dan 7 dapat dilihat bahwa algoritma KNN memiliki nilai *precision* yang sedikit lebih baik daripada NWKNN sedangkan algoritma NWKNN jauh lebih baik dalam *recall* sehingga *F1 measure* NWKNN lebih baik daripada KNN. Hal ini menunjukkan bahwa algoritma NWKNN memiliki kinerja yang lebih baik daripada algoritma KNN untuk mengklasifikasikan *corpus* teks berbahasa Indonesia yang tidak seimbang.



Gambar 6. Grafik Perbandingan F1 Measure NWKNN dan KNN



Gambar 7. Grafik Perbandingan *Precision* dan *Recall* NWKNN dan KNN

4. Simpulan

Kesimpulan yang dapat diperoleh dari penelitian ini adalah sebagai berikut :

1. Hasil uji coba klasifikasi dokumen bahasa Indonesia dengan NWKNN memberikan kinerja terbaiknya pada ekponen =2, *threshold* = 5 dan k antara 15 – 22.
2. Dari hasil pengujian, *feature selection* dengan metode *document frequency* cukup mampu meningkatkan kembali akurasi algoritma NWKNN degnan akurasi mengalami kenaikan pada saat *threshold* DF = 4 dan pada saat *threshold* DF = 5 nilai akurasinya mendekati nilai akurasi pada saat *threshold* DF = 1. Dimana selisih nilai *precision, recall* dan *F1 measure* pada saat *threshold* = 1 dan *threshold* = 5 adalah 0.2%, 0.9% dan 0.6%.
3. Hasil perbandingan kinerja antara WKNN dan KNN pada parameter terbaik menunjukkan *precision*, *recall* dan *F1* masing-masing sebesar 84.2%, 86.7% ,84.3% dan 84.6%, 79.8%, 81.7%. Nilai *recall* dan *F1 measure* NWKNN lebih tinggi 6.9% dan 2.6% daripada nilai *recall* dan *F1 measure* KNN. Hasil dari pengujian yang dilakukan menunjukkan bahwa algoritma NWKNN merupakan algoritma yang lebih baik untuk mengatasi permasalahan *corpus* teks yang tidak seimbang.

5. Saran

Saran yang diberikan untuk penelitian selanjutnya adalah sebagai berikut :

1. Dapat dilakukan perbandingan dengan metode pemilihan fitur yang lain seperti *Information Gain* (IG), *Chi Square* atau *Term Contribution* (TC).
2. Dapat diuji pengaruh *stemming* dari berbagai algoritma yang ada.

6. Daftar Pustaka

[1] Arifin, A.Z., Mahendra, I.P.A.K, Ciptaningtyas, H.T. 2008. *Enhanced Confix Stripping Stemmer And Ants Algorithm For Classifying News Document In Indonesian Language*. The 5th International Conference on Information & Communation Technology and Systems.

[2] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. 2004. *An kNN Model-based Approach and Its Application in Text Categorization*. European Comission Projects ICONS.

[3] Sebastiani, F. 2002. *Text Categorization*. In Alezandro Zanasi (ed.), Text Mining and Its Applications. WIT Press, Southampton, UK, 2005, pp. 109-129.

[4] Soucy, P., dan Mineau, G.W. 2005. *Beyong TFIDF Weighting For Text Categorization in the Vector Space Model*.

[5] Tan, S. 2005. *Neighbor-weighted K-nearest neighbor for unbalanced text corpus*. Expert System with Application (pp 667-671). ELSEVIER.

[6] Yang Y., Pederson J.O. 1997. *A Comparative Study on Feature Selection in Text Categorization*. Proc of ICML – 97 (pp 412-420).

[7] Zhang, X.F, Huang, H.Y, Zhang K.L. 2009. *KNN Text Categorization Algorithm Based on Semantic Centre*. 2009 International Conference on Information Technology and Computer Science