Pengkelasan Dokumen Web Menggunakan Teknik Support Vector Machine (SVM)

Mohd Shahizan Othman[a], Rozilawati Dollah @ Md. Zain[a], Lizawati Mi Yusuf [a] Juhana Salim[b] , Zarina Shukur[b] dan Chin Mae Yen[a]

[a]Fakulti Sains Komputer dan Sistem Maklumat, Universiti Teknologi Malaysia, 81310 Skudai, Johor
{shahizan, zeela, lizawati}@fsksm.utm.my

[b]Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor
{js, zs}@ftsm.ukm.my

ABSTRAK

Dewasa ini, kebanyakan enjin carian di internet menggunakan sistem pengindeksan subjek berbanding pengkelasan dokumen. Dalam sistem pengindeksan subjek, kosa kata atau kata kunci yang terkawal digunakan untuk menetapkan istilah pengindeksan pada dokumen-dokumen web. Manakala, pengkelasan dokumen pula akan mengkelaskan dokumen-dokumen web dalam satu struktur hirarki berdasarkan kategori subjek. Pengindeksan berdasarkan kata kunci berkemampuan untuk mencari dokumen-dokumen yang mengandungi kata kunci yang spesifik. Walau bagaimanapunia sukar untuk mengenalpasti dokumen-dokumen yang mempunyai kategori yang sama. Oleh yang demikian, pengkelasan teks secara automatik adalah diperlukan. Ini bertujuan untuk mengkelaskan dokumen-dokumen ke dalam kategori-ketegori yang berbeza berdasarkan kandungan teks. Sehubungan dengan itu, kertas kerja ini akan membincangkan tentang kajian pengkelasan teks dengan menggunakan kaedah Support Vector Machine (SVM). Set data yang digunakan dalam kajian ini diperolehi daripada Bank Search Information Consultancy Ltd. dan Jabatan Sains Komputer di University of Reading. Set data ini dipecahkan kepada empat kategori iaitu perbankan dan kewangan, bahasa pengaturcaraan, sains dan sukan. Hasil kajian ini menunjukkan peratus ketepatan pengkelasan dokumen web untuk set data yang digunakan adalah rendah dan kurang memuaskan.

Kata kunci
Pengkelasan, *Support Vector Machine* (SVM), dokumen web.

1 PENGENALAN

Pengkelasan web merupakan proses yang sistematik untuk menyusun atau mengorganisasikan dokumen web yang semakin bertambah dari masa ke semasa. Menurut McGovern dan Norton (2001), setiap hari terdapat tujuh juta dokumen baru yang diterbitkan menerusi web dan jumlahnya kini dianggarkan mencecah lebih 550 bilion. Perkembangan sumber maklumat web yang pesat ini telah menjadi satu cabaran kepada pengguna Internet untuk mencapai maklumat-maklumat terkini yang relevan dan berkualiti. Menurut Rachagan (2005), melayari laman web dengan kumpulan data yang terlalu banyak menyebabkan ramai individu terpaksa mengambil masa yang lama untuk mencari, mengumpul dan menyusun data. Ini

mengakibatkan pengurangan tahap produktiviti mereka.

Selain itu, fenomena perkembangan web yang pesat ini juga turut mendapat tumpuan komuniti yang membuat kajian tentang pengindeksan web. Menurut Davis et al (2003), isu-isu penting yang dititikberatkan dalam pengindeksan web, antaranya ialah mencari sumber maklumat yang tepat dan relevan, mempercayai kandungan maklumat, melayari sumber maklumat yang baru dan menggunakan kata kunci yang betul untuk mencapai maklumat yang dikehendaki.

Penyelidikan dalam pengkelasan dokumen merupakan gabungan bidang perlombongan data, pembelajaran mesin dan capaian maklumat. Kajian tentang pengkelasan teks yang pertama telah dilaksanakan oleh Maron (1961) membincangkan tentang pengindeksan automatik. Pada ketika itu, penyelidikan hanya tertumpu kepada pengkelasan dokumen teks yang terdapat di perpustakaan sahaja. Namun begitu, kepentingan pengkelasan teks telah berkembang dan menjadi penyelidikan utama dalam disiplin sistem maklumat pada awal tahun 90an. Hal ini disebabkan oleh perkembangan teknologi digital terutamanya internet dan web yang berkuantiti besar menyebabkan kesukaran mengkelaskan sumber web secara manual. Justeru itu, kajian ini dilaksanakan untuk mengenalpasti semua proses yang terlibat dalam pengkelasan teks, khususnya dokumen web. Pengujian keberkesanan terhadap proses pembelajaran dilakukan dengan mengukur peratus ketepatan dan peratus dapatan semula untuk dokumen web yang telah dikelaskan.

2 PENGKELASAN DOKUMEN WEB

Menurut Taylor (1999), pengkelasan teks bermaksud proses menyusun teks ke dalam kelas-kelas atau ketegori-kategori berdasarkan kepada perkongsian kualiti atau ciri-ciri tertentu. Sebastiani (2002) menyatakan pengkelasan teks merupakan aktiviti pembelajaran yang didefinisikan sebagai proses menentukan topik atau label kelas (pra-definisi) untuk dokumen-dokumen baru secara automatik berdasarkan persamaan dengan set latihan yang telah dilabel. Secara matematiknya, pengkelasan teks merupakan tugas menganggarkan fungsi output yang tidak diketahui iaitu $\Psi : D \times C \rightarrow \{T, F\}$ dengan fungsi $\Phi : D \times C \rightarrow \{T, F\}$ merupakan pengkelas (Sebastiani 2004). Fungsi Ψ dan Φ seharusnya mempunyai hasil yang hampir sama di mana,

$C = \{c_1, ..., c_m\}$ merupakan set tetap bagi kategori yang telah ditakrifkan

D = domain bagi dokumen

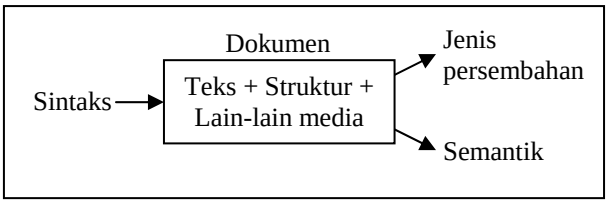
Secara umumnya, aktiviti pengkelasan teks telah bermula sejak awal tahun 60-an, tetapi hanya pada awal tahun 90-an aktiviti ini menjadi satu sub bidang utama dalam disiplin sistem maklumat. Ini disebabkan oleh peningkatan minat penyelidik dan adanya perkakasan yang lebih berkuasa. Pengkelasan teks kini diaplikasikan dalam pelbagai konteks, daripada pengindeksan dokumen berdasarkan kosa kata terkawal, penapisan dokumen, penjanaan metadata secara automatik sehingga melibatkan sebarang aplikasi yang memerlukan pengorganisasian dokumen (Sebastiani, 2002).

Pengkelasan dokumen web adalah berbeza dengan pengkelasan teks tradisional. Ini disebabkan oleh sifat dokumen web yang tidak berstruktur, mempunyai kadar hingar yang tinggi dan terdiri daripada gabungan pelbagai jenis tag dan teks. Pal et al. (2002) menyatakan kriteria dokumen web adalah tidak berlabel, teragih, heterogen, semistruktur, berubah mengikut masa dan berdimensi tinggi. Manakala Huang (2000) pula menyatakan, secara amnya kriteria dokumen web adalah:

i. Bersaiz sangat besar – keseluruhan dokumen web berjumlah 350 juta dokumen pada Julai 1998, dan saiz perkembangannya adalah 20 juta dokumen per bulan.

ii. Dokumen web yang bersifat dinamik – dokumen web berubah setiap hari berbanding dengan pangkalan data teks yang statik.
iii. Heterogen – internet mengandungi pelbagai jenis dokumen seperti imej, fail audio, teks dan skrip.
iv. Kepelbagaian bahasa – jumlah keseluruhan penggunaan bahasa di internet melebihi 100 bahasa.
v. Pautan yang tinggi – setiap dokumen web mempunyai sekurang-kurangnya 8 pautan kepada laman lain.

Dokumen web mempunyai ciri-ciri yang tersendiri di mana ianya bukan hanya terdiri daripada teks tetapi merupakan gabungan teks, struktur, audio, video, imej dan tag. Semua gabungan ini digunakan untuk menghasilkan satu laman web. Rajah 1 menunjukkan ciri-ciri dokumen web di mana sintaks dokumen akan menyatakan struktur, jenis persembahan, semantik ataupun tindakan luaran (Baeza-Yates dan Ribeiro-Neto 1999). Sintaks ini terdiri daripada arahan yang lengkap (gabungan tag). Terdapat sebanyak 109 tag HTML yang wujud. Walau bagaimanapun, bukan semuanya digunakan serentak dalam penghasilan sesebuah dokumen web.



Rajah 1: Ciri-ciri dokumen web (Baeza-Yates dan Ribeiro-Neto 1999)

Kajian ke atas struktur dokumen web oleh Etzioni (1996), Kosala dan Blockheel (2000) serta Mohd Shahizan et. al (2005) mendapati kebanyakan dokumen web lebih mementingkan jenis persembahan pada pelayar web berbanding dengan kandungannya. Ini terbukti daripada kajian yang dilaksanakan oleh Mohd Shahizan et. al (2005) yang mendapati peratusan penggunaan tag adalah lebih tinggi (48.75%) berbanding dengan kandungan dokumen webnya (26.12%). Jadual 1 menunjukkan peratusan penggunaan perkataan bagi dokumen web.

Jadual 1: Peratusan Penggunaan Perkataan Bagi Dokumen Web (Mohd Shahizan et.al, 2005)

Jenis Kandungan	Peratus
Tag	48.75
Kata Henti	21.52
Kandungan	26.12
Imbuhan	3.61

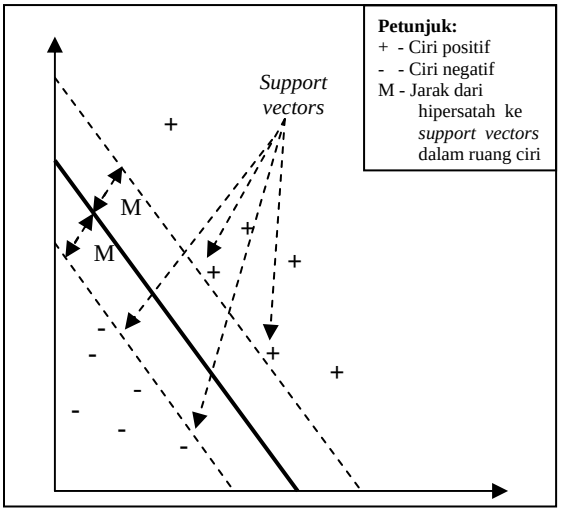
Terdapat banyak jenis algoritma pengkelasan yang telah dibangunkan, contohnya Bayes Naif, pepohon keputusan, peraturan keputusan, k-*Nearest Neighbour*, *Support Vector Machine* (SVM) dan rangkaian neural. Walau bagaimanapun, kertas kerja ini hanya tertumpu kepada kajian tentang pengkelasan teks menggunakan kaedah *Support Vector Machine* (SVM).

2.1 Kaedah *Support Vector Machine* (SVM)

Kaedah SVM telah diperkenalkan dalam bidang pengkelasan teks oleh Joachims dan digunakan oleh Dumais *et al.* (1998), Drucker *et al*. (1999), Yang dan Liu (1999) serta Klinkenberg dan Joachims (2000). SVM menyediakan dua sifat yang tidak terdapat pada algoritma pembelajaran yang lain iaitu proses memaksimumkan *margin* dan transformasi ruang input bukan-linear kepada ruang ciri menggunakan kaedah kernel (Cortes dan Vapnik, 1995). Algoritma SVM beroperasi dengan memetakan set latihan yang diberi kepada satu ruang ciri yang berdimensi tinggi dengan menempatkan satu pembahagi yang boleh mengasingkan model positif daripada model negatif dalam ruang tersebut. Dokumen-dokumen yang dilabelkan sebagai positif merupakan dokumen-dokumen yang berada dalam kategori c_i, manakala dokumen-dokumen yang dilabelkan sebagai negatif merupakan

dokumen-dokumen yang bukan berada di dalam kategori c_i.

Bentuk SVM yang paling mudah ialah SVM linear. SVM linear merupakan satu hipersatah iaitu sempadan kelas yang mengasingkan set data positif daripada set data negatif dengan *margin* yang maksimum di dalam ruang ciri. *Margin* (M) menandakan jarak dari hipersatah ke data positif dan negatif yang terdekat dalam ruang ciri (Yu et al., 2002). Rajah 2 menunjukkan contoh masalah dua-dimensi mudah yang boleh diasingkan secara linear.



Rajah 2: Perwakilan grafik satu SVM linear dalam kes dua dimensi (Yu et al, 2002)

Setiap ciri berpadanan dengan satu dimensi di dalam ruang ciri. Jarak dari hipersatah ke satu titik data adalah ditentukan oleh kekuatan setiap ciri dalam data tersebut. Sebagai contoh, satu pengkelas dokumen multimedia dipertimbangkan. Sekiranya satu dokumen mengandungi banyak ciri yang berkait dengan konsep "multimedia", contohnya perkataan-perkataan "multimedia", "grafik" atau "audio" pada bahagian kepala, dokumen ini akan digolongkan sebagai positif iaitu kelas multimedia di dalam ruang ciri. Lokasi titik datanya mesti jauh dari sempadan kelas di bahagian positif. Sebaliknya, satu dokumen lain yang mengandungi banyak ciri yang tidak berkait dengan multimedia sepatutnya ditempatkan jauh dari sempadan kelas di bahagian negatif (Yu *et al.*, 2002).

SVM mempunyai satu parameter *C* yang digunakan bagi kes-kes di mana titik-titik data tidak dapat diasingkan secara linear. Parameter ini merupakan penalti yang dikenakan ke atas data latihan yang jatuh ke bahagian yang salah dalam sempadan kelas. SVM akan menghitung hipersatah yang memaksimumkan jarak ke *support vectors* bagi satu nilai parameter yang diberi. Masalah-masalah yang tidak dapat diasingkan secara linear dapat diselesaikan dengan menggunakan kaedah kernel lanjutan yang berfungsi untuk mengubah satu ruang input bukan-linear kepada ruang ciri linear. Kaedah kernel lanjutan yang terdapat dalam SVM ialah polinomial, fungsi *radial basis* (RBF) dan *sigmoid tanh* (Lin *et al.*, 2003).

2.2 Justifikasi Pemilihan Kaedah Pengkelasan

Kaedah pengkelasan yang akan digunakan dalam perlaksanaan kajian ini ialah kaedah *Support Vector Machine* (SVM). Pemilihan kaedah ini didorong oleh faktor-faktor berikut:

i. Pemilihan perkataan-perkataan atau ciri-ciri yang spesifik untuk pembelajaran adalah tidak perlu kerana SVM merupakan algoritma yang tegap dengan perlindungan kepada masalah *overfitting* dan boleh menampung dimensi yang tinggi (Sebastiani, 2002).
ii. SVM boleh menampung jumlah ciri-ciri yang tinggi sehingga lebih daripada 10,000 dokumen untuk proses pembelajaran.
iii. Kebanyakan masalah dalam pengkelasan teks, khususnya kategori boleh diasingkan dan diselesaikan secara linear. Walau bagaimanapun, terdapat kategori-kategori seperti dalam set data *Reuters* yang tidak dapat diasingkan secara linear disebabkan oleh dokumen-dokumen yang kandungannya kurang bermakna. SVM boleh mencari pembahagi yang sesuai dengan menggunakan kaedah kernel

lanjutan sekiranya masalah linear ini wujud (Joachims, 1998).

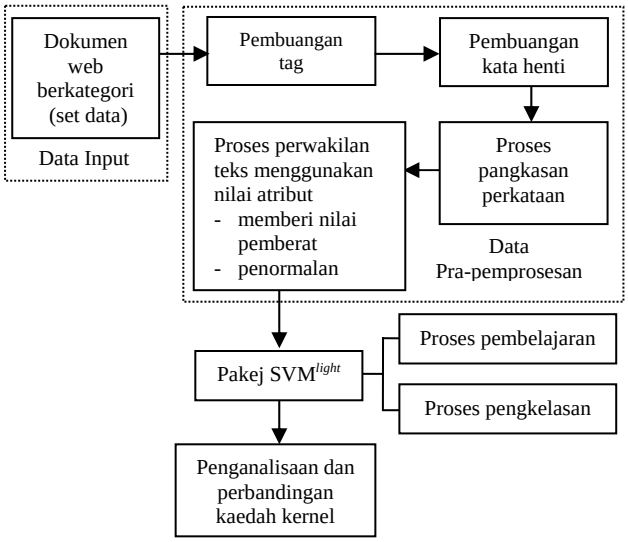
iv. SVM tidak memerlukan tenaga manusia atau tenaga mesin dalam menentukan nilai-nilai parameter yang sesuai kerana SVM telah menyediakan set parameter piawai yang telah terbukti efektif dalam pengkelasan teks (Sebastiani, 2002).

3 METOD

Kajian ini merupakan kajian kuantitatif dalam bidang pengkelasan teks yang melibatkan gabungan teknik capaian maklumat dan kaedah pembelajaran mesin, iaitu *Support Vector Machine* (SVM). Kajian yang dilaksanakan ini adalah bertujuan untuk menguji keberkesanan kaedah pengkelasan yang dipilih dengan menilai peratus ketepatan dan dapatan semula bagi proses pengkelasan yang dilakukan.

Set data yang digunakan dalam kajian ini diperolehi daripada *BankSearch Information Consultancy Ltd* dan Jabatan Sains Komputer di *University of Reading*, dan boleh dimuat turun melalui laman web http://www.pedal.rdg.ac.uk/banksearchdataset/. Set data yang digunakan dipecahkan kepada empat kategori iaitu perbankan dan kewangan, bahasa pengaturcaraan, sains dan sukan di mana setiap kategori terdiri daripada 2000 dokumen web.

Secara amnya, kajian ini dikategorikan kepada tiga bahagian, iaitu pra-pemprosesan data, pembelajaran serta pengkelasan dan pengujian. Rajah 3 menunjukkan rekabentuk operasi kajian yang telah dijalankan.



Rajah 3: Rekabentuk operasi kajian

3.1 Pra-Pemprosesan Data

Terdapat empat aktiviti yang terlibat di dalam perlaksanaan pra-pemprosesan iaitu:

i. Proses pembuangan tag

Proses pembuangan tag merupakan proses membuang semua tag HTML yang ada dalam dokumen web bagi mendapatkan kandungan teks dokumen web tersebut sahaja. Selepas proses pembuangan tag dilakukan, dokumen web diwakilkan sebagai dokumen yang tidak berstruktur (Embley *et al.*, 1998). Rajah 4 menunjukkan kod sumber dokumen web sebelum proses pembuangan tag, manakala Rajah 5 pula menunjukkan kandungan dokumen web selepas proses pembuangan tag.

Fungsi utama penggunaan tag-tag HTML ialah untuk paparan struktur dokumen web. Manakala, proses pengkelasan teks hanya menitik beratkan kandungan teks sebenar dalam dokumen web. Oleh itu, proses pembuangan tag mesti dilaksanakan untuk membuang semua tag HTML.

```
<Html> <head>
<META http-equiv=”Content-Type”
content=”text/html; charset=ISO-8859-1”>
<title>policy document</title>
<meta http-equiv=”Content-Type”
content=”text/html; charset=iso-8859-1”>
</head>
<body leftmargin=”0” topmargin=”0”
marginwidth=”0” marginheight=”0”>
<a name=”top”></a>
<p>This policy document contains full
details of the cover and conditions
<b>you</b> must satisfy and is the basis
on which all claims will be settled.<br>
<br>This policy constitutes a contract
between <b>you</b> and <b>us</b> and is
made up of the schedule and this policy
document, which together form the contract
of insurance, and is based upon the
information that <b>you</b> provided
during <b>your</b> application.<br>
<br>In return for the correct premium,
<b>we</b> will pay <b>you</b> or
<b>your</b> personal representative if
<b>you</b> make a valid claim.<br>
<br>Please be aware that this policy does
not cover for every eventuality.
<b>You</b> should read this policy
carefully to ensure that this meets with
<b>your</b> requirements. </p>….
```

Rajah 4: Kod sumber dokumen web sebelum proses pembuangan tag

```
policy document  apply back This policy
document contains full details of the
cover and conditions you must satisfy and
is the basis on which all claims will be
settled. This policy constitutes a
contract between you and us and is made up
of the schedule and this policy document,
which together form the contract of
insurance, and is based upon the
information that you provided during your
application. In return for the correct
premium, we will pay you or your personal
representative if you make a valid claim.
Please be aware that this policy does not
cover for every eventuality. You should
read this policy carefully to ensure that
this meets with your requirements.
```

Rajah 5: Kandungan dokumen web selepas proses pembuangan tag

ii. Proses pembuangan kata henti

Proses pembuangan kata henti merupakan proses membuang kata-kata henti seperti kata sandang (*a*, *an*, *the*), sendi nama (*in*, *of*, *at*), kata penghubung (*and*, *or*, *nor*) dan lain-lain daripada dokumen-dokumen yang telah melalui proses pembuangan tag (Guo *et al.*, 2002). Proses ini dilakukan dengan bantuan daripada satu senarai kata henti. Perbandingan dibuat antara dokumen dan senarai kata henti tersebut bagi membolehkan proses pembuangan kata henti dilaksanakan. Matlamat utama proses penghapusan kata henti ialah untuk menyingkirkan hingar daripada dokumen-dokumen set data kajian (Mittermayer *et al.*, 2001). Rajah 6 menunjukkan kandungan dokumen selepas semua kata henti dibuang.

```
policy document apply back policy
document full details cover conditions
satisfy basis claims settled policy
constitutes contract made schedule policy
document form contract insurance based
information provided application return
correct premium pay personal
representative make valid claim aware
policy cover eventuality read policy
carefully ensure meets requirements
```

Rajah 6: Kod sumber dokumen web sebelum proses pembuangan tag

iii. Proses pangkasan perkataan

Proses pangkasan perkataan pula merupakan proses membuang imbuhan akhiran pada setiap perkataan dalam dokumen untuk menjadikan perkataan tersebut sebagai kata dasar. Contoh kata dasar ialah perkataan *connect* yang telah dipangkas daripada perkataan-perkataan *connected*, *connecting*, *connection* dan *connections* (Baeza-Yates dan Ribeiro-Neto 1999). Proses pangkasan perkataan boleh memperluaskan liputan perkataan (*feature coverage*) dan seterusnya dapat meningkatkan ketepatan proses pengkelasan (Liao *et al.*, 2001).

Proses pangkasan perkataan dalam kajian ini dilakukan dengan menggunakan algoritma *Porter*. Algoritma ini ditulis oleh Martin Porter dan telah digunakan secara meluas sejak 20 tahun yang lalu. Algoritma *Porter* menggunakan satu senarai imbuhan akhiran untuk proses pangkasan. Algoritma ini mengaplikasikan satu siri peraturan kepada imbuhan akhiran perkataan dalam teks (Baeza-Yates dan Ribeiro-Neto 1999). Dua contoh aplikasi algoritma *Porter* adalah seperti berikut:

$$sses \rightarrow ss \quad (1)$$

$$s \rightarrow \emptyset \quad (2)$$

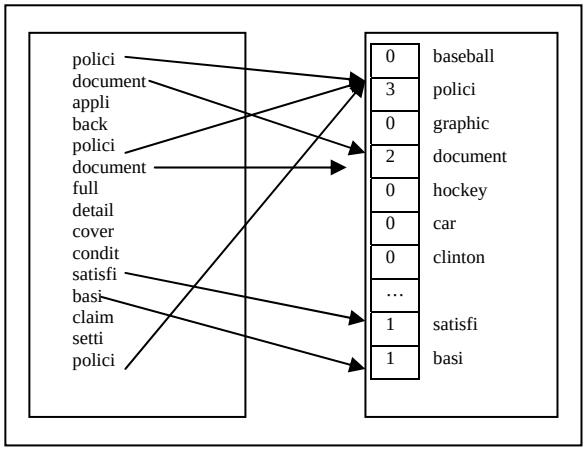
Berdasarkan peraturan (1), perkataan dengan aksara-aksara akhiran *sses* akan dipangkas imbuhan akhirannya iaitu *es*. Contohnya, perkataan *stresses* akan menjadi *stress* selepas melalui proses pangkasan. Berdasarkan peraturan (2) pula, perkataan majmuk yang mempunyai imbuhan akhiran *s* akan dipangkas menjadi kata dasar dengan membuang aksara *s* tersebut (Baeza-Yates dan Ribeiro-Neto 1999). Rajah 7 menunjukkan kandungan dokumen selepas semua perkataan dipangkas.

```
polici      cover       constitut
document    condit      contract
appli       satisfi     made
back        basi        schedul
polici      claim       form
document    settl       contract
full        polici      insure
detail
```

Rajah 7: Kandungan dokumen web selepas proses pangkasan perkataan

iv. Proses perwakilan teks

Proses perwakilan teks merupakan proses yang terakhir dalam perlaksanaan pra-pemprosesan data. Proses ini diperlukan untuk menukar setiap perkataan t_k dalam set data latihan kepada format yang difahami oleh SVM untuk dijadikan sebagai input kepada proses pembelajaran SVM. Proses perwakilan teks ini menggunakan perwakilan nilai atribut di mana setiap perkataan t_k bersamaan dengan satu vektor ciri. Rajah 8 menunjukkan contoh perwakilan teks sebagai vektor ciri.



Rajah 8: Perwakilan teks sebagai vektor ciri (Joachims, 1998)

Merujuk kepada Rajah 8, bahagian sebelah kiri menunjukkan kandungan teks dokumen dan bahagian sebelah kanan menunjukkan vektor-vektor ciri. Penomboran di sebelah kiri setiap vektor ciri merupakan bilangan vektor ciri tersebut muncul dalam kandungan teks dokumen. Contohnya, perkataan *baseball* tidak muncul langsung dalam kandungan teks dokumen yang dikaji, perkataan *polici* pula muncul sebanyak tiga kali dan perkataan *satisfi* muncul sekali sahaja.

Jumlah vektor ciri yang terlalu banyak akan melambatkan proses pembelajaran SVM. Oleh itu, hanya perkataan-perkataan yang muncul sekurang-kurangnya tiga kali di dalam keseluruhan set data latihan akan diambil kira sebagai vektor ciri dan diberi nombor ciri (Joachims, 1998).

Selepas vektor-vektor ciri yang dipilih diberi nombor ciri yang berpadanan, setiap vektor ciri t_k yang muncul dalam dokumen d_j akan diberi nilai pemberat dengan menggunakan kaedah capaian maklumat piawai *tfidf* (Sebastiani, 2002). Kaedah *tfidf* ini merupakan gabungan kaedah pengiraan *term frequency* (*tf*) dan *inverse document frequency* (*idf*). Pengiraan nilai pemberat untuk setiap vektor ciri t_k yang muncul dalam dokumen d_j adalah seperti formula berikut:

$$tfidf(t_k, d_j) = \#(t_k, d_j).\log \frac{|Tr|}{\#Tr(t_k)} \qquad (3)$$

di mana:

$\#(t_k,d_j)$ = bilangan kali vektor ciri t_k muncul dalam dokumen d_j

$\#Tr(t_k)$= bilangan dokumen kemunculan vektor ciri t_k

$|Tr|$ = jumlah keseluruhan dokumen latihan

Selepas setiap vektor ciri t_k yang muncul dalam dokumen d_j diberi nilai pemberat, setiap nilai pemberat ini perlu diwakilkan di antara julat [0, 1] untuk memperolehi vektor yang sama panjang. Oleh itu, pemberat-pemberat yang dikira daripada formula (3) perlu dinormalkan dengan menggunakan kaedah penormalan kosinus. Pengiraan penormalan pemberat-pemberat dilakukan dengan menggunakan formula berikut:

$$W_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{j=1}^{|Tr|}(tfidf(t_k, d_j))^2}} \qquad (4)$$

di mana:

$tfidf(t_k , d_j)$ = pemberat vektor ciri t_k dalam dokumen d_j

Selepas semua vektor ciri dalam semua dokumen latihan melalui fasa pemberian pemberat dan penormalan, fasa terakhir iaitu pelabelan positif dan negatif perlu dilakukan secara manual untuk setiap dokumen sebagai persediaan untuk melalui proses pembelajaran. Label positif menandakan dokumen dalam kategori c_i dan label negatif menandakan dokumen bukan dalam kategori c_i. Dokumen-dokumen yang telah dilabelkan disusun dalam satu fail latihan dan fail ini akan diinputkan kepada SVM untuk proses pembelajaran.

```
<line> .=. <target>
<feature>:<value> <feature>:<value>
...
<target> .=. +1 | -1
<feature> .=. <integer>
<value> .=. <float>
```

Rajah 9: Format fail SVM (Joachims, 2004)

Rajah 9 merupakan format fail yang difahami oleh SVM, manakala Rajah 10 pula menunjukkan keratan fail latihan untuk kategori perbankan dan kewangan. Berdasarkan Rajah 9, `<target>` merupakan label dokumen sama ada positif atau negatif. `<feature>` merupakan nombor ciri untuk satu-satu vektor ciri yang spesifik dalam bentuk integer dan `<value>` merupakan nilai yang telah dinormalkan untuk vektor ciri tersebut dalam bentuk nilai perpuluhan.

```
1 6:0.0198403253586671
15:0.0339873732306071
29:0.0360280968798065
31:0.0378103484117687
41:0.0456787263779904 63:0.021442413608662
74:0.0813238108919922
75:0.0201048944012214
81:0.0603996615380116
-1 6:0.0357790757594048
12:0.241194487897018 34:0.285141343931829
84:0.0699952392788633 85:0.073088116454553
97:0.297674662266474 118:0.312381698963476
216:0.103763566134826
218:0.270603391933215
```

Rajah 10: Keratan fail latihan untuk kategori perbankan dan kewangan

Berdasarkan Rajah 9, terdapat dua dokumen latihan di mana satu dilabelkan dengan positif dan satu lagi dilabelkan dengan negatif. Vektor ciri bernombor enam muncul dalam kedua-dua dokumen tetapi dengan nilai yang berbeza. Vektor-vektor ciri yang bernilai 0 diabaikan dan susunan vektor-vektor ini adalah secara berjujukan.

3.2 Proses Pembelajaran SVM

Set data yang digunakan akan dipecahkan kepada empat kategori iaitu perbankan dan kewangan, bahasa pengaturcaraan, sains dan sukan di mana setiap kategori terdiri daripada 2000 dokumen web. Dokumen ini

dibahagikan kepada dua set iaitu 70% sebagai set data latihan untuk proses pembelajaran dan 30% lagi sebagai set data pengujian untuk proses pengujian pengkelasan.

Perlaksanaan proses pembelajaran set data latihan dan pengkelasan set data pengujian dilakukan dengan menggunakan pakej SVM^{light} oleh Joachims (1998). Set data yang telah dinormalkan diinputkan kepada pakej SVM^{light} untuk proses pembelajaran.

3.3 Proses Pengkelasan SVM

Dokumen-dokumen pengujian mempunyai format yang sama dengan dokumen-dokumen latihan. Dokumen-dokumen pengujian yang telah melalui semua aktiviti dalam pra-pemprosesan data disusun dalam satu fail pengujian mengikut label dan diinputkan ke dalam SVM^{light} untuk proses pengkelasan. Selepas proses pengkelasan selesai, SVM^{light} akan menjana satu fail *predictions* yang menyimpan nilai-nilai jangkaan keputusan untuk dokumen-dokumen pengujian yang telah dikelaskan. Label positif atau negatif pada nilai-nilai ini menentukan kategori satu-satu dokumen pengujian.

Sekiranya nilai adalah positif, maka dokumen pengujian tersebut dijangka tergolong dalam kategori c_i. Sebaliknya, sekiranya nilai adalah negatif, maka dokumen pengujian tersebut dijangka bukan tergolong dalam kategori c_i. Set data pengujian dikelaskan untuk menguji keberkesanan dan ketepatan proses pembelajaran.

3.4 Pengukuran

Penganalisaan dibuat ke atas hasil pengkelasan set data pengujian dengan menggunakan kaedah pengukuran pencapaian statistik iaitu nilai ketepatan dan nilai dapatan semula. Kedua-dua pengukuran pencapaian ini digunakan secara meluas dalam bidang capaian maklumat dan diaplikasikan dalam masalah-masalah pengkelasan. Selain itu, pengukuran F_1 dan purata makro F_1 yang melibatkan nilai ketepatan dan nilai dapatan semula juga digunakan dalam penganalisaan kajian. Selepas proses penganalisaan, perbandingan antara kaedah kernel linear, polinomial dan fungsi *radial basis* dibuat untuk mendapatkan kaedah kernel yang terbaik bagi set data yang digunakan.

3.4.1 Nilai Ketepatan dan Nilai Dapatan Semula

Nilai ketepatan didefinisikan sebagai kebarangkalian satu dokumen yang dijangkakan tergolong dalam kategori c_i benar-benar tergolong dalam kategori c_i. Nilai dapatan semula pula didefinisikan sebagai kebarangkalian satu dokumen yang tergolong dalam kategori c_i dikelaskan ke dalam kategori c_i (Joachims, 1998). Kedua-dua pengukuran capaian ini digunakan dalam setiap pengujian pengkelasan yang dilakukan. Pengiraan nilai ketepatan diberi dalam formula berikut (Yang, 1998):

$$\text{Ketepatan} = \frac{\text{jumlah dokumen relevan yang dicapai}}{\text{jumlah dokumen dicapai}} \quad (5)$$

Berdasarkan persamaan (5) di atas, sekiranya terdapat sepuluh dokumen yang dicapai dan hanya dua daripadanya adalah relevan, maka nilai ketepatan ialah 0.2 atau 20%. Pengiraan nilai dapatan semula diberi dalam formula berikut (Yang, 1998):

$$\text{Dapatan semula} = \frac{\text{jumlah dokumen relevan yang dicapai}}{\text{jumlah dokumen relevan}} \quad (6)$$

Berdasarkan persamaan (6), sekiranya terdapat empat dokumen relevan yang dicapai dan sepuluh dokumen dalam koleksi adalah relevan, maka nilai dapatan semula ialah 0.4 atau 40%.

3.4.2 Pengukuran F_1 Dan Purata Makro F_1

Pengukuran F_1 merupakan pengukuran umum yang digunakan untuk

menggabungkan nilai ketepatan dan nilai dapatan semula menjadi satu pengukuran. Pengiraan F_1 diberi dalam formula berikut (Guo *et al.*, 2002):

$$F_1 \text{ (ketepatan, dapatan semula)} = \frac{2*\text{ketepatan}*\text{dapatan semula}}{\text{ketepatan + dapatan semu}} \quad (7)$$

Selain pengukuran F_1, purata makro F_1 juga digunakan untuk menilai pencapaian keseluruhan proses pembelajaran terhadap set data. Purata makro F_1 menghitung nilai F_1 untuk setiap kategori dan kemudian, mengambil purata semua nilai F_1 yang dihitung. Pengiraan purata makro F_1 adalah menggunakan formula berikut:

$$\text{Purata makro } F_1 = \frac{\sum_{i=1}^{m} F_1(i)}{m} \quad (8)$$

di mana:
$F_1(i)$ = nilai F_1 untuk kategori ke-i
m = jumlah bilangan kategori

Selepas proses penganalisaan dibuat untuk setiap kategori, pengukuran purata makro F_1 digunakan untuk membuat perbandingan antara kaedah kernel linear, polinomial dan fungsi *radial basis*. Nilai purata makro F_1 yang tertinggi menandakan bahawa kaedah kernel yang bersepadan merupakan kaedah kernel yang terbaik untuk set data yang digunakan.

4 ANALISIS HASIL PENGKELASAN

Analisis hasil pengkelasan merupakan analisa untuk menguji keberkesanan proses pembelajaran dan ketepatan proses pengkelasan. Sekiranya ketepatan proses pengkelasan adalah tinggi, maka proses pembelajaran yang dilaksanakan adalah berkesan. Hasil pengkelasan diukur dengan nilai ketepatan, nilai dapatan semula dan pengukuran F_1. Tiga kaedah kernel yang berbeza digunakan untuk proses pembelajaran iaitu kernel linear, polinomial dan fungsi *radial basis*.

4.1 Analisis Pengkelasan Kategori Perbankan dan Kewangan

Jadual 2 menunjukkan peratus ketepatan pengkelasan, nilai ketepatan, nilai dapatan semula dan nilai pengukuran F_1 untuk hasil pengkelasan set data pengujian kategori perbankan dan kewangan.

Jadual 2: Hasil pengkelasan kategori perbankan dan kewangan

Pengukuran / Kernel	%Ketepatan (*Accuracy*)	Ketepatan (*Precision*)	Dapatan semula	F_1
Linear	56.67	0.5357	1.0000	0.6977
Polinomial	56.67	0.5357	1.0000	0.6977
Fungsi *radial basis*	53.33	1.0000	0.0667	0.1251

Berdasarkan Jadual 2, ketiga-tiga kaedah kernel menghasilkan peratus ketepatan pengkelasan yang sederhana dengan purata 55.56%. Kaedah kernel linear dan polinomial mempunyai tahap ketepatan yang sama. Pencapaian kernel fungsi *radial basis* adalah kurang baik dengan nilai F_1 yang paling rendah iaitu 0.1251 sahaja berbanding dengan dua kernel lain yang mempunyai nilai F_1 yang agak tinggi iaitu 0.6977.

4.2 Analisis Pengkelasan Kategori Bahasa Pengaturcaraan

Jadual 3 menunjukkan peratus ketepatan pengkelasan, nilai ketepatan, nilai dapatan semula dan nilai pengukuran F_1 untuk hasil pengkelasan set data pengujian kategori bahasa pengaturcaraan.

Jadual 3: Hasil pengkelasan kategori bahasa pengaturcaraan

Pengukuran / Kernel	%Ketepatan (*Accuracy*)	Ketepatan (*Precision*)	Dapatan semula	F_1
Linear	43.33	0.4643	0.8667	0.6047
Polinomial	43.33	0.4643	0.8667	0.6047
Fungsi *radial basis*	50.00	0.5000	1.0000	0.6667

Berdasarkan Jadual 2, ketiga-tiga kaedah kernel menghasilkan peratus ketepatan pengkelasan yang agak rendah dengan purata 45.55%. Kaedah kernel linear dan polinomial mempunyai tahap ketepatan yang sama, seperti kes untuk kategori perbankan dan kewangan. Pencapaian kernel fungsi *radial basis* adalah lebih baik daripada dua kernel yang lain dengan nilai F_1 yang paling tinggi iaitu 0.6667, sepadan dengan peratus ketepatannya yang juga paling tinggi untuk kategori bahasa pengaturcaraan.

4.3 Analisis Pengkelasan Kategori Sains

Jadual 4 menunjukkan peratus ketepatan pengkelasan, nilai ketepatan, nilai dapatan semula dan nilai pengukuran F_1 untuk hasil pengkelasan set data pengujian kategori sains.

Jadual 4: Hasil pengkelasan kategori Sains

Pengukuran / Kernel	%Ketepatan (*Accuracy*)	Ketepatan (*Precision*)	Dapatan semula	F_1
Linear	50.00	0.5000	1.0000	0.6667
Polinomial	50.00	0.5000	1.0000	0.6667
Fungsi *radial basis*	50.00	0.5000	1.0000	0.6667

Berdasarkan Jadual 4, ketiga-ketiga kaedah kernel menghasilkan peratus ketepatan pengkelasan, nilai ketepatan, nilai dapatan semula dan nilai pengukuran F_1 yang sama. Peratus ketepatan pengkelasan yang dihasilkan adalah kurang memuaskan kerana hanya setengah daripada set data pengujian dikelaskan dengan betul.

4.4 Analisis Pengkelasan Kategori Sukan

Jadual 5 menunjukkan peratus ketepatan pengkelasan, nilai ketepatan, nilai dapatan semula dan nilai pengukuran F_1 untuk hasil pengkelasan set data pengujian kategori sukan.

Jadual 5: Hasil pengkelasan kategori Sukan

Pengukuran / Kernel	%Ketepatan (*Accuracy*)	Ketepatan (*Precision*)	Dapatan semula	F_1
Linear	76.67	0.6818	1.0000	0.8108
Polinomial	76.67	0.6818	1.0000	0.8108
Fungsi *radial basis*	53.33	0.5185	0.9333	0.6666

Berdasarkan Jadual 5, kaedah kernel linear dan polinomial menghasilkan peratus ketepatan pengkelasan yang sama dan memuaskan dengan 76.67%, sementara kernel fungsi radial basis menghasilkan peratus ketepatan yang agak rendah iaitu 53.33% sahaja. Selain itu, pengukuran F_1 untuk kaedah kernel linear dan polinomial juga agak tinggi dengan nilai 0.8108 melebihi semua kategori yang lain.

4.5 Perbandingan Antara Kaedah Kernel yang diGunakan

Proses pembelajaran set data latihan dijalankan sebanyak tiga kali oleh setiap kategori dengan menggunakan kaedah kernel yang berbeza iaitu kaedah kernel linear, polinomial dan fungsi *radial basis*. Ini membolehkan perbandingan dibuat antara kaedah kernel untuk menentukan kaedah kernel yang paling sesuai untuk pengkelasan teks, khususnya set data yang digunakan dalam kajian ini.

Berdasarkan analisis yang telah dilakukan, didapati bahawa kaedah kernel piawai dalam pakej SVM^{light} iaitu kernel polinomial adalah sesuai untuk diaplikasikan dalam pengkelasan teks dan khususnya, untuk set data latihan yang digunakan dalam proses pembelajaran. Selain itu, kaedah kernel linear yang mempunyai nilai purata makro F_1 yang sama dengan kernel polinomial juga boleh digunakan untuk set data latihan yang dikaji. Kaedah kernel fungsi *radial basis* mempunyai pencapaian yang sangat rendah dalam perbandingan ini dengan nilai purata makro F_1 setengah daripada kernel linear dan polinomial. Ini menandakan bahawa kaedah kernel fungsi *radial basis* tidak sesuai diaplikasikan untuk set data kajian.

5 KESIMPULAN

Hasil daripada kajian yang telah dijalankan, penganalisaan menunjukkan bahawa peratus ketepatan pengkelasan untuk set data yang digunakan adalah rendah dan kurang memuaskan. Terdapat tiga sebab yang mungkin menyebabkan peratus ketepatan pengkelasan yang rendah, iaitu:

i. Penggunaan jumlah set data latihan yang kecil.
ii. Kandungan dokumen-dokumen dalam set data adalah kurang bermakna.
iii. Penggunaan kaedah penormalan kosinus yang kurang sesuai dalam proses perwakilan teks.

Secara keseluruhannya, kategori sukan mempunyai pencapaian yang paling tinggi untuk kaedah kernel linear dan polinomial di mana peratus ketepatan pengkelasan adalah 76.67%. Ini menunjukkan bahawa kata-kata kunci yang digunakan dalam kategori sukan tidak luas dan kebanyakan dokumen dalam kategori ini menggunakan perkataan-perkataan yang sama.

Selain itu, perbandingan kaedah kernel yang dibuat selepas hasil penganalisaan kategori mendapati bahawa kaedah kernel linear dan polinomial merupakan kernel yang terbaik untuk set data yang digunakan dalam kajian ini. Keputusan ini menyokong kajian Joachims yang menyatakan bahawa kernel piawai iaitu kernel polinomial dalam pakej SVM^{light} adalah sesuai untuk diaplikasikan dalam pengkelasan teks. Walau bagaimanapun, kaedah kernel linear juga boleh dijadikan sebagai pilihan kerana kernel ini mempunyai nilai purata makro F_1 yang sama dengan kaedah kernel polinomial.

6 RUJUKAN

Baeza-Yates, R. dan Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Wokingham, Essex: Addison Wesley

Cortes, C. dan Vapnik, V. (1995). Support-Vector Network. *Machine Learning*. **20**(3), 273–297

Davis, C., Miller, K. dan O'Shea, A. (2003). Introduction of Classification of the Web. (atas talian). http://www.slais.ubc.ca/ courses/libr517/ projects/classification/ Intro.html (10 Disember 2003)

Drucker HD, Wu D dan Vapnik V (1999). Support vector machines for spam categorization. *IEEE Transactions On Neural Networks*. **10**(5):1048–1054

Dumais, S. T., Platt, J., Heckerman, D. dan Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. (PDF file) *In Proceedings of ACM-CIKM98*, Nov. 1998, pp. 148-155

Embley, D. W., Campbell, D. M., Jiang, Y. S., Ng, Y. K. dan Smith, R. D. (1998). *A Conceptual-Modeling Approach to Extracting Data for the Web*. Proceedings of the 17th International Conference on Conceptual Modeling (ER'98), hlm 78-91

Etzioni, O. (1996). The World Wide Web: quagmire or gold mine? *Communications of the ACM*, **39**(11):65-68

Guo G, Wang H, Bell D, Bi Y dan Greer Y (2004). Using kNN Model for Automatic Text Categorization. *Journal of Soft Computing*, Springer-Verlag Heidelberg.

Huang, L. (2000). A survey on web information retrieval technologies. Laporan Teknik ECSL http://www.ecsl.cs.sunysb.edu/tr/rpe8.ps. Z.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of 10th European Conference on Machine Learning*. Chemnitz, Germany: 137-142.

Joachims, T. (2004). SVM^{light} Support Vector Machine. (atas talian) pada

http://www.cs.
cornell.edu/People/tj/svm_light/ (3 Januari 2005)

Klinkenberg, R. dan Thorsten, J. (2000). Detecting Concept Drift with Support Vector Machines. *In Proceedings of the Seventeenth International Conference on Machine Learning*, hlm 487-494

Kosala, R. dan Blockeel, H. (2000). Web Mining Research: A Survey. ACM SIGKDD. **2**(1):1-15

Liao, C., Alpha, S. dan Dixon, P. (2001). *Feature Preparation in Text Categorization*. Oracle Corporation.

Lin, C. J., Hsu, C. W. dan Chang, C. C. (2003). *A Practical Guide to Support Vector Classification*. National Taiwan University, Taipei 106, Taiwan.

Maron, M (1961). Automatic indexing: An experimental inquiry. *Journal of the Association for Computing Machinery*. 8(3): 404-417.

McGovern, G. dan Norton, R. (2001). Content *Critical – Gaining Competitive Advantage Through High-Quality Web Content*. Financial Times Prentice Hall. London.

Mittermayer, M. A., Brucher, H. dan Knolmayer, G. (2001). Document Classification Methods for Organizing Explicit Knowledge. *Proceedings of the Third European Conference on Organizational Knowledge, Learning and Capabilities*, Athens.

Mohd Shahizan Othman , Lizawati Mi Yusuf, Juhana Salim dan Zarina Shukur. (2005). Kajian Terhadap Penggunaan Tag, Meta Tag Dan Perkataan Bagi Sumber Maklumat Web . *Simposium Kebangsaan Sains Matematik ke 13*. Hotel Holiday Villa, Alor Star. 31 Mei - 2 Jun. Jil. 2:975-984

Pal, S.K., Talwar, V. dan Mitra, P. (2002). Web Mining in Soft Computing Framework: Relevance, State of the Art and Future Direction. IEEE Transactions on Neural Networks. **13**(5):1163-1177

Rachagan, S. (2005). Rakyat tidak boleh harapkan usaha kerajaan tapis internet. *Berita Harian*, 11 Ogos:10

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, Vol. 34, No. 1: 1-47

Sebastiani, F. (2004). Text Classification for Web Filtering. POESIA Workshop. 21-22 Jan.

Taylor, A. G. (1999). *The Organization of Information* Englewood:Libraries Unlimited

Yang, Y. (1998). *An Evaluation of Statistical Approaches to Text Categorization*. The Netherlands: Kluwer Academic Publishers. 69-90

Yang, Y. dan Liu, X. (1999). A Re-examination of Text Categorization Methods. *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*: 42-49

Yu, H., Han, J. dan Chang, K. C. C. (2002). PEBL: Positive Example Based Learning for Web Page Classification Using SVM. *Proceedings of SIGKDD'02*. Edmonton, Alberta, Canada.