LaporanPenelitian

Sentiment Classification menggunakan Machine Learning: Metode Naïve-Bayes dan Support Vector Machines (Studi kasus: movie reviews imdb.com)

Tjatur Kandaga, S.Si., M.T.
Hendra Bunyamin, S.Si., M.T.
Diana Trivena Yulianti, S. Kom., M.T.

Desember2012
Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Maranatha

LEMBAR IDENTITAS

1. JudulPenelitian: *Sentiment Classification*menggunakan*Machine Learning*: MetodeNaïve Bayes dan Support Vector Machines (Studikasus: *moviereviews***imdb.com**).
2. Ketua/PenanggungJawabPelaksanaKegiatanPenelitian:
   Nama (lengkap dengan gelar) : TjaturKandaga, S.Si., M.T.
   NIK : 720080
   JabatanAkademik / Golongan : Lektor / III C
   Fakultas / Jurusan : Universitas Kristen Maranatha
3. Jumlah Tim Peneliti : 3 orang
4. LokasiPelaksanaPenelitian : FakultasTeknologiInformasi Universitas Kristen Maranatha
5. Lama Pelaksanaan : 6 minggu
6. Sumber Dana Penelitian : Universitas Kristen Maranatha
7. BiayaPenelitian :Rp. 8.000.000,-

Bandung, 20Desember2012

Ketua / PenanggungJawabPelaksana

TjaturKandaga, S.Si., M.T.

Menyetujui,

DekanFakultasTeknologiInformasi

Dr. Ir. MewatiAyub, M.T.

Mengetahui,

LPPM

Prof. Dr. Ir. Benjamin Soenarko, MSME

LEMBAR PENGESAHAN

JudulPenelitian	:	*Sentiment Classification*menggunakan*Machine Learning*: Metode Naïve Bayes dan Support Vector Machines (Studikasus: *movie reviews***imdb.com**)
Peneliti	:	1. TjaturKandaga, S.Si., M.T. 2. HendraBunyamin, S.Si., M.T. 3. Diana TrivenaYulianti, S.Kom., M.T.
LokasiPelaksanaPenelitian	:	FakultasTeknologiInformasi Universitas Kristen Maranatha Jl. Surya Sumantri no. 65 Bandung

Penelitianinitelahdiselesaikanpadatanggal 18Desember2012
sebagaisalahsatuperwujudanTridharmaPerguruanTinggiUniversitas Kristen Maranatha

Bandung, 20Desember2012
KetuaPeneliti

TjaturKandaga, S.Si., M.T.

DekanFakultasTeknologiInformasi

Dr. Ir. MewatiAyub, M.T.

KetuaLPPM

Prof. Dr. Ir. Benjamin Soenarko, MSME

ABSTRAK

Automatic text categorization adalah teknik untuk mengklasifikasikan dokumen-dokumen sesuai dengan label atau kelasnya secara otomatis. Hal ini merupakan masalah yang penting untuk diselesaikan terutama bila jumlah dokumen yang hendak diklasifikasi banyak (lebih dari 1000 dokumen) karena pengklasifikasikan secara manual untuk jumlah dokumen yang luar biasa banyaknya sudah tidak *feasible* untuk dilakukan secara manual.

Penelitian ini bertujuan untuk membandingkan akurasi dari dua algoritma *machine learning*, yaitu Naïve Bayes dan Support Vector Machines (SVM) yang akan digunakan untuk mengklasifikasikan *movie review* positif dan negatif dari *dataset movie reviews* imdb.com. Dalam penelitian ini, algoritma Naïve Bayes diterapkan dengan menggunakan *feature unigram*, sedangkan algoritma SVM diaplikasikan dengan menggunakan model C -Support *vector classification*dari *library* LIBSVM.

Hasil penelitian membuktikan bahwa akurasi algoritma Naïve Bayes lebih baik daripada akurasi algoritma SVM dalam mengklasifikasikan *movie reviews* positif dan negatif.

Kata kunci

Automatic text categorization, *movie reviews* imdb.com, Naïve Bayes, Support Vector Machines, LIBSVM

ABSTRACT

Automatic text categorization is a technique to automatically classify documents based on their labels or classes. The automatic classification problem is such an important problem to tackle especially when we have a large number of documents (number of documents are more than 1000) because this classification problem is not feasible to be solved manually.

This research aims to compare the accuracy of two famous machine learning algorithms, Naïve Bayes and Support Vector Machines (SVM), which will be used to classify positive and negative movie reviews. The feature of Naïve Bayes algorithm is implemented by utilizing unigram feature while the C -Support *vector classification* model from SVM algorithm is applied by using LIBSVM library.

The experiment of classifying positive and negative imdb.com movie reviews showsthat the accuracy of Naïve Bayes algorithm is better than the one of SVM algorithm.

Keywords

Automatic text categorization, imdb.com*movie reviews*, Naïve Bayes, Support Vector Machines, LIBSVM

Daftar Isi

1. LATAR BELAKANG

Saat ini banyak sekali informasi tersedia dalam bentuk dokumen *on-line*. Para peneliti berusaha menyelidiki masalah *automatic text categorization*sebagai bagian untuk mengorganisir informasi untuk pengguna [1].

Banyak hasil penelitian berfokus pada *topical categorization* dengan cara mengurutkan dokumen-dokumen menurut subjeknya (contoh: *sports* vs *politics*). Akan tetapi, belakangan ini muncul fokus baru yaitu bagaimana mengurutkan atau mengklasifikasikan dokumen-dokumen menurut *sentiment*-nya atau opini keseluruhan terhadap objek pembicaraan (contoh: apakah sebuah *product review* positif atau negatif). *Product review-product review* yang diberi label (positif atau negatif) mampu memberikan rangkuman yang cukup kepada pembaca untuk membantu mereka dalam memilih produk. *Sentiment classification* juga bermanfaat di dalam aplikasi *business intelligence* (contoh: sistem MindfulEye's Lexant[1]) dan *recommender systems* (contoh: Terveen et al. [2], Tatemura [3]) dengan *input* dan *feedback* dari *user* dapat dirangkum secara cepat. Problem lainnya yang dapat diselesaikan dengan *sentiment categorization*adalahmemproses *response* dari sebuah *survey* pengisian form dalam bentuk *natural language*. Lebih lanjut, aplikasi-aplikasi untuk *message filtering* dapat memanfaatkan informasi *sentiment* untuk mengenali dan membuang '*flames*' [4].

Penelitian ini bermaksud untuk menyelidiki keefektifan penggunaan teknik *machine learning* untuk menyelesaikan masalah *sentiment classification*. Aspek yang membedakan masalah ini dengan *topic-based classification* tradisional adalah topik-topik diidentifikasi hanya dengan *keywords*, sedangkan *sentiment* dapat diekspresikan dalam bentuk yang tidak langsung kelihatan. Contohnya, kalimat "*How could anyone sit through this movie?*" mengandung kata-kata yang jelas-jelas tidak negatif namun arti dari kalimat tersebut adalah negatif; film yang dimaksud sangat membosankan [1].

2. PENELITIAN-PENELITIAN SEBELUMNYA

Bab ini membahas *survey* tentang penelitian-penelitian *text categorization* sebelumnya yang berdasar *non-topic*.

[1] http://www.mindfuleye.com/about/lexant.htm

Biber [5] membahas mengenai bagaimana mengklasifikasikan dokumen berdasarkan *source*-nya atau *source style*, dengan *statistically-detected stylistic variation* sebagai *cue*. Contohnya, pengklasifikasian berdasarkan penulis, penerbit (contoh: *the New York Times* vs. *The Daily News*), dan latar belakang bahasa [6 – 9].

Bidang penelitian lain yang berhubungan adalah penelitian mengenai *genre* dari teks; *subjective genre* [9 – 11]. Penelitian lain berusaha mencari *features* yang menunjukkan bahwa *subjective language* digunakan di dalam teks [12, 13]. Meskipun teknik-teknik untuk *genre categorization* dan *subjectivity detection* dapat menyelesaikan masalah klasifikasi, teknik-teknik tersebut tidak dapat menentukan isi dari opini-opini tersebut

Penelitian-penelitian sebelumnya tentang *sentiment-based classification*umumnya menerapkan*knowledge-based* yang parsial. Beberapa hasil penelitian berfokus pada bagaimana mengklasifikasikan *semantic orientation* dari setiap kata atau frase dengan menggunakan *linguistic heuristics* atau himpunan *seed words* yang sudah disiapkan terlebih dahulu [14, 15].Penelitian tentang *sentiment-based classification*yang dikenakan pada seluruh dokumen umumnya menggunakan model *cognitive linguistics*[16, 17] atau *lexicons discriminant-word* yang dibuat secara manual atau semi-manual [18, 19, 20].

Penelitian tentang klasifikasi *review* yang dilakukan oleh Turney [21] juga menggunakan teknik *machine learning* tetapi beliau menggunakan teknik *unsupervised learning* yang khusus. Teknik yang digunakan berdasarkan pada nilai *mutual information* antara *document phrases* dan kata-kata ”**excellent**” dan ”**poor**”. Nilai *mutual information*dihitung berdasarkan data statistik yang diperoleh dari *search engine*.

3. MOVIE REVIEW DATASET

Dataset yang digunakan dalam penelitian ini diunduh dari *website*pribadi Prof. Lillian Lee[2]. Dataset terdiri dari 2000 *movie reviews*[3] yang dibagi dua yaitu *review* positif dan negatif.Selain *movie reviews*, dataset juga memuat daftar *stop word* yang dapat digunakan untuk proses *preprocessing*. Tabel 1 dan tabel 2 menggambarkan masing-masing satu contoh *movie review* yang positif dan negatif.

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/review_polarity.tar.gz
[3] Movie reviews dari www.imdb.com

you've got mail works alot better than it deserves to .
in order to make the film a success , all they had to do was cast two extremely popular and attractive stars , have them share the screen for about two hours and then collect the profits .
no real acting was involved and there is not an original or inventive bone in it's body (it's basically a complete re-shoot of the shop around the corner , only adding a few modern twists) .
essentially , it goes against and defies all concepts of good contemporary filmmaking .
it's overly sentimental and at times terribly mushy , not to mention very manipulative .
but oh , how enjoyable that manipulation is .
but there must be something other than the casting and manipulation that makes the movie work as well as it does , because i absolutely hated the previous ryan/hanks teaming , sleepless in seattle .
it couldn't have been the directing , because both films were helmed by the same woman .
i haven't quite yet figured out what i liked so much about you've got mail , but then again , is that really important ?
if you like something so much , why even question it ?
again , the storyline is as cliched as they come .
tom hanks plays joe fox , the insanely likeable owner of a discount book chain and meg ryan plays kathleen kelley , the even more insanely likeable proprietor of a family-run children's book shop called , in a nice homage , the shop around the corner .
fox and kelley soon become bitter rivals because the new fox books store is opening up right across the block from the small business .
little do they know , they are already in love with each other over the internet , only neither party knows the other person's true identity .
the rest of the story isn't important because all it does is serve as a mere backdrop for the two stars to share the screen .
sure , there are some mildly interesting subplots , but they all fail in comparison to the utter cuteness of the main relationship .
all of this , of course , leads up to the predictable climax .
but as foreseeable as the ending is , it's so damn cute and well-done that i doubt any movie in the entire year contains a scene the evokes as much pure joy as this part does .
when ryan discovers the true identity of her online love , i was filled with such , for lack of a better word , happiness that for the first time all year , i actually left the theater smiling .

Tabel 1 Contoh movie review yang positif

the happy bastard's quick movie review
damn that y2k bug .
it's got a head start in this movie starring jamie lee curtis and another baldwin brother (william this time) in a story regarding a crew of a tugboat that comes across a deserted russian tech ship that has a strangeness to it when they kick the power back on .
little do they know the power within . . .
going for the gore and bringing on a few action sequences here and there , virus still feels very empty , like a movie going for all flash and no substance .
we don't know why the crew was really out in the middle of nowhere , we don't know the origin of what took over the ship (just that a big pink flashy thing hit the mir) , and , of course , we don't know why donald sutherland is stumbling around drunkenly throughout .
here , it's just " hey , let's chase these people around with some robots " .
the acting is below average , even from the likes of curtis .

you're more likely to get a kick out of her work in halloween h20 . sutherland is wasted and baldwin , well , he's acting like a baldwin , of course . the real star here are stan winston's robot design , some schnazzy cgi , and the occasional good gore shot , like picking into someone's brain . so , if robots and body parts really turn you on , here's your movie . otherwise , it's pretty much a sunken ship of a movie .

Tabel 2 Contoh movie review yang negative

4. MASALAH KLASIFIKASI TEKS

Dalam klasifikasi teks, kita mempunyai deskripsi dokumen, $d \in \mathbb{X}$, dengan $\mathbb{X}$ adalah*document space*, dan himpunan dari kelas-kelas, $\mathbb{C} = \{c_1, c_2, \ldots, c_j\}$. Kelas-kelas ini disebut juga sebagai kategori atau label. Selanjutnya, kita definisikan himpunan dokumen latih $\mathbb{D}$ yang berisi dokumen dan labelnya (d, c), dengan $(d, c) \in \mathbb{X} \times \mathbb{C}$ [22]. Contoh:

$$(d, c) = (\text{Beijing joins the World Trade Organization}, \quad China)$$

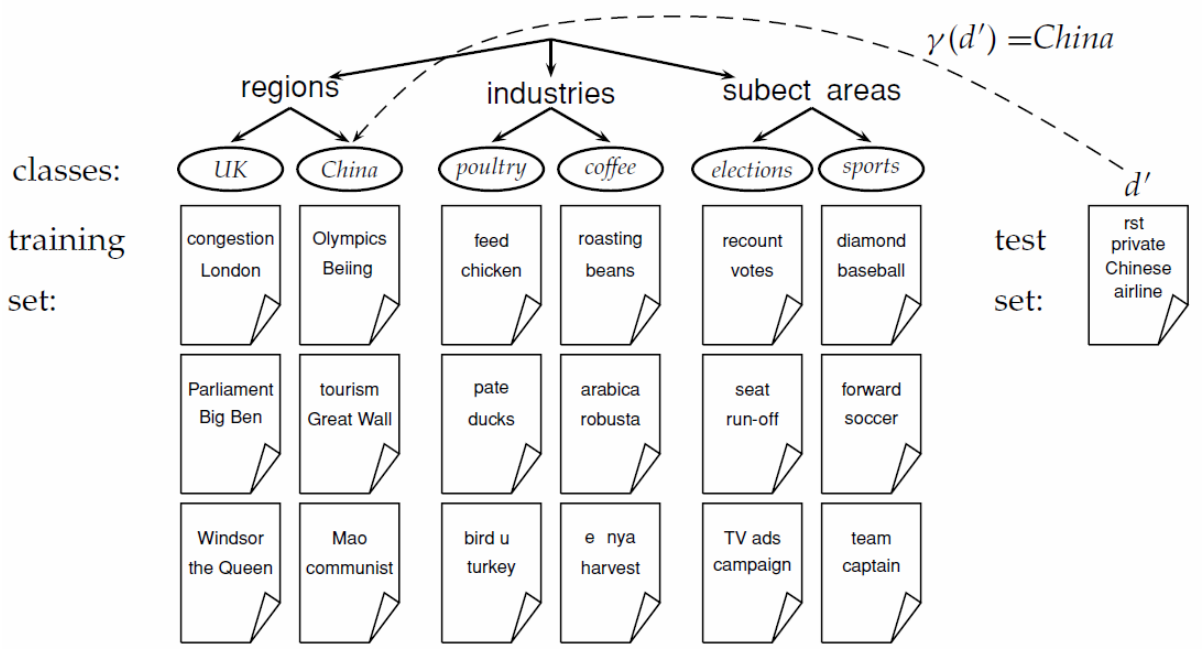
untuk dokumen berisi satu kalimat, $Beijing\ joins\ the\ World\ Trade\ Organization$ dan labelnya $China$.

Dengan menggunakan metode atau algoritma pembelajaran, kita ingin membentuk sebuah *classifier* atau *classification function*γ yang memetakan dokumen-dokumen menjadi kelas-kelas, sebagai berikut:

$$\gamma : \mathbb{X} \rightarrow \mathbb{C}$$

Jenis pembelajaran ini disebut *supervised learning* karena teknik ini seperti seorang *supervisor* (manusia yang mendefinisikan kelas-kelas dan label-label untuk dokumen latih)yang memegang peran untuk mengarahkan proses pembelajaran; ciri *supervised learning* juga adalah data latihnya yang memiliki label atau kelas. Kita memberikan notasi metode pembelajaran dengan simbol Γ dan menulis $\Gamma(\mathbb{D}) = \gamma$. Metode pembelajaran Γ menerima *training data*$\mathbb{D}$ sebagai input dan mengembalikan *output* berupa fungsi klasifikasi γ .

Gambar 1 memperlihatkan contoh klasifikasi teks dari koleksi artikel **Reuters-RCV1**. **Reuters-RCV1** mempunyai 6 (enam) kelas (UK, China, ..., sports) dan setiap kelas memiliki data latihnya masing-masing.Apabila kita mempunyai *classifier*γ , kita dapat menggunakan *classifier* tersebut pada data ujitersebut untuk menentukan labelnya,*first private Chinese airline*, yang belum diketahui kelas atau labelnya.



Gambar 1 Kelas, data latih (*training set*), dan data uji (*test set*) dalam klasifikasi teks [22].

Classifier di Gambar 1 memberikan label *China* kepada dokumen *first private Chinese airline*. Metode pembelajaran yang digunakan dalam penelitian ini metode **Naive Bayes** dan **Support Vector Machines**.

4.1 Klasifikasi teks Naïve Bayes

Metode klasifikasi teks Naive Bayes menggunakan model *multinomial***Naive Bayes** (NB) yang merupakan metode pembelajaran probabilistik [22]. Probabilitas sebuah dokumen d berada di kelas c dihitung dengan formula

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c) \qquad (1)$$

dengan $P(t_k|c)$ adalah *conditional probability* dari term t_k berada di dalam sebuah dokumen yang memiliki kelas c. $(t_1, t_2, \ldots, t_{n_d})$ adalah *token-token* di dalam d. Contohnya, $(t_1, t_2, \ldots, t_{n_d})$ untuk dokumen ***Beijing and Taipei join the WTO*** adalah $(\mathbf{Beijing, Taipei, join, WTO})$ dengan $n_d = 4$, jika kita menganggap ***and*** dan ***the*** sebagai *stop words*.

Dalam klasifikasi teks, tujuan kita adalah mencari kelas **terbaik** untuk suatu dokumen. Kelas terbaik dalam klasifikasi teks NB adalah kelas *maximum a posteriori* (MAP) c_{map} :

$$c_{map} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k|c) \quad (2)$$

Kita menulis $\hat{P}$ untuk P karena kita tidak mengetahui nilai sebenarnya (*true value*) dari parameter-parameter $P(c)$ dan $P(t_k|c)$; meskipun demikian, kita dapat menaksir semua parameter tersebut dari data latih (*training set*). Di persamaan (2) terdapat banyak *conditional probabilities* yang dikalikan satu dengan yang lainnya; hal ini dapat mengakibatkan hasil perkalian adalah bilangan yang kecil sekali dan lebih kecil daripada yang komputer dapat simpan di memori. Oleh karena itu, komputasi lebih baik dilakukan dengan menambahkan logaritma dari *probability* daripada mengalikan *probability*. Kelas dengan skor *log probability*tertinggi masih merupakan kelas yang paling mungkin karena sifat $\log(xy) = \log(x) + \log(y)$ dan fungsi logaritma adalah monotonik. Persamaan (2) dapat ditulis menjadi

$$c_{map} = \arg\max_{c \in \mathbb{C}} \left[\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c) \right] \quad (3)$$

Persamaan (3) mempunyai interpretasi yang sederhana. Setiap *conditional parameter*$\log \hat{P}(t_k|c)$ adalah bobot yang mengatakan seberapa baik indikator t_k untuk kelas c. Demikian juga, *prior*$\log \hat{P}(c)$ adalah bobot yang merupakan frekuensi relatif dari c. Jumlah dari log *prior* dan bobot-bobot dari *term*adalah ukuran seberapa banyak bukti atau *evidence*untuk dokumen tersebut berada di kelas c dan persamaan (3) memilih kelas yang memiliki paling banyak *evidence.*

Nilai $\hat{P}(c)$dan $P(t_k|c)$dihitung dengan menggunakan teknik *maximum likelihood estimate* (MLE) sehingga kita memperoleh

$$\hat{P}(c) = \frac{N_c}{N} \quad (4)$$

dan $P(t_k|c)$ adalah frekuensi relatif *term*t di dokumen yang berlabel kelas c :

$$P(t_k|c) = \frac{T_{ct}}{\sum_{t \in V} T_{ct}}, \quad (5)$$

dengan T_{ct} adalah banyak kemunculan t di dokumen latih dari kelas c .

Masalah dengan penaksir MLE adalah nilai nol untuk kombinasi *term*-kelas yang tidak muncul di data latih. Contohnya, jika *term* WTO di data latih hanya muncul di dokumen *China*, penaksir MLE untuk kelas-kelas lainnya, contohnya UK, adalah nol:

$$\hat{P}(\text{WTO}|UK) = 0 \quad . \tag{6}$$

Nilai nol ini dapat dieliminasi dengan menggunakan *add-one* atau *Laplace smoothing*:

$$\hat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t_a \in V}(T_{ct}+1)} = \frac{T_{ct}+1}{\left(\sum_{t_a \in V} T_{ct}\right)+\hat{B}} \tag{7}$$

dengan $B = |V|$ adalah banyaknya *term* di dalam *vocabulary*.

4.2 Contoh klasifikasi teks Naïve Bayes

Diberikan data latih dan data uji pada Tabel 3.

	docID	Kata-kata dokumen	c = China?
Dokumen latih	1	Chinese Beijing Chinese	Yes
	2	Chinese Chinese Shanghai	Yes
	3	Chinese Macao	Yes
	4	Tokyo Japan Chinese	No
Dokumen uji	5	Chinese Chinese Chinese Tokyo Japan	?

Tabel 3 Contoh data untuk klasifikasi teks Naïve Bayes

Prior dari contoh di Tabel 3 adalah $\hat{P}(c) = \frac{3}{4}$ dan $\hat{P}(\bar{c}) = \frac{1}{4}$. Selanjutnya kita dapat menghitung *conditional probabilities*:

$$\hat{P}(\text{Chinese}|c) = \frac{(5+1)}{(8+6)} = \frac{6}{14} = \frac{3}{7}$$

$$\hat{P}(\text{Tokyo}|c) = \hat{P}(\text{Japan}|c) = \frac{(0+1)}{(8+6)} = \frac{1}{14}$$

$$\hat{P}(\text{Chinese}|\bar{c}) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$

$$\hat{P}(\text{Tokyo}|\bar{c}) = \hat{P}(\text{Japan}|\bar{c}) = \frac{(1+1)}{(3+6)} = \frac{2}{9}$$

Penyebut-penyebutnya adalah $(8+6)$ dan $(3+6)$ karena panjang $text_c$ dan $text_{\bar{c}}$ adalah masing-masing 8 dan 3 dan konstanta B di persamaan 7 sebagai *vocabulary* adalah enam *terms*. Kemudian kita peroleh:

$$\hat{P}(c|d_5) \propto \frac{3}{4} \times \left(\frac{3}{7}\right)^3 \times \frac{1}{14} \times \frac{1}{14} \approx 0.0003 \quad .$$

$$\hat{P}(\bar{c}|d_5) \propto \frac{1}{4} \times \left(\frac{2}{9}\right)^3 \times \frac{2}{9} \times \frac{2}{9} \approx 0.0001 \quad .$$

Karena $\hat{P}(c|d_5) > \hat{P}(\bar{c}|d_5)$, *classifier* memberikan label $c = China$ kepada dokumen uji.

4.3 Klasifikasi dengan Support Vector Machines (SVM)

Support vector machines (SVM) adalah metode *machine learning* yang popular untuk menyelesaikan masalah klasifikasi (*classification*) dan regresi (*regression*). Penelitian ini menggunakan *library* SVM yang bernama LIBSVM[4] untuk mengklasifikasikan *movie reviews* antara *movie review* positif dan negatif.

Penggunaan LIBSVM membutuhkan dua langkah sebagai berikut [23]:

1. Melatih *data set* untuk memperoleh sebuah model.
2. Menggunakan model yang diperoleh di langkah nomor 1 untuk memprediksi informasi yang berkaitan dengan *testing data set*.

Formula SVM yang digunakan dalam penelitian ini adalah C-Support *vector classification*.Diberikan vektor-vektor $x_i \in R^n$, $i = 1, \dots, l$, yang masing-masing memiliki atau label 1 atau label 2, dan sebuah vektor indicator $y \in R^l$ sedemikian sehingga $y_i \in \{1, -1\}$, C-SVC [24–25] menyelesaikan masalah *primal optimization.*

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \quad (8)$$

subject to $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$,

$\xi_i \geq 0, i = 1, \dots, l$,

dengan $\phi(x_i)$ memetakan x_i ke ruang yang berdimensi lebih tinggi (*a higher-dimensional space*) dan $C > 0$ adalah *regularization parameter*. Karena vektor variabel w berkemungkinan memiliki dimensi yang tinggi, masalah *primal optimization* dapat ditulis menjadi*dual problem* berikut:

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \quad (9)$$

subject to $y^T \alpha = 0$,

$0 \leq \alpha_i \leq C$, $i = 1, \dots, l$,

[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

dengan $e = [1, \ldots, 1]^T$ adalah vektor yang semuanya berisi elemen satu, Q adalah matriks *semidefinite* positif berukuran l kali l , $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, dan $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ adalah fungsi kernel.

Apabila persamaan (9) diselesaikan, nilai optimum w memenuhi

$$w = \sum_{i=1}^{l} y_i \alpha_i \phi(x_i) \tag{10}$$

dan fungsi keputusan (*decision function*) adalah

$$\operatorname{sgn}(w^T \phi(x) + b) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b \right).$$

Kita simpan $y_i \alpha_i \; \forall i, b$, nama label, *vector support*, dan informasi lainnya seperti *parameterkernel* di dalam model yang akan digunakan untuk memprediksi.

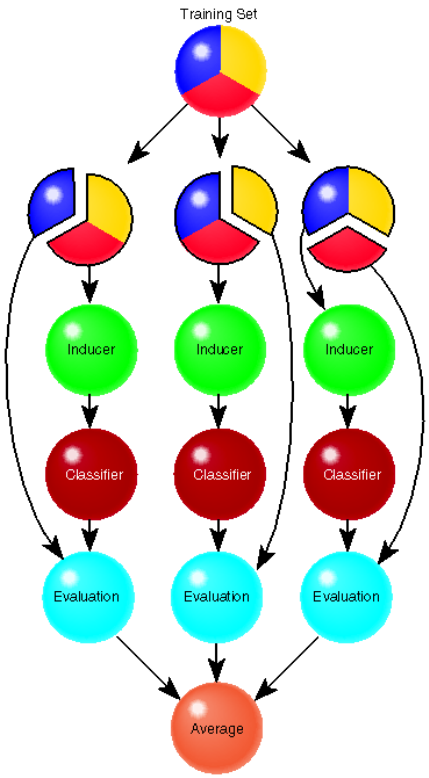
4.4 N-fold cross validation

Jumlah data memegang peranan penting di dalam algoritma *machine learning*. Jumlah data yang sedikit (< 100 *instance*) mungkin membuat algoritma *machine learning* tidak akurat.Algoritma *machine learning* merekomendasikan jumlah *instance* yang banyak (> 1000 *instance*) namun data itu sendiri tidak mudah untuk diperoleh; data memerlukan biaya dan harga data biasanya mahal.

N-fold cross validation adalah teknik yang dapat digunakan apabila kita memiliki jumlah data yang terbatas (jumlah *instance* tidak banyak). Cara kerja N-*fold cross validation* adalah sebagai berikut:

1. Total *instance* dibagi menjadi N bagian.
2. *Fold* ke-1 adalah ketika bagian ke-1 menjadi data uji (*testing data*) dan sisanya menjadi data latih (*training data*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.
3. *Fold* ke-2 adalah ketika bagian ke-2 menjadi data uji (*testing data*) dan sisanya menjadi data latih (*training data*). Selanjutnya, hitung akurasi berdasarkan porsi data tersebut.
4. Demikian seterusnya hingga mencapai *fold* ke-N.

5. Hitung rata-rata akurasi dari N buah akurasi di atas. Rata-rata akurasi ini menjadi akurasi final.

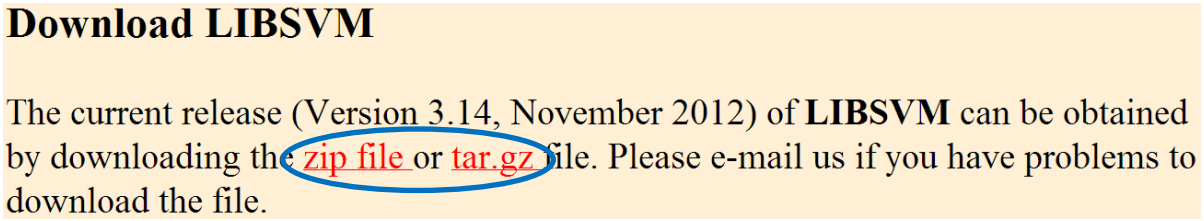


Gambar 2 Ilustrasi dari 3-*fold cross validation*

Gambar 2 mengilustrasikan 3-*fold cross validation*secara umum [26]. Dalam penelitian ini, kita membentuk *classifier*dan tidak mempunyai *inducer*.

5. INSTALASI LIBSVM UNTUK PYTHON 2.7

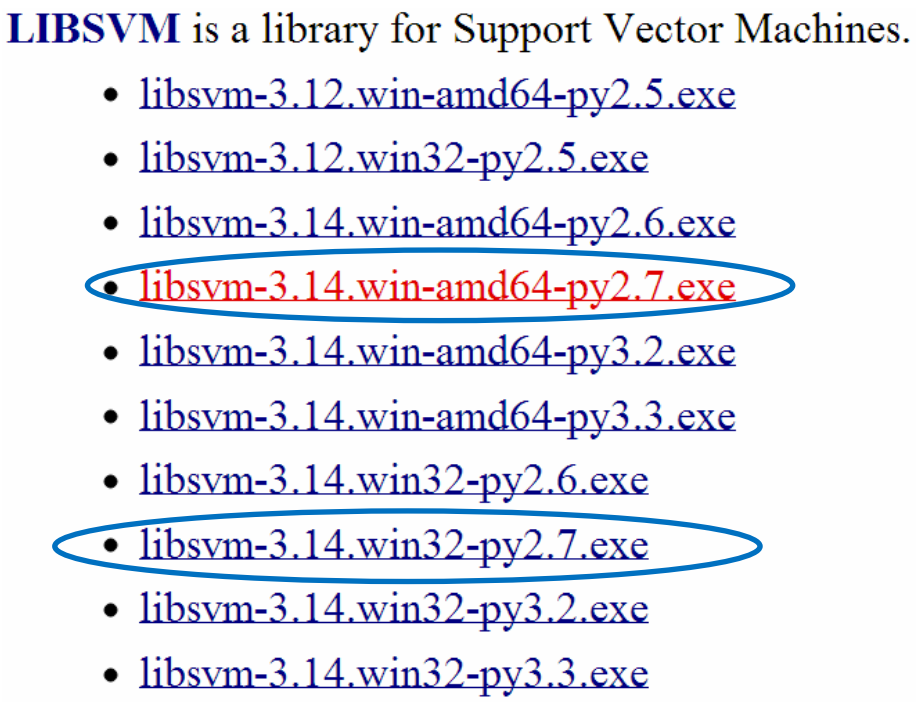
Setelah Python 2.7 sudah terinstall, kita mengunduh `libsvm` di *website* libsvm[5] seperti di gambar 2.



Gambar 3 Tampilan download LIBSVM

Selanjutnya, kita mengunduh *Python Extension packages*khusus untuk LIBSVM[6] seperti di gambar 3. *Installer* dapat disesuaikan dengan versi Windows yang digunakan; apakah versi Windows 32 atau 64 bit?

[5]http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[6] http://www.lfd.uci.edu/~gohlke/pythonlibs/



Gambar 4 Tampilan download Python *extension* untuk LIBSVM

Untuk mengkode bahasa pemrograman Python, kita dapat menginstall *code editor*, Wing IDE 101[7]. Wing IDE 101 dapat diunduh seperti di gambar 4 dan gratis.

Gambar 5 Tampilan download Wing IDE 101 untuk versi Windows

6. TUTORIAL SINGKAT MENGGUNAKAN LIBSVM

Dalam file yang diunduh (`libsvm-3.12.zip`), terdapat *data set* sampel mengenai jantung yang dapat digunakan untuk uji coba. *Data set* mempunyai 270 *instance*; sebagai gambaran, sampel dideskripsikan oleh tabel 4.

+1 1:0.708333 2:1 3:1 4:-0.320755 5:-0.105023 6:-1 7:1 8:-0.419847 9:-1 10:-0.225806 12:1 13:-1
-1 1:0.583333 2:-1 3:0.333333 4:-0.603774 5:1 6:-1 7:1 8:0.358779 9:-1 10:-0.483871 12:-1 13:1
+1 1:0.166667 2:1 3:-0.333333 4:-0.433962 5:-0.383562 6:-1 7:-1 8:0.0687023 9:-1 10:-0.903226 11:-1 12:-1 13:1

Tabel 4 Contoh tiga buah*instance* dari data set sampel

Kolom pertama menggambarkan label dari *instance*; dalam contoh ini hanya ada dua buah label adalah +1 dan -1. Kolom kedua dan seterusnya memiliki format <angka1>:<angka2>. Nilai <angka1> adalah nomor *attribute* dan <angka2> adalah nilai dari *attribute* tersebut.

[7] http://www.wingware.com/downloads/wingide-101/4.1.9-1/binaries

Dalam penelitian ini, kita menggunakan fungsi-fungsi *utility* yang tersedia di `svmutil.py`. Contoh penggunaan fungsi-fungsi di modul `svmutil.py`ini dijabarkan di tabel 5.

```
1. >>> from svmutil import *

2. # Read data in LIBSVM format
3. >>> y, x = svm_read_problem('../heart_scale')
4. >>> m = svm_train(y[:200], x[:200], '-c 4')
5. >>> p_label, p_acc, p_val = svm_predict(y[200:], x[200:], m)
```

Tabel 5 Menggunakan fungsi-fungsi di LIBSVM

Baris 1 menjelaskan bahwa kita mesti meng-*import*`module``svmutil` sebelum kita dapat menggunakan fungsi-fungsi di dalam `svmutil`.Baris 3 menjelaskan cara untuk membaca *file data set*(`heart_scale`) yang sudah memiliki format LIBSVM.Hasil baca disimpan variabel `y` dan `x`. Baris 4 menjelaskan bahwa kita menggunakan 200 buah *instance* dari baris ke-0 sampai dengan baris ke-199 sebagai data latih. Flag '-c 4' berarti kita menggunakan formula C -Support *vector classification*dengan $C = 4$.*Output* dari baris 4 adalah model yang dapat digunakan untuk memprediksi data uji.Baris 5 menjelaskan cara untuk memprediksi label dengan menggunakan *instance* ke-200 sampai dengan *instance* ke-269.*Output* dari baris 5 digambarkan di gambar 6.

```
>>> p_label, p_acc, p_val = svm_predict(y[200:], x[200:], m)
Accuracy = 84.2857% (59/70) (classification)
```

Gambar 6*Output* dari pemanggilan fungsi `svm_predict`

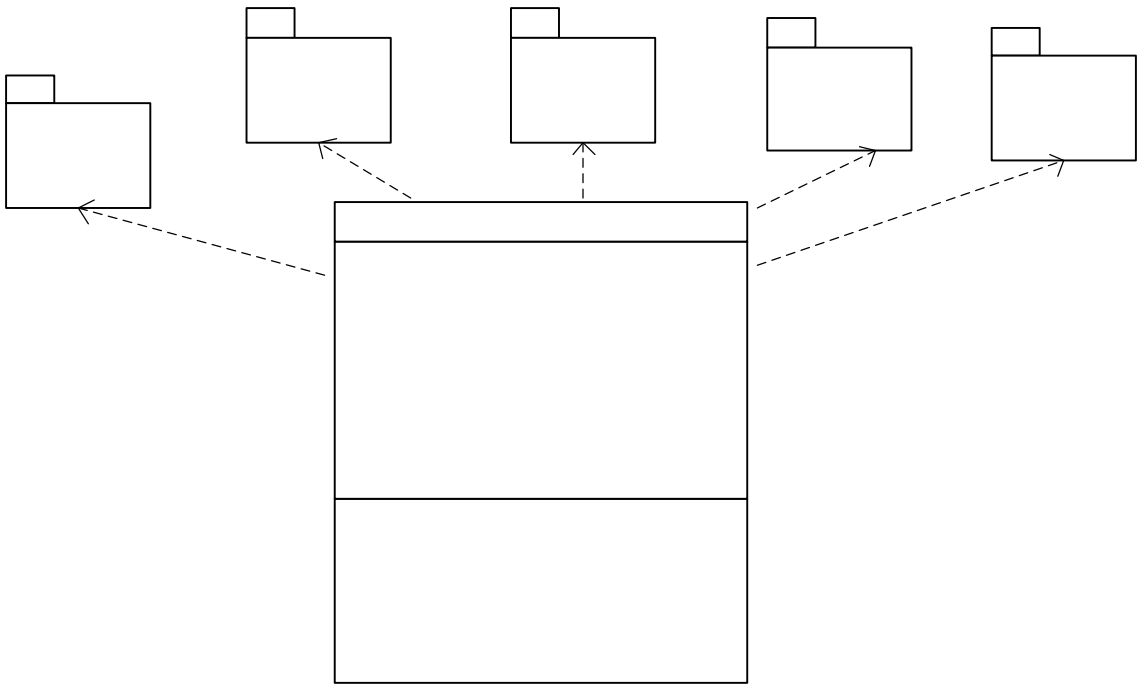
Variabel `p_label` adalah label-label hasil prediksi berdasarkan model, `m`. Variabel `p_acc` berisi *tuple* dengan elemen pertama adalah *accuracy*, elemen kedua adalah *mean-squared error*, dan elemen ketiga adalah *squared correlation coefficient*. Terakhir,`p_vals` adalah *list*yang berisi *decision values*atau *probability estimates*.

7. DESAIN CLASS DIAGRAM

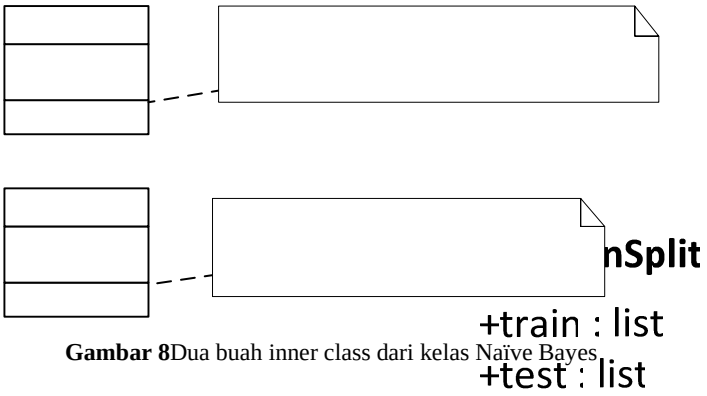
Desain kelas `NaiveBayes` (NB) dideskripsikanpada gambar 7. Kelas NB jugamempunyai 2 buah *inner class* (kelas di dalam kelas), yaitu `TrainSplit` dan `Example`. Kelas `TrainSplit`adalah struktur data untuk menyimpan setiap *fold* (*training data* + *testing data*).Kemudian kelas `Example` berfungsi untuk menyimpan setiap *instance* dari *training data*. Sebuah *instance*terdiri dari *words*, yaitu *list* dari *words*, dan kelasnya (atau positif atau negatif).Dua kelas ini direpresentasikan pada gambar 8.

Kelas NB, seperti yang dijelaskan oleh gambar 7, memerlukan beberapa kelas dari *library* Python, yaitu:

1. `defaultdict`: struktur data *dictionary*; struktur data yang memiliki *key* dan *value*; mirip dengan `java.util.Map`.
2. `getopt`: kelas untuk mem-*parsingargument-argument* di *command prompt*. Kelas ini dibutuhkan untuk membuat *options* dan *arguments* dari eksekusi program di *command prompt*.
3. `math`: kelas matematika; sistem yang dibangun membutuhkan kelas `math` untuk menghitung logaritma.
4. `sys`: kelas untuk mengambil *argument-argument* di *command prompt*.
5. `os`: kelas yang digunakan untuk me-*listingfile-file* di sebuah direktori; kelas ini berhubungan dengan proses *input-output*.



Gambar 7 Diagram kelas Naïve Bayes



Gambar 8 Dua buah inner class dari kelas Naïve Bayes

No.	Nama attributes	Tipe data	Deskripsi
1.	`stopList`	*set*	Daftar *stopwords* yang direpresentasikan oleh *attribute* bertipe himpunan (*set*)
2.	`FILTER_STOP_WORDS`	*boolean*	*Flag* untuk menandai apakah sistem menggunakan *stopwords* (`true`) atau tidak (`false`).
3.	`numFolds`	*integer*	Jumlah *fold* yang hendak digunakan.
4.	`kvmap`	*dictionary*	*Dictionary* untuk menyimpan semua *term* dan frekuensinya.
5.	`prior`	*dictionary*	Struktur data untuk menyimpan *prior* dari kelas positif dan negatif. Formula menghitung prior sesuai dengan persamaan (4)
6.	`T`	*dictionary of dictionary*	`T` dibagi menjadi menjadi dua (2), yaitu: `T` untuk kelas positif dan `T` untuk kelas negatif. Rumus menghitung `T` sesuai dengan T_{ct} di persamaan (7).
7.	`condProb`	*dictionary of dictionary*	Struktur data untuk menyimpan nilai $\hat{P}(t\|c)$ sesuai dengan persamaan (7).
8.	`total`	*dictionary*	Struktur data untuk menyimpan kelas-kelas dan frekuensi dari setiap kelasnya.

Tabel 6 Attributes dari kelas Naïve Bayes

No.	Nama methods	Tipe output	Deskripsi
1.	`addExample`	*void*	Menambah *instance* ke dalam *training*

			data.*Method* ini juga menghitung frekuensi untuk pembuatan *vocabulary*, pembuatan *prior*, dan membangun `T`.
2.	`buildSplits`	*list* dari *fold*	Membangun *split* atau *fold* untuk *training* dan *testing*.
3.	`classify`	*string*	Menentukan kelas dari *list of terms* (*words*).
4.	`constructCondProb`	*void*	Menghitung *prior* dari setiap kelas dan total semua *conditional probabilities*.
5.	`filterStopWords`	*list*dari*words*	Membuang *word* yang merupakan *stopwords*.
6.	`readFile`	*list* dari *words*	Membaca isi *file* dan mem-*parsing*-nya menjadi *list* dari *terms*.
7.	`segmentWords`	*list* dari *words*	Memisahkan kalimat menjadi *term*-*term* dan hasil pemisahan ini disimpan dalam *list*.

Tabel 7 Methods dari kelas Naïve Bayes

8. IMPLEMENTASI ALGORITMA NAIVE BAYES

Algoritma Naive Bayes yang diimplementasikan dalam penelitian ini berdasarkan *pseudo code*dari buku *Introduction to Information Retrieval* oleh Christopher Manning, et al [22].

TRAINMULTINOMIALNB($\mathbb{C}, \mathbb{D}$)
1 $V \leftarrow$ EXTRACTVOCABULARY($\mathbb{D}$)
2 $N \leftarrow$ COUNTDOCS($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ COUNTDOCSINCLASS($\mathbb{D}, c$)
5 $prior[c] \leftarrow N_c / N$
6 $text_c \leftarrow$ CONCATENATETEXTOFALLDOCSINCLASS($\mathbb{D}, c$)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ COUNTTOKENSOFTERM($text_c, t$)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** $V, prior, condprob$

APPLYMULTINOMIALNB($\mathbb{C}, V, prior, condprob, d$)
1 $W \leftarrow$ EXTRACTTOKENSFROMDOC(V, d)
2 **for each** $c \in \mathbb{C}$
3 **do** $score[c] \leftarrow \log prior[c]$
4 **for each** $t \in W$
5 **do** $score[c] \mathrel{+}= \log condprob[t][c]$
6 **return** $\arg\max_{c \in \mathbb{C}} score[c]$

Gambar 9 Pseudo code dari algoritma Naïve Bayes: Training dan Testing

Method `extractVocabulary` di baris ke-1 berfungsi untuk membangun *vocabulary* dari semua dokumen latih (*training document*) yang ada.Teknikmembangun *vocabulary*-nya dijelaskan di tabel 8.Implementasi*vocabulary*ada di dalam variabel `kvmap` yang bertipe *dictionary*

```
 # ========================
 #  membangun Vocabulary
 # ========================
for k in words:
```

```
        self.kvmap[k] = self.kvmap[k] + 1
```

Tabel 8 Membangun *vocabulary* untuk setiap *term*

Vocabulary adalah *keys* dari `kvmap`seperti pada tabel 9; di kode, *vocabulary* direpresentasikan oleh variabel `allVocabs`.

```
allVocabs = self.kvmap.keys()
```

Tabel 9`allVocabs` adalah semua *term* yang unik

Baris ke-2 menceritakan langkah untuk menghitung semua dokumen latih (*training document*) yang ada.Langkah-langkah berikutnya dapat diverifikasi sesuai dengan contoh di subbab 4.2 dan formula (7).

9. HASIL ALGORITMA NAIVE BAYES

*Setting dataset*untuk eksperimen algoritma Naive Bayes (NB)dalam penelitian ini adalah sebagai berikut:

1. Total *dataset*adalah 2000 buah *movie reviews*.
2. Teknik pembagian *dataset* adalah 10-*fold cross validation*.

Satu *fold* terdiri 1800 *movie reviews*sebagai *training data*dan 200 *reviews* sebagai *testing data*. Keakuratan dari algoritma dihitung dengan formula di persamaan

$$\text{Keakuratan} = \frac{\#\text{prediksi yang benar}}{\#\text{instance di testing data}} = \frac{\#\text{prediksi yang benar}}{200} \quad (11).$$

Hasil 10-*fold cross validation*untuk algoritma Naive Bayes ditampilkan di tabel 10.

Fold ke-	Keakuratan
1	76%
2	82.5%
3	82.5%

4	83%
5	80%
6	83%
7	83%
8	83.5%
9	75.5%
10	82%
Rata-Rata =	**81.1%**

Tabel 10 Hasil akurasi algoritma NB untuk 10-*fold cross validation*dan rata-rata akurasinya

Perintah untuk menjalankan eksperimen 10-*fold cross validation* algoritma Naive Bayes adalah **`python mymath.py –f  ..\data\imdb1`**.

Hasil rata-rata akurasi Naïve Bayes sebesar 81.1% menunjukkan bahwa algoritma Naïve Bayes cukup baik untuk menyelesaikan masalah klasifikasi teks untuk *data set movie reviews* imdb.com.

10.HASIL ALGORITMA SUPPORT VECTOR MACHINES

Dengan menggunakan *library*LIBSVM pada *dataset*dan *setting*yang sama seperti untuk algoritma Naive Bayes, hasil eksperimen yang diperoleh adalah seperti pada tabel 11.

Fold ke-	Keakuratan
1	71%
2	69.5%
3	67%
4	63%
5	60.5%
6	68.5%
7	71%
8	65.5%
9	73%
10	66%
Rata-Rata =	**67.5%**

Tabel 11 Hasil akurasi algoritma LIBSVM untuk 10-*fold cross validation* dan rata-rata akurasinya

Rata-rata akurasi LIBSVM sebesar 67.5% menunjukkan bahwa algoritma LIBSVM dengan setting, yaitu C -Support *vector classification* dan nilai $C = 4$ kurang cocok digunakan untuk mengklasifikasi imdb *movie reviews*.

11.SIMPULAN

Penelitian ini berhasil membandingkan keakuratan algoritma Naïve Bayes dan LIBSVM untuk *dataset movie reviews*imdb.com. Dalam penelitian ini, keakuratan algoritma Naïve Bayes lebih baik daripada algoritma LIBSVM; hal ini bertentangan dengan hasil penelitian yang dilakukan oleh Pang, Lee, dan Vaithyanathan[1].Kami menduga bahwa penyebab perbedaan ini adalah *setting parameter* untuk algoritma LIBSVM. Dalam penelitian ini, kami menggunakan *setting default* dari LIBSVM, yaitu C -Support *vector classification* dengan nilai $C = 4$.

Oleh karena itu, saran untuk penelitian berikutnya adalah menyelidiki *setting parameter* dari LIBSVM. Penyelidikan mengenai *parameter* LIBSVM ini akan bersinggungan dengan bidang *convex optimization* yang merupakan bidang yang sangat menarik.

12.DESKRIPSI PEKERJAAN

Deskripsi pekerjaan adalah sebagai berikut :

a. Tjatur Kandaga, S.Si., M.T.sebagai Analis, Desain, dan Dokumentasi.
b. Hendra Bunyamin, S.Si., M.T. sebagai Analis dan Implementasi.
c. Diana Trivena Yulianti, S. Kom., M.T. Sebagai Analis dan Ujicoba.

13.REFERENCES

[1] Pang, B., Lee, L., dan Vaithyanathan, S. (2002) **Thumbs up? Sentiment Classification using Machine Learning Techniques**. In Proceedings of EMNLP 2002.

[2] Terveen, L., Hill, W., Amento, B., McDonald, D., dan Creter, J. (1997) **PHOAKS: System for sharing recommendations**. Communications of the ACM.

[3] Tatemura, J. (2000) **Virtual reviewers for collaborative exploration of movie reviews**. In Proceedings of the 5th International Conference on Intelligent User Interfaces.

[4] Spertus, E. (1997) **Smokey: Automatic recognition of hostile messages**. In Proceedings of Innovative Applications of Artificial Intelligence (IAAI).

[5] Biber, D. (1988) **Variation across Speech and Writing**. Cambridge University Press.

[6] Mosteller, F. dan Wallace, D. L. (1984) **Applied Bayesian and Classical Inference: The Case of the Federalist Papers**. Springer-Verlag.

[7] Argamon-Engelson, S., Koppel, M., dan Avneri, G. (1998) **Style-based text categorization: What newspaper am I reading?** In Proceedings of the AAAI Workshop on Text Categorization.

[8] Tomokiyo, L. M. dan Jones, R. (2001) **You're not from round here, are you? Naive-Bayes detection of non-native utterance text**. In Proceedings of the Second NAACL.

[9] Kessler, B, Nunberg, G., dan Schutze, H. (1997) **Automatic detection of text genre**. In Proceedings of the 35th ACL/8th EACL.

[10] Karlgren, J. dan Cutting, D. (1994) **Recognizing text genres with simple metrics using discriminant analysis**. In Proceedings of COLING.

[11] Finn, A., Kushmerick, N., dan Smyth B. (2002) **Genre classification and domain transfer for information filtering**. In Proceedings of the European Colloquium on Information Retrieval Research, Glasgow.

[12] Hatzivassiloglou, V., dan Wiebe, J. (2000) **Effects of adjective orientation and gradability on sentence subjectivity**. In Proceedings of COLING.

[13] Wiebe, J.M., Wilson, T., dan Bell, M. (2001) **Identifying collocations for recognizing opinions**. In Proceedings of the ACL/EACL Workshop on Collocation.

[14] Hatzivassiloglou, V. dan McKeown, K. (1997) **Predicting the semantic orientation of adjectives**. In Proceedings of the 35th ACL/8th EACL.

[15] Turney, P.D. dan Littman, M.L. (2002) **Unsupervised learning of semantic orientation from a hundred-billion word corpus**. Technical Report EGB-1094, National Research Council Canada.

[16] Heast, M. (1992) **Direction-based text interpretation as an information access refinement**. In Paul Jacobs, editor, *Text-Based Intelligent Systems*. Lawrence Erlbaum Associates.

[17] Sack, W. (1994) **On the computation of point of view**. In Proceedings of the Twelfth AAAI. Student abstract.

[18] Huettner, A. dan Subasic, P. (2000) **Fuzzy typing for document management**. In ACL 2000 Companion Volume: Tutorial Abstracts and Demonstration Notes.

[19] Das, S. dan Chen, M. (2001)**Yahoo! for Amazon: Extracting market sentiment from stock message boards**. In Proceedings of the 8th Asia Pasific Finance Association Annual Conference (APFA 2001).

[20] Tong, R. M. (2001)**An operational system for detecting and tracking opinions in on-line discussion**. Workshop note, SIGIR 2001 Workshop on Operational TextClassification.

[21] Turney, P. (2002) **Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews**. In Proceedings of the ACL.

[22] Manning, C.D., Raghavan, P., dan Schutze, H. (2009) **Introduction to Information Retrieval**. Cambridge UP.

[23] Chang, C.C. dan Lin, C. J. (2011) **LIBSVM: a library for support vector machines**. ACM Transactions on Intelligent Systems and Technology.

[24] Boser, B.E., Guyon, I., dan Vapnik, V. (1992) **A training algorithm for optimal margin classifiers**. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory. ACM Press.

[25] Cortes, C. dan Vapnik, V. (1995) **Support-vector network**. Machine Learning.

[26] Silicon Graphics International (SGI) Corp (2011)**Chapter 8. MineSet Inducers and Classifiers**. URL address = http://techpubs.sgi.com/library/tpl/cgi-bin/getdoc.cgi/0530/bks/SGI_EndUser/books/MineSet_UG/sgi_html/ch08.html Diakses pada tanggal 11 Desember 2012.