BAB II

LANDASAN TEORI

2.1 *Text Mining*

Text Mining adalah proses ekstraksi pola (informasi dan pengetahuan yang berguna) dari sejumlah besar sumber data tak terstruktur. Penambangan teks memiliki tujuan dan menggunakan proses yang sama dengan penambangan data, namun memiliki masukan yang berbeda. Masukan untuk penambangan teks adalah data yang tidak (atau kurang) terstruktur, seperti dokumen Word, PDF, kutipan teks, dll., sedangkan masukan untuk penambangan data adalah data yang terstruktur (Ronen Feldman, 2007). Penambangan teks dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini dengan menggunakan teknik dan alat yang sama dengan penambangan data.

Area penerapan penambangan teks *yang* paling populer adalah:

1. Ekstraksi informasi (*information extraction*): Identifikasi frasa kunci dan keterkaitan di dalam teks dengan melihat urutan tertentu melalui pencocokan pola.
2. Pelacakan topik (*topic tracking*): Penentuan dokumen lain yang menarik seorang pengguna berdasarkan profil dan dokumen yang dilihat pengguna tersebut.
3. Perangkuman (*summarization*): Pembuatan rangkuman dokumen untuk mengefisienkan proses membaca.
4. Kategorisasi (*categorization*): Penentuan tema utama suatu teks dan pengelompokan teks berdasarkan tema tersebut ke dalam kategori yang telah ditentukan.

5. Penggugusan (*clustering*): Pengelompokan dokumen yang serupa tanpa penentuan kategori sebelumnya (berbeda dengan kategorisasi di atas).
6. Penautan konsep (*concept linking*): Penautan dokumen terkait dengan identifikasi konsep yang dimiliki bersama sehingga membantu pengguna untuk menemukan informasi yang mungkin tidak akan ditemukan dengan hanya menggunakan metode pencarian tradisional.
7. Penjawaban pertanyaan (*question answering*): Pemberian jawaban terbaik

2.2 *Sentiment Analysis*

Sentiment Analysis atau opinion mining merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentiment yang terkandung dalam suatu kalimat opini. *Sentiment Analysis* dilakukan untuk melihat pendapat atau kecenderungan opini terhadap sebuah masalah atau objek oleh seseorang, apakah cenderung berpandangan atau beropini negatif atau positif (Bo Pang, 2002).

Sentiment Analysis dapat dibedakan berdasarkan sumber datanya, beberapa level yang sering digunakan dalam penelitian *Sentiment Analysis* adalah *Sentiment Analysis* pada level dokumen dan *Sentiment Analysis* pada level kalimat (Fink Clayton, 2011). Berdasarkan level sumber datanya *Sentiment Analysis* terbagi menjadi 2 kelompok besar yaitu :

1. *Coarse-grained Sentiment Analysis*
2. *Fined-grained Sentiment Analysis*

2.2.1 *Coarse-grained Sentiment Analysis*

Pada *Sentiment Analysis* jenis ini, *Sentiment Analysis* yang dilakukan adalah pada level dokumen. Secara garis besar fokus utama dari *Sentiment*

Analysis jenis ini adalah menganggap seluruh isi dokumen sebagai sebuah sentiment positif atau sentiment negatif (Fink Clayton, 2011). Salah satu contoh yang menggambarkan coarse-grained *Sentiment Analysis* adalah sebagai berikut :

> *" Time and time again, the wily filmmakers sprinkle the overarching storyline of the fall and decline of Larry Gopnik's life (a masterful, wide-ranging and sensitive performance from Michael Stuhlbarg) with a fine combination of overt, discreet and subliminal set-ups whose payoffs give their film extra punch and an unstoppable pace."* (Fink Clayton, 2011)

Pada review diatas, dapat disimpulkan dengan jelas bahwa sentiment yang terkandung dalam review tersebut adalah positif, dapat di jelaskan dengan kata-kata subjektif positif seperti "*masterful*" , "*extra punch*", dan "*unstopabble pace*".

2.2.2 *Fined-grained Sentiment Analysis*

Fined-grained Sentiment Analysis adalah *Sentiment Analysis* pada level kalimat. Fokus utama fined-greined *Sentiment Analysis* adalah menentukan sentiment pada setiap kaliamat pada suatu dokumen, dimana kemungkinan yang terjadi adalah terdapat sentiment pada level kalimat yang berbeda pada suatu dokumen (Fink Clayton, 2011).

2.3 Klasifikasi

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu : pertama, Pembangunan model sebagai *prototype* untuk disimpan sebagai memori dan kedua, Penggunaan model tersebut untuk melakukan pengenalan/ klasifikasi/

prediksi pada suatu objek data lain agar diketahui di kelas mana objek data tersebut dalam model yang mudah disimpan (Prasetyo, Eko. 2012).

Contoh aplikasi yang sering ditemui adalah pengklasifikasian jenis hewan, yang mempunyai sejumlah atribut. Dengan atribut tersebut, jika ada hewan baru, kelas hewannya bisa langsung diketahui. Contoh lain adalah bagaimana melakukan diagnosis penyakit kulit kanker melanoma (Prasetyo, Eko. 2012), yaitu dengan melakukan pembangunan model berdasarkan data latih yang ada, kemudian menggunakan model tersebut untuk mengidentifikasi penyakit pasien baru sehingga diketahui apakah pasien tersebut menderita kanker atau tidak.

2.4 *K-Nearest Neighbor*

Algoritma *k-nearest neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. KNN termasuk algoritma supervised learning dimana hasil dari query instance yang baru diklasifikan berdasarkan mayoritas dari kategori pada KNN. Nanti kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi.

Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan training sample. Classifier tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik query, akan ditemukan sejumlah k obyek atau (titik training) yang paling dekat dengan titik query. Klasifikasi menggunakan voting terbanyak diantara klasifikasi dari k obyek.. algoritma k-nearest neighbor (KNN) menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari query instanceyang baru.

Algoritma metode *K-Nearest Neighbor* (KNN) sangatlah sederhana, bekerja berdasarkan jarak terpendek dari query instance ke training

sample untuk menentukan KNN-nya. Training sample diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dari data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi training sample. Sebuah titik pada ruang ini ditandai kelas cjika kelas cmerupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat dari titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan Euclidean Distance.

Jarak Euclidean paling sering digunakan menghitung jarak. Jarak euclidean berfungsi menguji ukuran yang bisa digunakan sebagai interpretasi kedekatan jarak antara dua obyek. yang direpresentasikan sebagai berikut :

$$D(a,b) = \sqrt{\sum_{k=1}^{d}(a_k - b_k)^2}, \quad (2.1)$$

dimana matriks D(a,b) adalah jarak skalar dari kedua vektor adan b dari matriks dengan ukuran ddimensi.

Semakin besar nilai D akan semakin jauh tingkat keserupaan antara kedua individu dan sebaliknya jika nilai D semakin kecil maka akan semakin dekat tingkat keserupaan antar individu tersebut.

Nilai k yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai k yang tinggi akan mengurangi efek noise pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai k yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan cross-validation. Kasus khusus dimana klasifikasi diprediksikan berdasarkan training data yang paling dekat (dengan kata lain, k = 1) disebut algoritma nearest neighbor.

Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini

sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

Langkah-langkah untuk menghitung metode *K-Nearest Neighbor* :

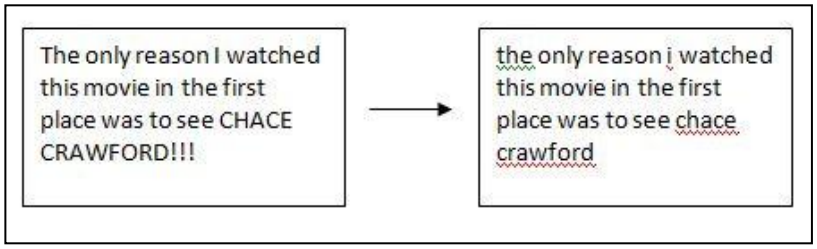
1. Menentukan parameter K (jumlah tetangga paling dekat).
2. Menghitung kuadrat jarak euclid (query instance) masing-masing obyek terhadap data sampel yang diberikan.
3. Kemudian mengurutkan objek-objek tersebut kedalam kelompok yang mempunyai jarak euclid terkecil.
4. Mengumpulkan kategori Y (Klasifikasi nearestneighbor)
5. Dengan menggunakan kategori nearest neighbor yang paling mayoritas maka dapat dipredisikan nilai query instance yang telah dihitung.

2.5 *Text Preprocessing*

Struktur data yang baik dapat memudahkan proses komputerisasi secara otomatis. Pada *Text Mining*, informasi yang akan digali berisi informasi-informasi yang strukturnya sembarang. Oleh karena itu, diperlukan proses pengubahan bentuk menjadi data yang terstruktur sesuai kebutuhannya untuk proses dalam data mining, yang biasanya akan menjadi nilai-nilai numerik. Proses ini sering disebut *Text Preprocessing* (Ronen Feldman, 2007). Setelah data menjadi data terstruktur dan berupa nilai numerik maka data dapat dijadikan sebagai sumber data yang dapat diolah lebih lanjut. Berberapa proses yang dilakukan adalah sebagai berikut :

1. Case folding

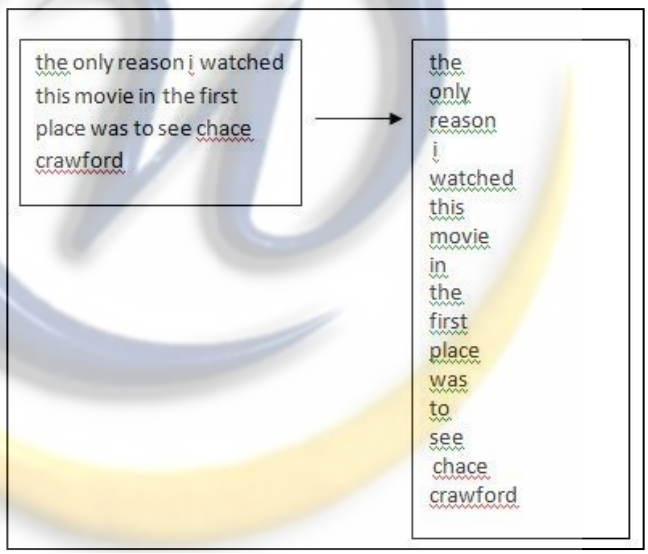
Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter (Ronen Feldman, 2007).



Gambar 2. 2 Proses case folding

2. Tokenizing

Tahap *Tokenizing* adalah tahap pemotongan *string input* berdasarkan tiap kata yang menyusunnya.



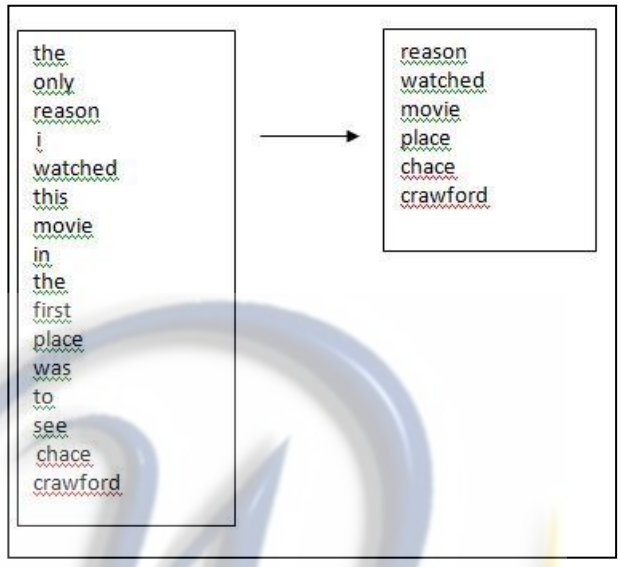
Gambar 2. 3Proses Tokenizing

2.5.1 *Text Transformation*

1. *Stopword removal* atau *filtering*

   Tahap *filtering* adalah tahap mengambil kata - kata penting dari hasil token. Bisa menggunakan algoritma stoplist (membuang kata yang kurang penting) atau wordlist (menyimpan kata penting). Stoplist / stopword adalah kata-kata yang tidak deskriptif yang

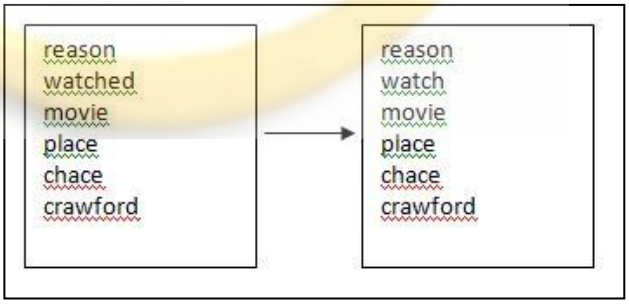
dapat dibuang dalam pendekatan bag-of-words (Ronen Feldman, 2007).



Gambar 2. 4 Proses stopword removal atau filtering

2. *Stemming*

Tahap *stemming adalah* tahap mencari root kata dari tiap kata hasil filtering. Pada tahap ini dilakukan proses pengembalian berbagai bentukan kata ke dalam suatu representasi yang sama.



Gambar 2. 5 Proses stemming

2.6 TF (*Term Frequency*)

Term frequency merupakan salah satu metode untuk menghitung bobot tiap *term* dalam teks. Dalam metode ini, tiap *term* diasumsikan memiliki nilai kepentingan yang sebanding dengan jumlah kemunculan *term* tersebut pada teks (Mark Hall & Lloyd Smith, 1999). Bobot sebuah *term* t pada sebuah teks dirumuskan dalam persamaan berikut :

$$W(d,t) = TF(d,t)$$

Dimana TF(d,t) adalah *term frequency* dari *term* t di teks d. *Term frequency* dapat memperbaiki nilai *recall* pada *information retrieval*, tetapi tidak selalu memperbaiki nilai *precision*. Hal ini disebabkan *term* yang *frequent* cenderung muncul di banyak teks, sehingga *term-term* tersebut memiliki kekuatan diskriminatif/keunikan yang kecil. Untuk memperbaiki permasalahan ini, *term* dengan nilai frekuensi yang tinggi sebaiknya dibuang dari *set term*. Menemukan *threshold* yang optimal merupakan fokus dari metode ini.

2.7 DF (*Document Frequency*)

Document frequency adalah jumlah dokumen yang mengandung suatu *term* tertentu. Tiap *term* akan dihitung nilai *document frequency*-nya (DF). Lalu *term* tersebut diseleksi berdasarkan jumlah nilai DF. Jika nilai DF berada di bawah *threshold* yang telah ditentukan, maka *term* akan dibuang. Asumsi awalnya adalah bahwa *term* yang lebih jarang muncul tidak memiliki pengaruh besar dalam proses pengelompokkan dokumen. Pembuangan *term* yang jarang ini dapat mengurangi dimensi fitur yang besar pada *text mining*.

Perbaikan dalam pengelompokkan dokumen ini juga dapat terjadi jika *term* yang dibuang tersebut juga merupakan *noise term*. *Document frequency* merupakan metode *feature selection* yang paling sederhana dengan waktu komputasi yang rendah (Yiming Yang & Jan O. Pedersen, 1997). Ilustrasi dari metode *document frequency* ini adalah sebagai berikut. Jika terdapat data berjumlah 5000 dokumen, dan jumlah dokumen yang mengandung *term* "teknologi" adalah 150 dokumen. Maka nilai DF(teknologi) adalah 150.