Twitter Sentiment Analysis

Abstrak

Dalam laporan ini, membahas masalah klasifikasi sentimen pada dataset twitter. menggunakan sejumlah metode pembelajaran mesin dan pembelajaran mendalam untuk melakukan analisis sentimen. Pada akhirnya, digunakan metode ensemble suara mayoritas dengan 5 model terbaik kami untuk mencapai akurasi klasifikasi 83,58% di papan peringkat publik kaggle. membandingkan berbagai metode yang berbeda untuk analisis sentimen pada tweet (masalah klasifikasi biner). Dataset pelatihan diharapkan menjadi file CSV dengan tipe tweet_id, sentimen, tweet di mana tweet_id adalah bilangan bulat unik yang mengidentifikasi tweet, sentimen adalah 1 (positif) atau 0 (negatif), dan tweet adalah tweet yang disertakan dalam "" . Demikian pula, dataset pengujian adalah file CSV dengan tipe tweet_id, tweet. Harap perhatikan bahwa header CSV tidak diharapkan dan harus dihapus dari set data pelatihan dan pengujian.

Pengantar

Analisis Sentimen Twitter berarti, menggunakan teknik penambangan teks canggih untuk menyelidiki sentimen teks (di sini, tweet) dalam jenis positif, negatif, dan netral. itu juga disebut Opinion Mining, terutama untuk menganalisis percakapan, opini, dan berbagi pandangan (semua dalam jenis tweet) untuk memutuskan strategi bisnis, analisis politik, dan juga untuk menilai tindakan publik. Analisis sentimen sering ingin mengidentifikasi tren dalam konten tweet, yang kemudian dianalisis dengan algoritma pembelajaran mesin.

Tinjauan Literatur

Sejumlah pekerjaan terkait sebelumnya yang layak telah dilakukan pada analisis sentimen ulasan pengguna, blog/artikel web, dan analisis sentimen tingkat frase, Ini berbeda dari Twitter terutama berkat batas 140 karakter per tweet yang memaksa pengguna untuk opini tertentu dikompresi dalam teks yang sangat sangat singkat. Hasil paling sederhana dicapai dalam klasifikasi sentimen menggunakan teknik pembelajaran terawasi seperti *Naive Bayes* dan *Support Vector Machines*, tetapi pelabelan manual yang diperlukan untuk pendekatan terawasi sangat mahal.

Berbagai peneliti sedang menguji fitur dan teknik klasifikasi baru. Dia sering membandingkan hasil mereka dengan kinerja dasar. Ada keinginan untuk mengoreksi dan Perbandingan formal antara hasil ini dibuat dengan fitur yang berbeda dan teknik klasifikasi untuk memilih fitur yang paling efektif dan paling efektif Teknik klasifikasi untuk aplikasi tertentu.

1. Pernyataan Masalah

Twitter adalah situs jejaring sosial populer di mana anggota membuat dan berinteraksi dengan pesan yang dikenal sebagai "tweet". Ini berfungsi sebagai sarana bagi individu untuk mengekspresikan pikiran atau perasaan mereka tentang subjek yang berbeda. Berbagai pihak seperti konsumen dan pemasar telah melakukan analisis sentimen pada tweet tersebut untuk mengumpulkan wawasan tentang produk atau untuk melakukan analisis pasar. Dalam laporan ini melakukan analisis sentimen pada “tweet” menggunakan berbagai algoritma pembelajaran mesin yang berbeda. Mengklasifikasikan polaritas tweet apakah positif atau negatif. Jika tweet memiliki elemen positif dan negatif, sentimen yang lebih dominan harus dipilih sebagai label

akhir. Menggunakan berbagai algoritma pembelajaran mesin untuk melakukan analisis sentimen menggunakan fitur yang diekstraksi. Ensembling adalah bentuk teknik algoritma meta learning di mana saya menggabungkan classier yang berbeda untuk meningkatkan akurasi prediksi.

2. Deskripsi Data

Data yang diberikan berupa file nilai yang dipisahkan koma dengan tweet dan sentimen terkaitnya. Dataset pelatihan adalah file csv dari tipe tweet_id,sentimen,tweet di mana tweet_id unik.

	Total	Unique	Average	Max	Positive	Negative
Tweets	800000	-	-	-	400312	399688
User Mentions	393392	-	0.4917	12	-	-
Emoticons	6797	-	0.0085	5	5807	990
URLs	38698	-	0.0484	5	-	-
Unigrams	9823554	181232	12.279	40	-	-
Bigrams	9025707	1954953	11.28	-	-	-

Table 1: Statistik dataset train yang telah diproses

	Total	Unique	Average	Max	Positive	Negative
Tweets	200000	-	-	-	-	-
User Mentions	97887	-	0.4894	11	-	-
Emoticons	1700	-	0.0085	10	1472	228
URLs	9553	-	0.0478	5	-	-
Unigrams	2457216	78282	12.286	36	-	-
Bigrams	2257751	686530	11.29	-	-	-

Table 2: Statistik dataset test yang telah diproses sebelumnya

dan emotikon berkontribusi untuk memprediksi sentimen, tetapi URL dan referensi ke orang tidak. Oleh karena itu, URL dan referensi dapat diabaikan. Kata-kata tersebut juga merupakan campuran dari kata-kata yang salah eja, tanda baca tambahan, dan kata-kata dengan banyak huruf yang berulang. Tweet, oleh karena itu, harus diproses terlebih dahulu untuk menstandarisasi kumpulan data.

3. Metodologi dan Implementasi

3.1 Pra-pemrosesan

Tweet mentah yang diambil dari twitter umumnya menghasilkan kumpulan *noisy dataset.* Tweet memiliki karakteristik khusus tertentu seperti retweet, emotikon, sebutan pengguna, dll. yang harus diekstraksi dengan tepat. Oleh karena itu, data mentah twitter harus dinormalisasi untuk membuat kumpulan data yang dapat dengan mudah dipelajari oleh berbagai pengklasifikasi. Kami telah menerapkan sejumlah besar langkah pra-pemrosesan untuk menstandardisasi kumpulan data dan mengurangi ukurannya. Kami pertama-tama melakukan beberapa pra-pemrosesan umum pada tweet yaitu sebagai berikut.

- Ubah tweet menjadi huruf kecil.

• Ganti 2 atau lebih titik (.) dengan spasi.

• Hapus spasi dan tanda kutip (" dan ') dari akhir tweet.

• Ganti 2 spasi atau lebih dengan satu spasi.

Kami menangani fitur twitter khusus sebagai berikut.

3.1.1 URL

Pengguna sering membagikan hyperlink ke halaman web lain di tweet mereka. URL tertentu tidak penting untuk klasifikasi teks karena akan menghasilkan fitur yang sangat jarang. Oleh karena itu, kami mengganti semua URL di tweet dengan kataURL. Ekspresi reguler yang digunakan untuk mencocokkan URL adalah ((www\.[\S]+)|(https?://[\S]+)).

3.1.2 Sebutan Pengguna

Setiap pengguna twitter memiliki pegangan yang terkait dengan mereka. Pengguna sering menyebut pengguna lain dalam tweet mereka dengan @menangani. Kami mengganti semua sebutan pengguna dengan kata USER_MENTION. Ekspresi reguler yang digunakan untuk mencocokkan penyebutan pengguna adalah @[\S]+.

Emoticon(s)	Type	Regex	Replacement
:), :), :-), (:, (:, (-:, :')	Smile	(:\s?\)|:-\)|\(\s?:|\(-:|:\'\))	EMO_POS
:D, : D, :-D, xD, x-D, XD, X-D	Laugh	(:\s?D|:-D|x-?D|X-?D)	EMO_POS
;-), ;), ;-D, ;D, (;, (-;	Wink	(:\s?\(|:-\(|\)\s?:|\)-:)	EMO_POS
<3, :*	Love	(<3|:*)	EMO_POS
:-(, : (, :(,):,)-:	Sad	(:\s?\(|:-\(|\)\s?:|\)-:)	EMO_NEG
:,(, :'(, :"(	Cry	(:,\(|:\'\(|:"\()	EMO_NEG

Table 3:Daftar emotikon yang cocok dengan metode ini

3.1.3 Emoticon

Pengguna sering menggunakan sejumlah emotikon yang berbeda dalam tweet mereka untuk menyampaikan emosi yang berbeda. Tidak mungkin untuk mencocokkan semua emotikon berbeda yang digunakan di media sosial secara menyeluruh karena jumlahnya terus meningkat. Namun, kami mencocokkan beberapa emotikon umum yang sangat sering digunakan. Kami mengganti emotikon yang cocok denganEMO_POS atau EMO_NEG tergantung pada apakah itu menyampaikan emosi positif atau negatif. Daftar semua emotikon yang cocok dengan metode kami diberikan dalam tabel3.

3.1.4 Tagar

Hashtag adalah frase tanpa spasi yang diawali dengan simbol hash (#) yang sering digunakan oleh pengguna untuk menyebutkan trending topic di twitter. Kami mengganti semua hashtag dengan kata-kata dengan simbol hash. Sebagai contoh, #Halo digantikan oleh Halo. Ekspresi reguler yang digunakan untuk mencocokkan tagar adalah #(\S+).

3.1.5 Retweet

Retweet adalah tweet yang telah dikirim oleh orang lain dan dibagikan oleh pengguna lain. Retweet dimulai dengan hurufRT. Kami menghapus RT dari tweet karena ini bukan fitur penting untuk klasifikasi teks. Ekspresi reguler yang digunakan untuk mencocokkan retweet

adalah \brt\b.Setelah menerapkan pra-pemrosesan tingkat tweet, kami memproses kata-kata individual dari tweet sebagai berikut.

- Hapus tanda baca ['"?!,.():;] dari kata.
- Ubah 2 atau lebih pengulangan huruf menjadi 2 huruf. Beberapa orang mengirim tweet sepertiSaya sangat senang menambahkan beberapa karakter untuk menekankan pada kata-kata tertentu. Ini dilakukan untuk menangani tweet semacam itu dengan mengonversinya menjadiSaya sangat senang.
- Hapus - dan '. Ini dilakukan untuk menangani kata-kata seperti t-shirt dan kata-katanya dengan mengubahnya menjadi bentuk yang lebih umum tshirt dan kata-kata mereka.
- Periksa apakah kata itu valid dan terima hanya jika itu benar. Kami mendefinisikan kata yang valid sebagai kata yang dimulai dengan alfabet dengan karakter berturut-turut menjadi alfabet, angka atau salah satu titik (.) dan garis bawah (_).

Beberapa contoh tweet dari dataset pelatihan dan versi normalnya ditunjukkan pada tabel 4.

3.2 Ekstraksi Fitur

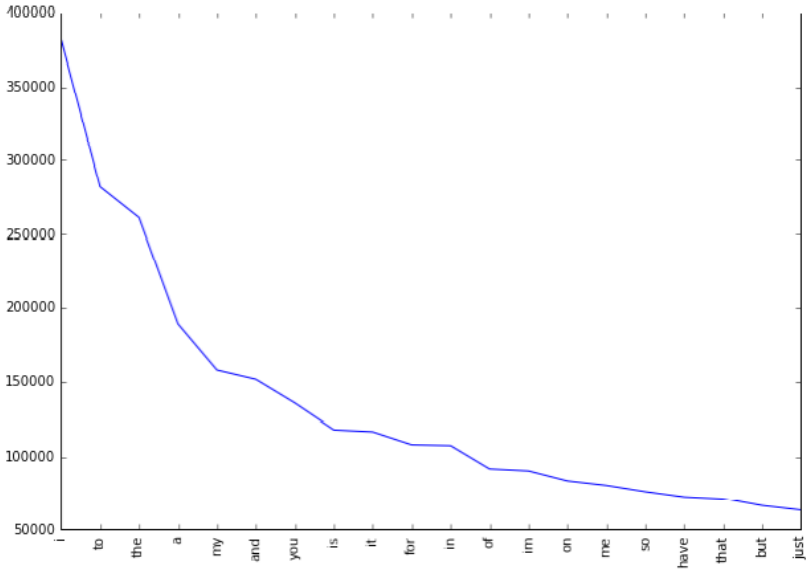
Kami mengekstrak dua jenis fitur dari dataset kami, yaitu unigram dan bigram. Kami membuat distribusi frekuensi unigram dan bigram yang ada di dataset dan memilih topn unigram dan bigram untuk analisis kami.

3.2.1 Unigram

Mungkin fitur yang paling sederhana dan paling umum digunakan untuk klasifikasi teks adalah adanya satu kata atau token dalam teks. Kami mengekstrak kata-kata tunggal dari dataset pelatihan dan membuat distribusi frekuensi kata-kata ini. Sebanyak 181232 kata unik diambil dari

Raw	misses Swimming Class. http://plurk.com/p/12nt0b
Normalized	misses swimming class URL
Raw	@98PXYRochester HEYYYYYYYYY!! its Fer from Chile again
Normalized	USER_MENTION heyy its fer from chile again
Raw	Sometimes, You gotta hate #Windows updates.
Normalized	sometimes you gotta hate windows updates
Raw	@Santiago_Steph hii come talk to me i got candy :)
Normalized	USER_MENTION hii come talk to me i got candy EMO_POS
Raw	@bolly47 oh no :'(r.i.p. your bella
Normalized	USER_MENTION oh no EMO_NEG r.i.p your bella

Table 4: Contoh tweet dari dataset dan versi normal



Gambar 1: Frekuensi 20 unigram teratas

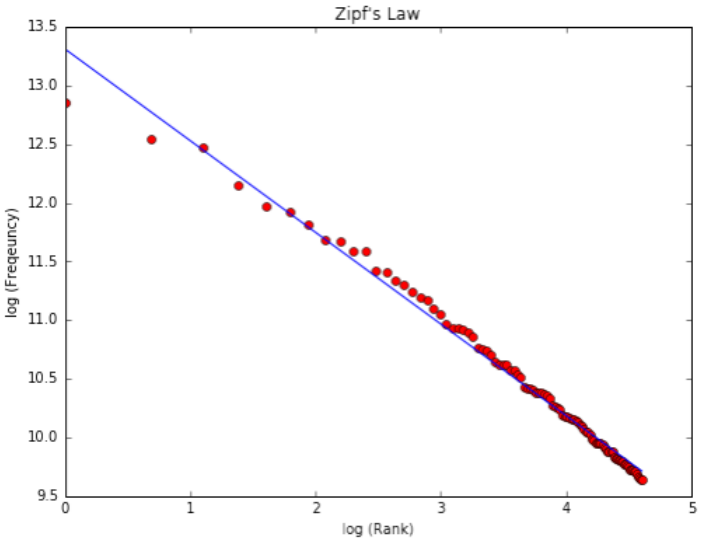
kumpulan datanya. Dari kata-kata ini, sebagian besar kata-kata di ujung spektrum frekuensi adalah *noise* dan muncul sangat sedikit untuk mempengaruhi klasifikasi. Oleh karena itu, kami hanya menggunakan topn katakata dari ini untuk membuat kosakata kami di mana n adalah 15000 untuk klasifikasi vektor jarang dan 90000 untuk klasifikasi vektor padat. Distribusi frekuensi 20 kata teratas dalam kosakata kami ditunjukkan pada gambar 1.

3.2.2 Bigram

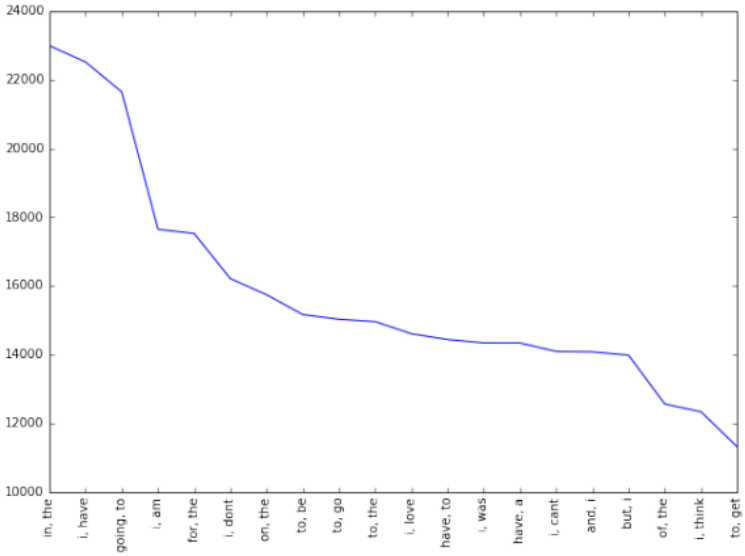
Bigrams adalah pasangan kata dalam dataset yang terjadi berturut-turut di corpus. Fitur-fitur ini adalah cara yang baik untuk memodelkan negasi dalam bahasa alami seperti dalam frasa – Ini tidak bagus. Sebanyak 1954953 bigram unik diekstraksi dari dataset. Dari jumlah tersebut, sebagian besar bigram di ujung spektrum frekuensi adalah noise dan terjadi sangat sedikit untuk mempengaruhi klasifikasi. Oleh karena itu hanya menggunakan 10.000 bigram teratas dari ini untuk membuat kosa kata kami.

3.3 Representasi Fitur

Setelah mengekstrak unigram dan bigram, kami mewakili setiap tweet sebagai vektor fitur baik dalam representasi vektor jarang atau vektor padat tergantung pada metode klasifikasi.



Gambar 2 Frekuensi Unigram mengikuti Hukum Zipf.



Gambar 3 Frekuensi 20 bigram teratas.

3.3.1 Representasi *Sparse Vector*

Bergantung pada apakah kita menggunakan fitur bigram atau tidak, representasi vektor jarang dari setiap tweet memiliki panjang 15000 (bila mempertimbangkan hanya unigram) atau 25000 (bila mempertimbangkan unigram dan bigram). Nilai positif pada indeks unigram (dan bigram) tergantung pada jenis fitur yang kami tentukan yang merupakan salah satu dari kehadiran dan frekuensi.

- ***presence*** Dalam kasus kehadiran tipe fitur, vektor fitur memiliki 1 pada indeks unigram (dan bigram) yang ada dalam tweet dan 0 di tempat lain.
- ***frequency*** Dalam kasus frekuensi tipe fitur, vektor fitur memiliki bilangan bulat positif pada indeks unigram (dan bigram) yang merupakan frekuensi unigram (atau bigram)

itu di tweet dan 0 di tempat lain. Matriks vektor term-frekuensi tersebut dibangun untuk seluruh dataset pelatihan dan kemudian setiap frekuensi term diskalakan dengan frekuensi dokumen terbalik dari term (idf) untuk menetapkan nilai yang lebih tinggi ke term penting. Frekuensi dokumen terbalik dari suatu istilah T didefinisikan sebagai.

$$idf(t) = \log\left(\frac{1+n_d}{1+df(d,t)}\right)+1$$

di mana *nd* adalah jumlah total dokumen dan *df(d,t)* adalah jumlah dokumen di mana istilah t terjadi.

3.3.2 Representasi Vektor Padat

Untuk representasi vektor padat, kami menggunakan kosakata unigram berukuran 90.000 yaitu 90.000 kata teratas dalam kumpulan data. Kami menetapkan indeks bilangan bulat untuk setiap kata tergantung pada peringkatnya (mulai dari 1) yang berarti bahwa kata yang paling umum diberi nomor 1, kata paling umum kedua diberi nomor 2 dan seterusnya. Setiap tweet kemudian diwakili oleh vektor dari indeks ini yang merupakan vektor padat.

3.4 PengklasifikasI

3.4.1 Naive Bayes

Naive Bayes adalah model sederhana yang dapat digunakan untuk klasifikasi teks. Dalam model ini, kelas ditugaskan ke tweet T, di mana

$$\hat{c} = \underset{c}{argmax}\ \mathrm{P(c|t)}$$

$$\mathrm{P(c|t)} \propto \mathrm{P(c)}\prod_{i=1}^{n}\mathrm{P}(f_i|c)$$

Dalam rumus di atas, FSaya mewakili Saya-fitur dari total n fitur. P(c) dan P(FSaya|c) dapat diperoleh melalui estimasi kemungkinan maksimum.

3.4.2 Maximum Entropy

Model Pengklasifikasi Entropi Maksimum didasarkan pada Prinsip Entropi Maksimum. Gagasan utama di baliknya adalah untuk memilih model probabilistik paling seragam yang memaksimalkan entropi, dengan batasan yang diberikan.

$$P_{ME}(c|d,\lambda) = \frac{exp[\sum_i \lambda_i f_i(c,d)]}{\sum_{c'} exp[\sum_i \lambda_i f_i(c,d)]}$$

3.4.3 Decision Tree

Pohon keputusan adalah model pengklasifikasi di mana setiap simpul pohon mewakili pengujian pada atribut kumpulan data, dan anak-anaknya mewakili hasil. Node daun mewakili kelas akhir dari titik data. Ini adalah model pengklasifikasi terawasi yang menggunakan data dengan label yang diketahui untuk membentuk pohon keputusan dan kemudian model tersebut diterapkan pada data uji. Untuk setiap simpul di pohon, kondisi pengujian atau keputusan terbaik.

3.4.4 Random Forest

Random Forest adalah algoritma pembelajaran ensemble untuk klasifikasi dan regresi. Random Forest

menghasilkan banyak pohon keputusan yang diklasifikasikan berdasarkan keputusan agregat dari pohon-pohon tersebut. Untuk satu set tweet x1, x2, . . . xn dan label sentimen masing-masing kamu1, kamu2, . . .n mengantongi berulang kali memilih sampel acak (xB, kamuB) dengan penggantian.

3.4.5 XGBoost

Xgboost adalah bentuk algoritma peningkatan gradien yang menghasilkan model prediksi yang merupakan kumpulan pohon keputusan prediksi yang lemah. Kami menggunakan ansambel model K dengan menambahkan outputnya dengan cara berikut:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F$$

di mana F adalah ruang pohon, xSaya adalah masukan danSaya merupakan keluaran akhir.

$$L(\Phi) = \sum_i l(\hat{y}_i, y_i) + \sum \Omega(f_k)$$

$$where\ \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

di mana Ω.adalah istilah regularisasi

3.4.6 SVM

SVM, juga dikenal sebagai mesin vektor pendukung, adalah pengklasifikasi linier biner non-probabilistik.hyperplane margin maksimum yang membagi titik dengankamuSaya =1 dan kamuSaya =1. Persamaan hyperplane adalah sebagai berikut: w·x b =0.

3.4.7 Perceptron Multi-Lapisan

MLP atau Multilayer perceptron adalah kelas jaringan saraf feed-forward, yang memiliki setidaknya tiga lapisan neuron. Setiap neuron menggunakan fungsi aktivasi non-linier, dan belajar dengan pengawasan menggunakan algoritma backpropagation. Ini berkinerja baik dalam masalah klasifikasi yang kompleks seperti analisis sentimen dengan mempelajari model non-linier.

3.4.8 Convolutional Neural Networks

Convolutional Neural Networks atau CNN adalah jenis jaringan saraf yang melibatkan lapisan yang disebut lapisan konvolusi yang dapat menginterpretasikan data spasial. Lapisan konvolusi memiliki sejumlah filter atau kernel yang dipelajari untuk mengekstrak jenis fitur tertentu dari data. Kernel adalah jendela 2D yang digeser di atas data input yang melakukan operasi konvolusi. Kami menggunakan konvolusi temporal dalam eksperimen kami yang cocok untuk menganalisis data sekuensial seperti tweet.

3.4.9 Multi-Layer Perceptron

Recurrent Neural Network adalah jaringan node seperti neuron, masing-masing dengan koneksi terarah (satu arah) ke setiap node lainnya. Dalam RNN, keadaan tersembunyi dilambangkan denganHT bertindak sebagai

memori jaringan dan mempelajari informasi kontekstual yang penting untuk klasifikasi bahasa alami. Output pada setiap langkah dihitung berdasarkan memoriHT pada waktu T dan masukan saat ini xT. Fitur utama dari RNN adalah keadaan tersembunyinya, yang menangkap ketergantungan berurutan dalam informasi. Kami menggunakan jaringan Long Term Short Memory (LSTM) dalam percobaan kami yang merupakan jenis khusus dari RNN yang mampu mengingat informasi dalam jangka waktu yang lama.

4. Eksperimen

Menggunakan 10% dari dataset pelatihan untuk validasi model kami untuk memeriksa terhadap overfitting yaitu menggunakan 720000 tweet untuk pelatihan dan 80000 tweet untuk validasi. Untuk Naive Bayes, Entropi Maksimum, Pohon Keputusan, Hutan Acak, XGBoost, SVM, dan Perceptron Multi-Layer, kami menggunakan representasi vektor sparse dari tweet. Untuk *Recurrent Neural Network dan Convolutional Neural Network* menggunakan *the dense vector representation*.

4.1 Dasar

Untuk garis dasar, kami menggunakan metode penghitungan kata positif dan negatif sederhana untuk menetapkan sentimen ke tweet tertentu. Kami menggunakan Kumpulan Data Opini kata positif dan negatif untuk mengklasifikasikan tweet. Dalam kasus ketika jumlah kata positif dan negatif sama, kami menetapkan sentimen positif. Dengan menggunakan model dasar ini, kami mencapai akurasi klasifikasi 63,48% di papan peringkat publik Kaggle.

4.2 Naif Bayes

Kami menggunakan MultinomialNB dari sklearn.naive_bayes paket dari scikit-belajar untuk klasifikasi Naive Bayes. Kami menggunakan Naive Bayes versi pemulusan Laplace dengan parameter pemulusan disetel ke nilai defaultnya 1. Kami menggunakan representasi vektor sparse untuk klasifikasi dan menjalankan eksperimen menggunakan keduanyakehadiran dan frekuensi jenis fitur. Kehadiran fitur mengungguli frekuensi fitur karena Naive Bayes pada dasarnya dibangun untuk bekerja lebih baik pada fitur integer daripada *floats*. Memperoleh akurasi validasi terbaik sebesar 79,68% menggunakan Naive Bayes dengankehadiran dari unigram dan bigram.

4.3 Maximum Entropy

Nltk *library* menyediakan beberapa alat analisis teks. Menggunakan Pengklasifikasi Maksent untuk melakukan analisis sentimen pada tweet yang diberikan. Unigram, bigram, dan kombinasi keduanya diberikan sebagai fitur input ke classifier. Algoritme Penskalaan Iteratif yang Ditingkatkan untuk pelatihan memberikan hasil yang lebih baik daripada Penskalaan Iteratif Umum. Kombinasi fitur unigram dan bigram, memberikan akurasi yang lebih baik sebesar 80,98% dibandingkan hanya unigram (79,34%) dan hanya bigram (79,2%).

4.4 Decision Tree

menggunakan Pengklasifikasi Pohon Keputusan dari sklearn.tree paket yang disediakan oleh scikit-belajar untuk membangun model kami. GINI digunakan untuk mengevaluasi split pada setiap node dan split terbaik selalu dipilih. Model tampil sedikit lebih baik menggunakan fitur kehadiran dibandingkan dengan frekuensi. Juga menggunakan unigram dengan atau tanpa bigram tidak membuat peningkatan yang signifikan. Akurasi terbaik yang dicapai dengan menggunakan pohon keputusan adalah 68,1%.

4.5 Random Forest

Kami menerapkan algoritma hutan acak dengan menggunakan RandomForestClassifier dari sklearn. Ensemble disediakan oleh scikit-belajar. Kami bereksperimen menggunakan 10 estimator (pohon) menggunakan keduanya kehadiran dan frekuensi fitur. kehadiran fitur berkinerja lebih baik daripada frekuensi meskipun peningkatannya tidak signifikan.

4.6 XGBoost

Set maksimum tree ke 25 di mana ini mengacu pada Set maksimum tree dan digunakan untuk mengontrol over-fitting karena nilai yang tinggi dapat mengakibatkan hubungan pembelajaran model yang terkait dengan data pelatihan. Karena XGboost adalah algoritme yang menggunakan kumpulan pohon yang lebih lemah, penting untuk menyesuaikan jumlah penduga yang digunakan. Kami menyadari bahwa menyetel nilai ini ke 400 memberikan hasil terbaik. Hasil terbaik adalah 0.78.72 yang berasal dari konfigurasi kehadiran dengan Unigram + Bigrams.

4.7 SVM

Kami menggunakan pengklasifikasi SVM yang tersedia di sklearn. Menetapkan istilah C menjadi 0,1. Istilah C adalah parameter penalti dari istilah kesalahan. Mempengaruhi kesalahan klasifikasi pada fungsi tujuan. Kami menjalankan SVM dengan baik Unigram maupun Unigram + Bigram. Kami juga menjalankan konfigurasi dengan frekuensi dan kehadiran. Hasil terbaik adalah 81,55 yang datang pada konfigurasi frekuensi dan Unigram + Bigram.

LSTM Units	Dense Units	`max_length`	Loss	Embedding Initialization	Accuracy
100	32	40	MSE	Random	79.8%
100	32	40	BCE	Random	82.2%
50	32	40	MSE	Random	78.96%
50	32	40	BCE	Random	81.97%
100	600	20	BCE	GloVe	82.7%
128	64	40	BCE	GloVe	83.0%

Table 5 Perbandingan Model LSTM. MSE adalah mean squared error dan BCE adalah binary cross entropy.

4.8 Multi-Layer Perceptron

Kami menggunakan keras dengan TensorFlow backend untuk mengimplementasikan model Multi-Layer Perceptron. Kami menggunakan Neural Network 1-hidden dengan 500 unit tersembunyi. Keluaran dari neural network adalah nilai tunggal yang kita lewati non-linier sigmoid untuk menekannya dalam kisaran [0, 1].

Algorithms	Presence		Frequency	
	Unigrams	Unigrams+Bigrams	Unigrams	Unigrams+Bigrams
Naive Bayes	78.16	**79.68**	77.52	79.38
Max Entropy	79.96	**81.52**	79.7	81.5
Decision Tree	**68.1**	68.01	67.82	67.78
Random Forest	76.54	**77.21**	76.16	77.14
XGBoost	77.56	**78.72**	77.42	78.32
SVM	79.54	81.11	79.83	**81.55**
MLP	80.1	**81.7**	80.15	81.35

Table 6 Perbandingan klasifikasi yang menggunakan sparse vector representation.

4.9 Convolutional Neural Networks

Menggunakan Keras dengan TensorFlow backend untuk mengimplementasikan model Convolutional Neural Network. Kami menggunakan representasi vektor padat dari tweet untuk melatih model CNN. Menggunakan kosakata 90000 kata teratas dari *training dataset*.

4.10 Recurrent Neural Networks

Menggunakan Neural Networks dengan lapisan LSTM dalam percobaan kami. Kami menggunakan kosakata 20000 kata teratas dari dataset pelatihan. Kami menggunakan representasi vektor padat untuk melatih model kami. Kami mengisi atau memotong setiap representasi vektor padat untuk membuatnya sama denganpanjang maksimal yang merupakan parameter yang kami ubah dalam eksperimen kami.

4.11 Ansambel

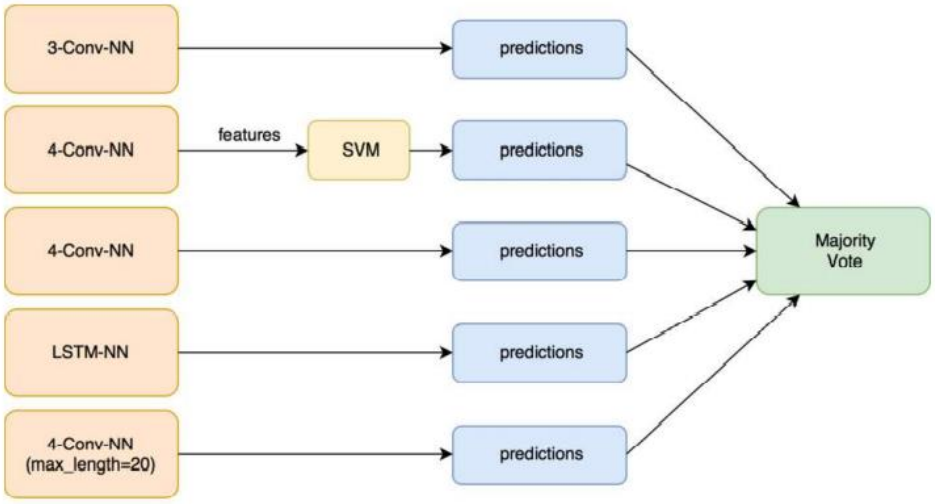
Dalam upaya untuk lebih meningkatkan akurasi, kami mengembangkan model ensemble sederhana. Kami pertamatama mengekstrak vektor fitur 600 dimensi untuk setiap tweet dari lapisan kedua dari belakang dengan kinerja terbaik kami 4-Konv-NN model. Setiap tweet sekarang diwakili oleh vektor fitur 600 dimensi. Kami menggunakan fitur ini untuk mengklasifikasikan tweet menggunakan model SVM linier dengan C=1. Kami mengklasifikasikan tweet menggunakan model SVM ini. Kami kemudian mengambil suara mayoritas prediksi dari 5 model berikut.

1. LSTM-NN
2. 4-Konv-NN
3. 4-Konv-NN fitur + SVM
4. 4-Konv-NN dengan panjang_maks = 20
5. 3-Konv-NN

5. Kesimpulan

5.1 Ringkasan pencapaian

Tweet yang disediakan merupakan gabungan kata, emoticon, URL, hastag, user mention, dan simbol. Sebelum training, kami melakukan pra-proses tweet agar cocok untuk dimasukkan ke dalam model.

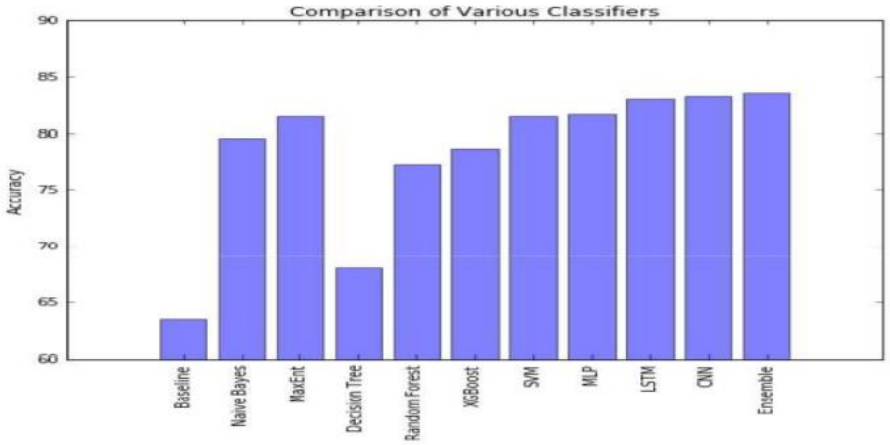


Gambar 4 Flowchart dari Mayoritas voting ensemble

	Accuracy
LSTM-NN	83.00
4-Conv-NN	83.34
4-Conv-NN features + SVM	83.39
4-Conv-NN with max_length = 20	82.85
3-Conv-NN	82.95
Majority Vote Ensemble	83.58

Table 7Models digunakan untuk ensemble dan akurasi pada peringkat publik kaggle

Menggunakan dua jenis fitur yaitu unigram dan bigram untuk klasifikasi dan mengamati bahwa menambah vektor fitur dengan bigram meningkatkan akurasi. Setelah fitur diekstraksi, fitur tersebut direpresentasikan sebagai vektor jarang atau vektor padat. Telah diamati bahwa kehadiran dalam representasi vektor jarang mencatat kinerja yang lebih baik daripada frekuensi.



Gambar 5 Perbandingan dari akurasi dari various model

Diskusi dan Hasil

Memberikan hasil untuk analisis sentimen di Twitter. Model unigram yang dikembangkan sebelumnya diusulkan sebagai dasar kami dan kami melaporkan keuntungan keseluruhan untuk dua tugas pemeringkatan: biner, positif versus negatif, dan tiga kali lipat positif versus negatif versus netral. kami menyediakan serangkaian eksperimen yang komprehensif untuk masing-masing dari dua tugas ini pada data beranotasi manual yang merupakan sampel acak dari tweet. kami melihat dua jenis model: kernel pohon dan model berbasis fitur dan menunjukkan bahwa kedua model mengungguli baseline Unigram.

Untuk pendekatan berbasis fitur kami, kami menganalisis fitur yang mengungkapkan bahwa fitur terpenting adalah fitur yang menggabungkan pra-polaritas kata dengan tanda bagian ucapannya. kami menyimpulkan pada awalnya bahwa analisis sentimen data Twitter tidak jauh berbeda dengan analisis sentimen jenis lainnya. Dalam pekerjaan masa depan, kita akan mengeksplorasi analisis linguistik yang lebih kaya, misalnya, parsing, analisis semantik, dan pemodelan subjek.

Menganalisis Tesis Positif VS Negatif. Itu adalah tugas klasifikasi biner dengan dua kelas polaritas sentimen: positif dan negatif. Menggunakan set data seimbang dari 1709 instans untuk setiap kelas dan oleh karena itu peluang dasar adalah 50%