Journal of Scientech Research and Development
Volume 5, Issue 2, December 2023
P-ISSN: 2715-6974 E-ISSN: 2715-5846
Open Access at: http://idm.or.id/JSCR

SEMEVAL 2017 ASSIGNMENT 4: SENTIMENT ANALYSIS ON TWITTER

SEMEVAL 2017 TUGAS 4: ANALISIS SENTIMEN DI TWITTER

Brian Arnesto Sitorus[1], Zakiul Fahmi Jailani[2], Dita Nurmadewi[3]
[1,2,3] Program Studi Sistem Informasi Fakultas Teknik dan Ilmu Komputer Universitas Bakrie
E-mail: brian.sitorus@bakrie.ac.id[1], zakiul.jailani@bakrie.ac.id[2], dita.nurmadewi@bakrie.ac.id[3]

ARTICLE INFO

Correspondent:

Brian Arnesto Sitorus
brian.sitorus@bakrie.ac.id

Key words:
sentiment, twitter

Website:
https://idm.or.id/JSCR/index.php/JSCR

Page: **1081 - 1096**

ABSTRACT

The dataset utilized in this study, SemEval, comprises 11 sets of tweet datasets from the Twitter platform collected between 2013 and 2016. The obtained dataset requires several preprocessing steps to address errors, such as tweets separated by tabs and commas, resulting in overlapping tweets within a dataset. Due to two datasets having excessive performance errors, only 9 sets of datasets are used in this study. During the preprocessing process, dataset errors are analyzed using the Spacy library. Subsequently, @mentions referring to usernames mentioned in the tweets are removed, lemmatization is performed using Spacy, and characters not conforming to standard spelling are deleted. The dataset consists of three classes: neutral, positive, and negative, but these classes have disproportionate numbers. When the dataset proportions are imbalanced, the training process may produce machine learning models biased towards the most predominant class. To address this bias issue, oversampling and undersampling techniques are applied. When implementing these techniques, the Synthetic Minority Over-sampling Technique (SMOTE) exhibits the best performance compared to other methods. Furthermore, various classifiers have been tested concurrently with SMOTE, and Logistic Regression demonstrates the most superior performance.

INFO ARTIKEL

Koresponden

Brian Arnesto Sitorus
brian.sitorus@bakrie.ac.id

Kata kunci:
sentimen, twitter

Website:
https://idm.or.id/JSCR/index.php/JSCR

Hal: 1081 - 1096

ABSTRAK

Dataset SemEval yang digunakan dalam penelitian ini mencakup 11 set dataset tweet dari platform twitter yang dikumpulkan antara tahun 2013 hingga 2016. Dataset yang didapatkan masih memerlukan beberapa proses preprocessing agar kesalahan dalam dataset tersebut dapat teratasi seperti adanya tweet yang dipisahkan mengggunakan tab dan koma, sehingga dalam satu dataset dapat memuat beberapa tweet yang saling bertumpuk. Dikarenakan ada 2 dataset yang memiliki terlalu banyak kesalahan performattan, hanya 9 set dataset yang digunakan dalam penelitian ini. Pada proses praprocessing, kesalahan dalam dataset dianalisis menggunakan library Spacy, selanjutnya tanda @mention yang merujuk kepada username yang dimention dalam tweet tersebut dihapus, lemmatisasi dilakukan dengan menggunakan Spacy, serta karakter yang tidak sesuai dengan ejaan standar dihapuskan. Terdapat tiga kelas diadalam dataset tersebut yaitu neutral, positive dan negative, namun antara ketiga kelas ini memiliki proporsi jumlah yang tidak seimbang. Ketika proporsi dataset tidak seimbang, pada proses training akan menghasilkan model machine learning yang bias pada kelas set yang paling mayoritas. Untuk mengatasi kendala bias ini , maka teknik oversampling dan undersampling diterapkan. Ketika mengimplementasikan kedua teknik ini, metode SMOTE dari teknik oversampling memiliki performa yang terbaik dibandingkan metode lainnya. Selanjutnya, beragam classifier telah diuji bersamaan dengan SMOTE, dan Logistic Regression menunjukkan performa yang paling superior.

PENDAHULUAN

Ada sekitar 186 orang yang secara aktif menggunakan twitter setiap hari, dan setidaknya ada 500 juta tweet yang diposting setiap hari. Para pengguna twitter saling berkirim pesan singkat yang disebut dengan "tweet" pada media sosial ini. Para pengguna memposting tentang pandangan mereka dan perasaan mereka melalui postingan ini. Walau postingan di twitter hanya dibatasi 140 karakter, namun postingan ini memuat bermacam-macam informasi, dari info komersial sampai dengan konten politik [15, 39, 40].

Website Micro-blogging seperti twitter yang didiskusikan sebelumnya menjadi sangat populer belakangan ini sebagai media untuk menyebarkan dan mendapatkan informasi. Media sosial ini menjadi cukup populer di kalangan peneliti untuk penelitian tentang sentimen analisis. Mengingat bahwa siapa saja bisa memposting di twitter, kualitas data dan masih menjadi tantangan mengingat bahwa banyak sekali pengguna yang tidak memposting tweet mereka dalam kalimat yang baku dan dengan ejaan yang tepat. Dengan meningkatnya penetrasi internet di dunia begitu juga dengan penggunaan sosial media seperti twitter yang juga semakin meningkat, yang di mana penggunaan twitter ini memproduksi dataset dalam bentuk text dalam

jumlah yang cukup besar dalam bentuk tweet yang dapat dianalisis menggunakan metode sentimen analisis. Sebagai dampaknya, sekarang kita memiliki data dalam jumlah yang sangat besar di mana sampai kita tidak dapat mengecek dataset tweet tersebut secara satu persatu, atau merangkum dan menyusun dataset tersebut dikarenakan jumlahnya yang sudah sangat masive.

Selain itu, sering sekali para pengguna twitter menggunakan bentuk tidak baku dari sebuah kata, yang mana ini dapat mengakibatkan miisinterpretasi dari konteks kalimat tersebut. Bahasa informal mencakup penggunaan bahasa sehari-hari dan bahasa gaul dalam komunikasi, serta standar bahasa singkatan seperti 'wouldn't' [10]. Proses analisis dan pengambilan keputusan dapat terhambat karena tidak semua classifier machine learning sudah di training dan mengenali makna emosi dalam kalimat informal.

Para pengguna twitter juga menggunakan emotikon untuk berinteraksi satu dengan yang lain. Dikarenakan tidak memungkinakan untuk menggunakan bahasa tubuh, emotikon menjadi alternatif para pengguna twittter untuk mengekspreksikan diri mereka sehingga penerima pesan memiliki persepsi yang lebih baik terhadap tanggapan pengirim pesan atau lawan interaksi mereka [10, 12]. Sebagai contoh, emotikon menunjukkan kondisi mental yang ceria. Saat ini, algoritma ataupun machine learning atau deep learning model yang tersedia tidak memiliki data yang cukup untuk memahami dan menafsirkan emotikon tertentu. Penggunaan singkatan dan emotikon juga banyak digunakan dalam layanan pesan singkat seperti SMS untuk menghemat penggunaan karakter teks pada pesan singakt yang dikirimkan. Dikarenakan Twitter memiliki batasan karakter sejumlah 140 karater, bentuk singkat lebih sering digunakan [6]. Misalnya, frasa "to be announced" disingkat menjadi "Tba".

SemEval 2017 adalah tempat yang baik bagi para peneliti untuk meneliti tentang analisis sentimen di Twitter [33]. Penelitian ini berfokus pada Tugas pada point 4: Analisis Sentimen di Twitter Subtugas A: Klasifikasi Polaritas Pesan, eksperimen dengan berbagai pengklasifikasi yang berbeda dan analisis kinerjanya. Dengan menggunakan skala tiga kelas, penelitian ini mencoba untuk menentukan polarisasi tweet secara keseluruhan yaitu: Positif, Negatif, dan Netral.

Ketika melakukan klasifikasi sentimen pada twit, tugas tersebut dapat dianggap sebagai analisis sentimen pada tingkat kalimat. Hal ini disebabkan oleh batasan karakter pada twit, yang membuat teksnya lebih pendek dan kurang kompleks dibandingkan dengan teks pada tingkat dokumen atau paragraf. Oleh karena itu, fokus analisis sentimen pada twit cenderung lebih pada setiap kalimat secara terpisah, daripada pada keseluruhan teks. Batasan karakter yang terbatas pada twit mempengaruhi pendekatan dan fitur-fitur yang digunakan dalam tugas klasifikasi sentimen tersebut. Dalam penelitian ini [20, 41, 42], mereka menggunakan berbagai fitur pemrosesan bahasa alami tradisional. Fitur-fitur ini mencakup fitur linguistik (n-gram kata, POS tags, dan lain-lain), fitur leksikon sentimen (skor berdasarkan delapan leksikon sentimen), dan fitur konten domain (misalnya, jumlah kata yang memiliki jumlah POS tags yang sama) (misalnya, emotikon, kata-kata kapital, kata-kata yang diperpanjang, dll.).

Tweet itu mengandung metadata yang cukup besar, penelitian ini mencoba untuk mengeksplor fitur apapun yang mungkin ada didalam metadata tesebut. Untuk mengeksplor fitur yang ada didalam dataset tersebut metode seperti word emberding diterapkan. Metode ini digunakan baik seluruh kalimat yang ada didalam dataset

tersebut. Eksperimen ini dilakukan untuk mengobservasi bagaimana setiap fitur dan algoritma machine learning dengan metode supervised learning bekerja.

METODE

Text Representation

Untuk bagi experiment, penelitian ini menggunakan representasi teks dengan menggunakan tf-idf (term frequency inverse document frequency). Bobot yang ada pada metode tf-idf adalah skema pembobotan yang sering digunakan dalam pencarian informasi dan penggalian teks. Bobot ini digunakan untuk menentukan kegunaan kata-kata tertentu dalam sebuah koleksi atau korpus. Semakin sering sebuah kata muncul dalam sebuah dokumen, semakin penting kata tersebut. Di sisi lain, kemunculan kata tersebut di dalam korpus akan mengurangi signifikansinya. Menurut mesin pencari, algoritma pembobotan TF-IDF sering digunakan untuk menentukan peringkat dan menganalisis relevansi situs dengan kueri pengguna tertentu. Metode ini dapat digunakan untuk memastikan frekuensi kemunculan frasa dokumen tertentu [1]. Kami menggunakan implementasi Scikit-learn dari tf-idf untuk mengubah data mentah menjadi matriks fitur dari metode tf-idf.

Classifiers

Kami menggunakan implementasi Scikit-learn dari beberapa machine learning classifier.

1. ***Logistic Regression Classifier***

   Logistic Regression adalah pengklasifikasi linier untuk tugas klasifikasi biner yang memprediksi probabilitas kelas. Regresi ini menggunakan fungsi sigmoid untuk menghitung probabilitas. Ia menggunakan pendekatan satu-vs-semua dalam kasus masalah multikelas. Mode Lienar Regression akan menghasilkan output nilai biner, seperti benar/salah, ya/tidak, atau yang serupa dengan itu. Meskipun begitu, logistic regession juga dapat digunakan untuk menyelesaikan masalah klafisikasi yang multi class. Logistic Regression dapat digunakan untuk melihat apakah sebuah sampel data point yang diuji termasuk kedalam kategori yang sudah ditentukan sebelumnya [11].

2. ***Random Forest Classifier*.**

   Random forest, atau disebut juga sebagai pohon keputusan acak, adalah teknik pembelajaran ensembel untuk tugas klasifikasi, regresi, dan tugas-tugas lain yang memerlukan proses training pohon keputusan dalam jumlah yang cukup besar yang mana kemudian menghasilkan kelas yang merupakan modus atau prediksi rata-rata dari pohon-pohon keputusan tersebut tersebut (regresi) [2]. Random forest digunakan untuk tugas klasifikasi dan regresi. Mpdel dibangun dari beberapa pohon keputusan pada sampel yang berbeda dan dibutuhkan mayoritas vote dari pohon-pohon keputusan ini untuk melakukan klasifikasi.

3. **K-Nearest Neighbors Classifier**.

   Metode K-Nearest Neighbors adalah algoritma klasifikasi sederhana yang bekerja dengan menghitung jarak antara data point yang diuji dan model training dan memilih kelas K yang memiliki tendensi yang terdekat terhadap data point uji tersebut.

   Algoritma ini bersifat non-parametrik, yang berarti tidak membuat asumsi apa pun tentang data. Teknik ini, yang juga dikenal sebagai "algoritma pembelajar malas", tidak langsung mengenali data training. Sebaliknya, metode ini menyimpan data tersebut dan kemudian mengkategorikannya. Algoritma KNN hanya menyimpan data pada proses training, kemudian algoritma ini mengumpulkan dan

mengkategorikan informasi baru. Data dari masa lalu dan masa kini dikelompokkan bersama dalam kategori yang sama [3].

d. SGD Classifier.

SGD adalah pengklasifikasi linier yang menggunakan penurunan gradien stokastik untuk mempelajari klasifikasi. Gradien kerugian setiap sampel dihitung secara terpisah, dan model diperbarui saat proses berlangsung. Stochastic Gradient Descent Classifier (SGDC) dapat menjadi sangat efektif ketika berurusan dengan sejumlah besar atribut dan data. SGD Classifier dan SGD Regressor cocok dengan model linier dengan log loss menggunakan fungsi loss yang berbeda untuk klasifikasi dan regresi. SGD Classifier cocok dengan Regresi Logistik, Mesin Vektor Dukungan Linier dengan kehilangan engsel, dan Regresi Logistik dengan kehilangan engsel. SGD Classifier menghasilkan model linier yang teratur menggunakan SGD [38].

e. Perceptron Classifier.

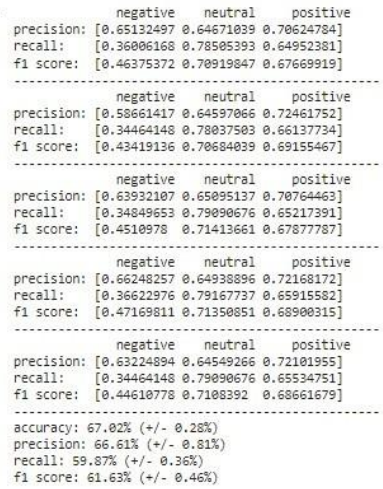
Perceptron adalah algoritma klasifikasi linier. Ini adalah jenis mode jaringan saraf. Masukan, yang sering direpresentasikan sebagai urutan vektor, diperiksa untuk menentukan ke dalam kategori mana masukan tersebut. Perceptron adalah jaringan saraf satu lapis. Algoritma ini terdiri dari empat komponen utama: nilai input, bobot dan bias, jumlah bersih, dan fungsi aktivasi. Perceptron adalah jaringan saraf satu lapis, sedangkan jaringan saraf adalah perceptron multi lapis. Pengklasifikasi linier adalah contoh pengklasifikasi perceptron (biner). Selain itu, strategi ini digunakan dalam pembelajaran yang diawasi. Strategi ini memudahkan untuk mengklasifikasikan data input. Perceptron adalah perangkat matematika yang membagi data menjadi dua segmen yang berbeda. Setelah itu, dijelaskan dengan menggunakan Pengklasifikasi Biner Linier [14].

HASIL DAN PEMBAHASAN

Imbalanced Learning

Seperti yang telah disebutkan sebelumnya, dataset ini tidak terdistribusi secara merata, dengan kelas netral tiga kali lebih besar daripada kelas negatif dan kelas positif 2,5 kali lebih besar daripada kelas negatif. Dalam percobaan ini, kami mencoba menyesuaikan model dengan berbagai variasi data (asli, down sampling, oversampling). Setelah itu, kami mengevaluasi kinerja dari setiap metode pengambilan sampel.

Kami menggunakan linear logistik dengan Tf-IDF Vectorizer untuk membandingkan setiap metode pengambilan sampel. TFIDF adalah singkatan dari *Term Frequency-Inverse Document Frequency*, dan merupakan metode lain untuk mengubah data tekstual menjadi bentuk numerik. Menggabungkan kedua istilah ini, TF dan IDF, menghasilkan nilai vektor. Untuk validasi, teknik K-fold Cross Validation (KCV) digunakan. Teknik K-fold Cross Validation adalah teknik pembelajaran mesin yang populer untuk pemilihan model dan estimasi kesalahan dalam pengklasifikasi. Sebagian himpunan bagian digunakan untuk melatih model secara iteratif, sementara himpunan bagian yang tersisa digunakan untuk menilai kinerjanya [4]. Karena ukuran dataset yang kecil (41705 tweet), kami tidak membaginya untuk pengujian dan validasi, dan dengan demikian berisiko kehilangan wawasan yang menarik.



Gambar 1. Dataset Original yang Tidak Seimbang dengan Linear Regression

Data original yang tidak seimbang

Kami mengevaluasi kinerja dengan data asli yang tidak seimbang dengan menggunakan model regresi linier dengan TF-Idf. Gambar 1 menggambarkan hasilnya. Tanpa menerapkan *resampling* apa pun pada dataset, kami melihat bahwa presisi keseluruhan lebih besar daripada recall. Kelas negatif memiliki recall yang rendah, dengan sekitar 34-36%, dan presisi tinggi 58-66%. Nilai recall yang rendah menunjukkan bahwa pengklasifikasi yang digunakan memiliki jumlah False Negative yang tinggi sebagai akibat dari dataset yang tidak seimbang. Kelas netral memiliki nilai recall dan f1 tertinggi di antara yang lain, dengan recall 78-79% dan nilai f1 70-71% untuk setiap fold. Secara umum, dengan dataset yang tidak seimbang, kami memiliki presisi yang tinggi tetapi recall yang rendah.

Oversampling

Oversampling dengan kelas yang tidak seimbang mengakibatkan peningkatan dataset dimana dataset sintentis adalah duplikat dari sampel sebelumnya yang diambil secara acak dari kumpulan data yang lebih besar. Kita memerlukan jumlah sampel yang cukup untuk melakukan eksperimen secara efektif. Ada banyak teknik pengambilan sampel berlebih yang tersedia; dalam eksperimen ini, kami akan menggunakan dua di antaranya:

1. RandomOverSampling
2. SMOTE (Synthetic Minority Over-Sampling Technique)

Ketika menggunakan data *oversamping* untuk *cross validation*, kita harus mempertimbangkan kemungkinan "*overfitting*". Frasa "*overfitting*" mengacu pada model yang secara akurat memodelkan "dataset training" namun memiliki akurasi yang buruk pada "dataset testing". *Overfitting* terjadi ketika sebuah model mempelajari begitu banyak detail dan noise dari data pelatihan sehingga menurunkan kinerja model pada data baru. *Overfitting* terjadi karena kita membocorkan informasi yang sudah ada di *training* set, jadi ini seperti melatih dengan dataset duplikat. Ketika fungsi validasi silang "lr_cv()" dalam kode sumber digunakan, pipeline hanya akan diisi dengan data dari set pelatihan setelah divalidasi silang. Hasilnya, tidak ada informasi set validasi yang akan bocor ke model.

a. *RandomOverSampling.*

RandomOverSampling mengkompensasi kekurangan data point dari kelas minoritas dengan menghasilkan data point tambahan dari kelas mayoritas. Karena kedua kelas diwakili secara merata, fungsi keputusan pengklasifikasi dapat

mencapai kesimpulan yang terinformasi dengan baik. *Oversampling* terjadi sebagai hasil dari proses replikasi dan meningkatkan bobot kelas minoritas. *Oversampling* tidak menambahkan informasi baru ke dalam dataset, tetapi dengan meningkatkan jumlah sampel yang digunakan, hal ini meningkatkan kemungkinan terjadinya *overfitting* [5] .

Untuk mengilustrasikan cara kerja pengambilan sampel berlebih, misalkan kita memiliki 100 sampel di kelas minoritas dan 1000 sampel di kelas mayoritas. Ini akan secara acak memilih 100 sampel dari kelas minoritas dan menempatkannya di kumpulan data kelas minoritas. Sebagai contoh, kita akan mulai dari baris ke-89 dan terus ke atas halaman, mengulangi prosesnya sampai kelas mayoritas memiliki poin terbanyak. Algoritma pengambilan sampel acak bekerja dengan cara ini.

```
              negative    neutral     positive
precision: [0.50468457 0.69066667 0.70635452]
recall:    [0.6229761  0.66512583 0.67047619]
f1 score:  [0.55762595 0.67765568 0.68794788]
--------------------------------------------------
              negative    neutral     positive
precision: [0.49165673 0.69737562 0.71148087]
recall:    [0.63608327 0.65527871 0.67851476]
f1 score:  [0.55462185 0.6756721  0.69460689]
--------------------------------------------------
              negative    neutral     positive
precision: [0.50428922 0.69771198 0.70474282]
recall:    [0.63454125 0.66581043 0.66962869]
f1 score:  [0.56196654 0.68138801 0.68673718]
--------------------------------------------------
              negative    neutral     positive
precision: [0.51537979 0.69749671 0.72265493]
recall:    [0.63299923 0.67993835 0.67724532]
f1 score:  [0.56816609 0.68860562 0.69921363]
--------------------------------------------------
              negative    neutral     positive
precision: [0.4996873  0.69023034 0.71905565]
recall:    [0.61603701 0.6696635  0.6766106 ]
f1 score:  [0.55179558 0.6797914  0.6971877 ]
--------------------------------------------------
accuracy: 66.39% (+/- 0.40%)
precision: 63.69% (+/- 0.43%)
recall: 65.67% (+/- 0.36%)
f1 score: 64.42% (+/- 0.40%)
```

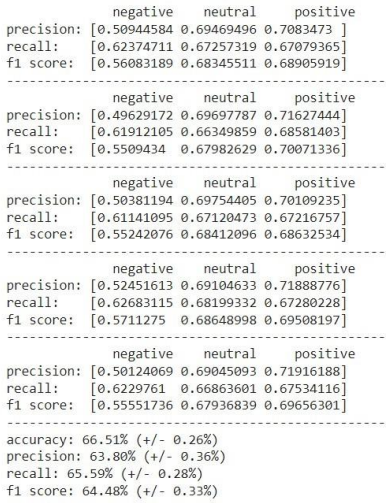
Gambar 2. Random Over Sampling dengan Linear Regression

Kami melihat adanya peningkatan skor recall pada percobaan ini jika dibandingkan dengan percobaan sebelumnya seperti pada gambar 2. Sebelumnya, nilai recall untuk kelas negatif hanya 34-36 persen; namun, setelah mengimplementasikan RandomOverSampling, nilai recall meningkat menjadi sekitar 61-62 persen. Sebaliknya, nilai presisi untuk kelas negatif menurun dari 58-66 persen menjadi 49-51 persen. Percobaan ini menunjukkan recall yang tinggi tetapi presisi yang rendah, menyiratkan tingkat positif palsu yang tinggi. Beberapa teks salah diklasifikasikan sebagai negatif, padahal sebenarnya, beberapa teks memang negatif, tetapi sebagian besar tidak.

Secara keseluruhan, RandomOverSampling meningkatkan skor f1 secara keseluruhan, yang merupakan rata-rata tertimbang dari Precision dan Recall, dari 61.63% menjadi 64.42% ketika data tidak seimbang. Namun, akurasi sedikit lebih rendah dengan RandomOverSampling pada 64,42%, sementara itu 3% lebih tinggi dengan data yang tidak seimbang pada 67,02%.

b. *SMOTE (Synthetic Minority Oversampling Technique).*
Oversampling dapat dicapai hanya dengan mereproduksi elemen kelas minoritas yang ada pada *training* set, seperti yang telah disebutkan sebelumnya di bagian RandomOverSampling. Akan tetapi, teknik ini dikenal rentan terhadap *overfitting* [8, 23]. Untuk mengurangi risiko ini, sampel tambahan dapat dibuat secara artifisial dengan memperhitungkan distribusi kelas minoritas. Salah satu metodenya adalah

dengan menggunakan Teknik Pengambilan Sampel Berlebih Minoritas Sintetis (Synthetic Minority Over-sampling Technique, SMOTE).



Gambar 3. SMOTE (*Synthetic Minority Oversampling Technique)*

Prosedur berikut ini diikuti untuk membuat sampel sintetis: Asumsikan bahwa ada perbedaan antara vektor fitur yang sedang diperiksa (sampel) dan vektor fitur dari tetangga terdekatnya. Perbedaan ini kemudian dikalikan dengan nilai acak antara 0 dan 1 dan ditambahkan ke vektor fitur yang sesuai. Tindakan ini memilih lokasi acak di sepanjang segmen garis yang menghubungkan dua fitur yang berbeda. Dengan menggunakan strategi ini, wilayah keputusan kelas minoritas secara efektif dipaksa untuk menjadi lebih beragam." Ini berarti bahwa ketika SMOTE menghasilkan data sintetis baru, SMOTE akan memilih satu data untuk diduplikasi dan memeriksa k tetangga terdekatnya sebelum membuat data sintetis baru. SMOTE akan menghasilkan nilai acak dalam ruang fitur antara sampel awal dan tetangganya, dan proses ini akan diulang sampai ruang fitur habis [7]. Dibandingkan dengan RandomOverSampling, seperti yang ditunjukkan pada gambar 3 SMOTE berkinerja lebih baik di semua skor metrik, dengan akurasi yang sedikit lebih tinggi serta presisi, recall, dan skor f1. Pengambilan sampel SMOTE tampaknya memiliki masalah yang sama dengan RandomOverSampling, menghasilkan recall yang tinggi tetapi presisi yang rendah untuk negatif.

Secara keseluruhan, ketika menguji dengan metode oversampling, SMOTE mengungguli RandomOverSampling.

Downsampling

Down-sampling adalah proses memilih dan membuang observasi secara acak dari kelas mayoritas untuk mencegah sinyal kelas mayoritas mendominasi algoritma pembelajaran selama fase pembelajaran [31]. Heuristik yang paling sering digunakan untuk hal ini adalah *resampling* tanpa penggantian.

Hal ini analog dengan prosedur sebelumnya yang dikenal sebagai *oversampling*. Langkah-langkahnya adalah sebagai berikut: Untuk memulai, kita akan memisahkan observasi ke dalam DataFrame untuk setiap kelas yang akan kita analisis. Pada langkah berikutnya, kelas mayoritas akan disampling ulang tanpa penggantian dengan ukuran sampel yang sama dengan kelas minoritas. Terakhir, kita akan menghubungkan DataFrame kelas mayoritas yang telah disampling ke DataFrame kelas minoritas untuk menyelesaikan transformasi.

a. *RandomUnderSampler.*

RandomUnderSampler menghilangkan sebagian besar kemunculan kelas secara acak untuk membuat kumpulan data yang seimbang [32]. Hasilnya, versi set data pelatihan yang telah disesuaikan akan mengurangi jumlah contoh di kelas mayoritas dengan faktor dua. Teknik ini dapat diulang sebanyak yang diperlukan untuk mencapai distribusi kelas yang diinginkan, seperti jumlah sampel yang sama di masing-masing dari tiga kelas [13].

Strategi ini mungkin lebih sesuai untuk dataset dengan ketidakseimbangan kelas seperti yang kita temui dalam kasus studi ini; namun, jika kelas minoritas memiliki jumlah contoh yang cukup, model yang berguna dapat disesuaikan dengan metode ini. *Undersampling* memiliki kelemahan yaitu menghapus contoh dari kelas mayoritas yang mungkin berguna, signifikan, atau bahkan penting dalam menyesuaikan batas keputusan yang kuat dengan model batas keputusan yang diberikan. Karena contoh dihapus secara acak, tidak ada cara untuk mengidentifikasi atau menyimpan contoh yang "baik" atau lebih kaya informasi dari kelas mayoritas, yang menjadi masalah karena dihapus secara acak [13].

Akurasi keseluruhan lebih rendah setelah mengimplementasikan RandomUnderSampler dibandingkan dengan percobaan pertama dengan data yang tidak seimbang. Karakteristik presisi rendah dan recall tinggi tetap ada pada percobaan ini, dan skor f1 secara keseluruhan secara signifikan lebih rendah daripada sebelumnya.

```
              negative    neutral     positive
precision: [0.43133641 0.69519899 0.68186356]
recall:    [0.72166538 0.56522856 0.65047619]
f1 score:  [0.53994808 0.62351275 0.66580016]
---------------------------------------------------
              negative    neutral     positive
precision: [0.42179885 0.70188557 0.68783245]
recall:    [0.73400154 0.55458515 0.65661695]
f1 score:  [0.53573438 0.61960109 0.67186232]
---------------------------------------------------
              negative    neutral     positive
precision: [0.43217232 0.69839774 0.67977151]
recall:    [0.72706245 0.57102492 0.64201841]
f1 score:  [0.5421098  0.62832109 0.6603558 ]
---------------------------------------------------
              negative    neutral     positive
precision: [0.43567518 0.71061919 0.69195251]
recall:    [0.73631457 0.56896995 0.66582037]
f1 score:  [0.5474348  0.63195435 0.67863497]
---------------------------------------------------
              negative    neutral     positive
precision: [0.42614679 0.69604768 0.69492096]
recall:    [0.71626831 0.56999743 0.65566487]
f1 score:  [0.53436871 0.62674763 0.6747224 ]
---------------------------------------------------
accuracy: 62.43% (+/- 0.39%)
precision: 60.57% (+/- 0.37%)
recall: 64.90% (+/- 0.41%)
f1 score: 61.21% (+/- 0.38%)
```

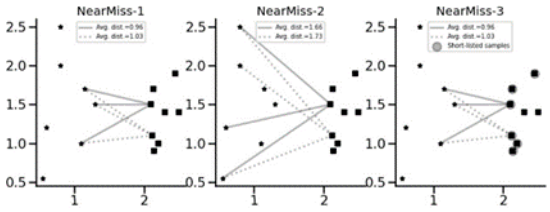
Gambar 4. Random Under Sampler dengan Linear Regression

b. ***NearMiss.***

Algoritma ini disebut sebagai algoritma nyaris-hampir, dan dapat digunakan untuk membantu menyeimbangkan kumpulan data yang tidak seimbang. Ini adalah teknik untuk menyeimbangkan data yang diklasifikasikan sebagai algoritma undersampling. Hal ini dilakukan dengan menganalisis distribusi kelas yang lebih

besar dan secara acak mengambil sampel darinya. Ketika dua titik dalam distribusi termasuk dalam kelas yang berbeda dan cukup dekat satu sama lain, strategi ini menghilangkan titik data dari kelas yang lebih besar dalam upaya untuk mengembalikan keseimbangan distribusi. Berbeda dengan *Random Undersampling*, yang mengambil sampel dari kelas mayoritas secara acak tanpa memperhatikan

aturan apa pun, NearMiss menggabungkan sejumlah heuristik untuk memastikan bahwa sampel dari kelas mayoritas adalah representatif [16].



Gambar 5. Tiga versi dari NearMiss [16]

Cara kerja dari algoritma ini adalah sebagai berikut: pertama, algoritma menghitung dan membagi jarak antara semua titik dalam kelas yang lebih besar dan lebih kecil. Hal ini dapat membuat teknik *undersampling* menjadi lebih mudah. Pilih contoh dari kelas yang lebih besar yang paling mirip dengan contoh dari kelas yang lebih kecil. Koleksi n kelas ini harus disimpan sebelum dapat dihapus. Fungsi ini mengembalikan m*n contoh dari kelas yang lebih besar jika kelas yang lebih kecil memiliki m contoh. Ada tiga versi NearMiss seperti yang disebutkan pada gambar 5.

1) ***Near Miss 1.***

NearMiss-1 memilih contoh kelas mayoritas dengan jarak ratarata terpendek ke N contoh berikutnya[16]. NearMiss-1 melacak lokasi kelas mayoritas yang memiliki jarak rata-rata terpendek dari N titik terdekat di kelas minoritas. Dengan kata lain, ini akan menyimpan semua titik dari kelas mayoritas yang mirip dengan titik-titik dari kelas minoritas.

Jika dibandingkan dengan random undersampling, akurasi, skor F1, dan presisi terlihat menurun seperti yang ditunjukkan pada gambar 6. Namun, karena *undersampling* pada kelas mayoritas, nilai recall masih lebih tinggi daripada presisi.

```
                negative    neutral     positive
precision: [0.41171171 0.69303136 0.65672101]
recall:    [0.70470316 0.51078582 0.67777778]
f1 score:  [0.51976116 0.58811354 0.66708327]
--------------------------------------------------------
                negative    neutral     positive
precision: [0.40521114 0.70330056 0.66483012]
recall:    [0.69545104 0.51451323 0.68930498]
f1 score:  [0.51206358 0.59427385 0.67684637]
--------------------------------------------------------
                negative    neutral     positive
precision: [0.41192412 0.69941197 0.65327565]
recall:    [0.70316114 0.51939378 0.67089813]
f1 score:  [0.51951011 0.59610849 0.66196963]
--------------------------------------------------------
                negative    neutral     positive
precision: [0.41286863 0.70930638 0.66512059]
recall:    [0.71241326 0.52273311 0.68264043]
f1 score:  [0.52277228 0.60189293 0.67376664]
--------------------------------------------------------
                negative    neutral     positive
precision: [0.4015887  0.69463315 0.67007346]
recall:    [0.70161912 0.52530182 0.66582037]
f1 score:  [0.5108055  0.59821559 0.66794015]
--------------------------------------------------------
accuracy: 60.73% (+/- 0.31%)
precision: 59.02% (+/- 0.31%)
recall: 63.31% (+/- 0.32%)
f1 score: 59.41% (+/- 0.29%)
```

Gambar 6. Near Miss 1 dengan Linear Regression

2) ***Near Miss 2.***

NearMiss-2, berbeda dengan NearMiss-1, menyimpan titik-titik dari kelas mayoritas yang memiliki jarak rata-rata terpendek ke titik-titik terjauh kelas minoritas. Dengan kata lain, ini akan mempertahankan karakteristik kelas dominan yang berlawanan dengan karakteristik kelas minoritas.

Seperti yang diilustrasikan pada Gambar 7, NearMiss-2 memiliki kinerja terburuk dari varian NearMiss. Akurasinya sedikit menurun menjadi 56,36 persen, dan nilai metrik lainnya juga lebih rendah.

```
                negative    neutral     positive
precision: [0.33019786 0.6789511  0.71674877]
recall:    [0.7848882  0.49203903 0.55428571]
f1 score:  [0.46484018 0.57057772 0.62513426]
--------------------------------------------------
                negative    neutral     positive
precision: [0.33080974 0.67800772 0.73817292]
recall:    [0.77486507 0.49653224 0.57442082]
f1 score:  [0.46366782 0.5732503  0.64608246]
--------------------------------------------------
                negative    neutral     positive
precision: [0.3243071  0.68515797 0.72379032]
recall:    [0.7848882  0.47906499 0.56966043]
f1 score:  [0.45897205 0.56386999 0.63754218]
--------------------------------------------------
                negative    neutral     positive
precision: [0.32366845 0.68333333 0.74335548]
recall:    [0.79182729 0.48445929 0.56807363]
f1 score:  [0.45950783 0.56696227 0.64400072]
--------------------------------------------------
                negative    neutral     positive
precision: [0.32596336 0.68211441 0.74222959]
recall:    [0.79568234 0.48394554 0.56839099]
f1 score:  [0.46246919 0.56619083 0.64378145]
--------------------------------------------------
accuracy: 56.39% (+/- 0.30%)
precision: 58.05% (+/- 0.33%)
recall: 61.35% (+/- 0.23%)
f1 score: 55.65% (+/- 0.28%)
```

Gambar 7. Near Miss 2 dengan Linear Regression

3) ***Near Miss 3.***

Algoritma NearMiss-3 adalah algoritma dua tahap yang menggabungkan fitur-fitur terbaik dari algoritma NearMiss-1 dan NearMiss2. Pertama-tama, algoritma ini menyimpan N tetangga terdekat dari setiap contoh kelas mayoritas. Akhirnya, contoh kelas mayoritas adalah mereka yang memiliki jarak rata-rata terbesar antara mereka dan N tetangga terdekatnya [16].

```
                negative    neutral     positive
precision: [0.44832905 0.68180333 0.63628186]
recall:    [0.67232074 0.53595275 0.67365079]
f1 score:  [0.53793954 0.60014378 0.65443331]
--------------------------------------------------
                negative    neutral     positive
precision: [0.44078947 0.67963022 0.64807931]
recall:    [0.67154973 0.54764963 0.66391622]
f1 score:  [0.53223342 0.60654339 0.65590218]
--------------------------------------------------
                negative    neutral     positive
precision: [0.45724713 0.68039591 0.63050744]
recall:    [0.67617579 0.54739276 0.6585211 ]
f1 score:  [0.54556765 0.60669039 0.64420987]
--------------------------------------------------
                negative    neutral     positive
precision: [0.45594823 0.70577452 0.6819222 ]
recall:    [0.70624518 0.59337272 0.66201206]
f1 score:  [0.55414398 0.64471114 0.67181965]
--------------------------------------------------
                negative    neutral     positive
precision: [0.4340176  0.68777568 0.66933675]
recall:    [0.6846569  0.56075006 0.66296414]
f1 score:  [0.53125935 0.61780105 0.6661352 ]
--------------------------------------------------
accuracy: 61.70% (+/- 1.05%)
precision: 59.59% (+/- 0.98%)
recall: 63.45% (+/- 1.03%)
f1 score: 60.46% (+/- 0.98%)
```

Gambar 8. Near Miss 3 dengan Linear Regression

Secara keseluruhan, seperti yang ditunjukkan pada gambar 8, NearMiss 3 mengungguli jenis NearMiss lainnya, tetapi masih kalah dengan RandomUnderSampling dalam hal akurasi, presisi, recall, dan skor f1.

	Before			**After**		
	Neutral	*Positive*	*Negative*	*Neutral*	*Positive*	*Negative*
NearMiss-1	19466	15754	6485	6485	6485	6485
NearMiss-2	19466	15754	6485	6485	6485	6485
NearMiss-3	19466	15754	6485	6485	6485	6485

Gambar 9. Distribusi data sebelum dan sesudah menerapkan NearMiss-1, NearMiss-2, dan NearMiss-3

Gambar 9 menunjukkan bahwa setelah mengimplementasikan semua varian NearMiss, distribusi kelas menjadi seimbang untuk ketiga kelas (netral, positif, dan negatif).

Pembahasan

Berdasarkan hasil pada gambar 10, kita dapat melihat bahwa untuk teknik ketidakseimbangan data yang menggunakan oversampling maupun downsampling, metode SMOTE memiliki akurasi yang paling tinggi di antara yang lain, begitu pula dengan presisi, recall, dan f1-score yang baik.

	accuracy	macro average precision	macro average recall	macro average f1 score
Original Imbalanced data	67.02% (+/- 0.28)	66.61% (+/- 0.81)	59.87% (+/- 0.36)	61.63% (+/- 0.46)
-oversampling-				
RandomOver Sampler	66.39% (+/- 0.40)	63.69% (+/- 0.43)	65.67% (+/- 0.36)	64.42% (+/- 0.40)
SMOTE	66.51% (+/- 0.26)	63.80% (+/- 0.36)	65.59% (+/- 0.28)	64.48% (+/- 0.33)
-downsampling-				
DownUnder Sampler	62.43% (+/- 0.39)	60.57% (+/- 0.37)	64.90% (+/- 0.41)	61.21% (+/- 0.38)
NearMiss-1	60.73% (+/- 0.31)	59.02% (+/- 0.31)	63.31% (+/- 0.32)	59.41% (+/- 0.29)
NearMiss-2	56.39% (+/- 0.30)	58.05% (+/- 0.33)	61.35% (+/- 0.23)	55.65% (+/- 0.28)
NearMiss-3	61.70% (+/- 1.05)	59.59% (+/- 0.98)	63.45% (+/- 1.03)	60.46% (+/- 0.98)

Gambar 10. Hasil dari percobaan validasi silang 5 kali lipat (Menggunakan Regresi Logistik Tanpa Penyetelan Hiperparameter)

Selain itu, kami mengevaluasi metode SMOTE dengan berbagai pengklasifikasi untuk menentukan seberapa baik kinerjanya. Sejumlah publikasi sebelumnya telah menunjukkan bahwa Regresi Logistik memiliki kinerja terbaik dalam skenario ini, yang juga didukung oleh hasil percobaan ini, yang ditunjukkan pada Gambar 11. Regresi logistik adalah salah satu pengklasifikasi yang paling efektif untuk set data teks.

	accuracy	macro average precision	macro average recall	macro average f1 score
Logistic Regression	*66.46% (+/- 0.69%)*	*63.77% (+/- 0.77%)*	*65.63% (+/- 0.86%)*	*64.48% (+/- 0.81%)*
Random Forest	53.25% (+/- 1.56%)	51.70% (+/- 1.71%)	50.97% (+/- 1.29%)	49.62% (+/- 1.47%)
KNN	42.68% (+/- 6.23%)	50.48% (+/- 2.49%)	39.66% (+/- 5.08%)	30.50% (+/- 3.99%)
Perceptron	63.59% (+/- 0.57%)	61.19% (+/- 0.66%)	59.89% (+/- 0.72%)	60.41% (+/- 0.64%)
Stochastic Gradient Descent	64.19% (+/- 0.44%)	62.19% (+/- 0.40%)	66.44% (+/- 0.41%)	62.88% (+/- 0.45%)

Gambar 11. Hasil dari percobaan validasi silang 5 kali lipat dengan SMOTE (Menggunakan Regresi Logistik, Random Forest, KNN, Perceptron, Stochastic Gradient Descent)

Karena ukuran dataset setelah penerapan imbalanced learning yang kecil untuk setiap kelas, hal ini mempengaruhi performa dari *classifier* yang diuji, terutama Random Forest dan KNN. Ketika dataset pelatihan cukup besar, pengklasifikasi ini bekerja dengan baik. Selain itu, setelah melakukan langkah *preprocessing*, terutama ketika menghapus karakter yang berulang, hasilnya masih ambigu. Sebagai contoh,

menghapus karakter 'o' yang tidak perlu dari kata 'soooo' dan menyisakan dua karakter 'o' saja. Karena 'so' dan 'soo' memiliki arti yang sedikit berbeda, hal ini dapat menyebabkan kebingungan. Beberapa set data tidak diurai dengan benar; misalnya, ada dua set data yang tidak dipisahkan dengan benar menggunakan pemisah, sehingga menyebabkan tweet saling bertumpuk. Pada akhirnya, kami memutuskan untuk mengabaikan dataset ini dan menggunakan sembilan dataset lainnya. Hasilnya, kami memiliki dataset yang sedikit lebih kecil untuk diproses, yang membantu kinerja model. Terlepas dari kendala-kendala ini, kami dapat mengatasi masalah recall rendah presisi tinggi yang terjadi pada data asli yang tidak seimbang dengan menggunakan metode SMOTE dari pengambilan sampel yang berlebihan. Ketika dikombinasikan dengan pengklasifikasi Regresi Logistik, metode ini mengungguli semua metode pembelajaran tidak seimbang lainnya.

SIMPULAN

Dataset yang disediakan tidak proporsional dan memerlukan prapemrosesan yang ekstensif. Ada 11 dataset yang disediakan, tetapi kami menghapus dua dataset karena format file yang tidak tepat. Beberapa entri tampaknya tidak dipisahkan dengan separator tab yang tepat, menyebabkan beberapa tweet menjadi saling bertumpuk dan melebihi panjang tweet standar 140 karakter. Dataset dibagi menjadi tiga kelas (netral, positif, dan negatif), dan distribusi kelas-kelas ini tidak seimbang. Data yang tidak seimbang menyebabkan overfitting, di mana model berkinerja baik di kelas mayoritas tetapi buruk di kelas minoritas. Untuk mengatasi masalah ini, kami menggunakan undersampling dan oversampling dengan berbagai metode yang tersedia di keduanya. Kami menemukan bahwa metode SMOTE dari oversampling berkinerja baik dan memberikan akurasi tertinggi di antara yang lain seperti yang disebutkan pada Gambar 11.

Selain itu, penelitian sebelumnya telah menunjukkan bahwa regresi logistik berkinerja baik dengan dataset SEMEVAL, yang didukung oleh percobaan yang kami lakukan dalam penelitian ini, seperti yang ditunjukkan pada Gambar 11.

DAFTAR PUSTAKA

[1] Ajith Abraham, Jaime Lloret Mauri, John Buford, Junichi Suzuki, and Sabu M Thampi. 2011. Advances in Computing and Communications, Part I: First International Conference, ACC 2011, Kochi, India, July 22-24, 2011. Proceedings. Vol. 190. Springer Science & Business Media.

[2] Charu C Aggarwal et al. 2016. Recommender systems. Vol. 1. Springer.

[3] Syed Thouheed Ahmed, Syed Muzamil Basha, Sajeev Ram Arumugam, and Mallikarjun M Kodabagi. 2021. Pattern Recognition: An Introduction. MileStone Research Publications.

[4] Davide Anguita, Luca Ghelardoni, Alessandro Ghio, Luca Oneto, and Sandro Ridella. 2012. The 'K' in K-fold cross validation. In 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN). i6doc. com publ, 441–446.

[5] Francis Effirim Botchey, Zhen Qin, and Kwesi Hughes-Lartey. 2020. Mobile Money Fraud Prediction—A Cross-Case Analysis on the Efficiency of Support Vector Machines, Gradient Boosted Decision Trees, and Naïve Bayes Algorithms. Information 11, 8 (2020), 383.

[6] Danah Boyd, Scott Golder, and Gilad Lotan. 2010. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In 2010 43rd Hawaii international conference on system sciences. IEEE, 1–10.

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 16 (2002), 321–357.

[8] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. 2004. Special issue on learning from imbalanced data sets. ACM SIGKDD explorations newsletter 6, 1 (2004), 1–6.

[9] Gilles Cohen, Mélanie Hilario, Hugo Sax, Stéphane Hugonnet, and Antoine Geissbuhler. 2006. Learning from imbalanced data in surveillance of nosocomial infection. Artificial intelligence in medicine 37, 1 (2006), 7–18.

[10] Montserrat Comesaña, Ana Paula Soares, Manuel Perea, Ana P Piñeiro, Isabel Fraga, and Ana Pinheiro. 2013. ERP correlates of masked affective priming with emoticons. Computers in Human Behavior 29, 3 (2013), 588–595.

[11] Thomas Edgar and David Manz. 2017. Research methods for cyber security. Syngress.

[12] Rajesh Sinha Priyankar Sinha Adarsh Pradhan Eriq-Ur Rahman, Rituparna Sarma. 2018. A Survey on Twitter Sentiment Analysis. International Journal of Computer Sciences and Engineering 6 (11 2018), 644–648. Issue 11. https://doi.org/10.26438/ ijcse/v6i11.644648

[13] Haibo He and Yunqian Ma. 2013. Imbalanced learning: foundations, algorithms, and applications. (2013).

[14] Kamal Kant Hiran, Ritesh Kumar Jain, Kamlesh Lakhwani, and Ruchi Doshi. 2021. Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition). BPB Publications.

[15] Bernard J Jansen, Mimi Zhang, Kate Sobel, and Abdur Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. Journal of the American society for information science and technology 60, 11 (2009), 2169–2188.

[16] Annie Kim. 2021. Optimal Selection of Resampling Methods for Imbalanced Data with High Complexity. Ph. D. Dissertation. Department of Biostatistics and Computing and the Graduate School of Yonsei University.

[17] Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research 18, 1 (2017), 559–563.

[18] Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. 2014. TeamX: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). 628–632.

[19] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016). 31–41.

[20] Saif M Mohammad. 2016. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In Emotion measurement. Elsevier, 201–237.

[21] Maria Carolina Monard and GEAPA Batista. 2002. Learning with skewed class distributions. Advances in Logic, Artificial Intelligence and Robotics 85 (2002), 173–180.

[22] Aminu Muhammad, Nirmalie Wiratunga, and Robert Lothian. 2014. A hybrid sentiment lexicon for social media mining. In 2014 IEEE 26th International Conference on Tools with Artificial Intelligence. IEEE, 461–468.

[23] Iman Nekooeimehr and Susana K Lai-Yuen. 2016. Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. Expert Systems with Applications 46 (2016), 405–416.

[24] Elisavet Palogiannidi, Athanasia Kolovou, Fenia Christopoulou, Filippos Kokkinos, Elias Iosif, Nikolaos Malandrakis, Harris Papageorgiou, Shrikanth Narayanan, and Alexandros Potamianos. 2016. Tweester at SemEval-2016 Task 4: Sentiment analysis in Twitter using semantic-affective model adaptation. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). 155–163.

[25] Georgios Paltoglou, Stéphane Gobron, Marcin Skowron, Mike Thelwall, and Daniel Thalmann. 2010. Sentiment analysis of informal textual communication in cyberspace. In Proc. Engage 2010, Springer LNCS State-of-the-Art Survey (2010), 13–25.

[26] Georgios Paltoglou and Mike Thelwall. 2012. Twitter, MySpace, Digg: Unsupervised sentiment analysis in social media. ACM Transactions on Intelligent Systems and Technology (TIST) 3, 4 (2012), 1–19.

[27] Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. Found.Trends Inf. Retr. 2, 1–2 (jan 2008), 1–135. https://doi.org/10.1561/1500000011

[28] Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In International workshop on semantic evaluation. 19–30.

[29] Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). 486–495.

[30] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). Association for Computational Linguistics, Dublin, Ireland, 27–35. https://doi.org/10.3115/v1/S14-2004

[31] Alireza Rahnama, Sam Clark, and Seetharaman Sridhar. 2018. Machine learning for predicting occurrence of interphase precipitation in HSLA steels. Computational Materials Science 154 (2018), 169–177.

[32] Farshid Rayhan, Sajid Ahmed, Asif Mahbub, Rafsan Jani, Swakkhar Shatabda, and Dewan Md Farid. 2017. Cusboost: Cluster-based under-sampling with boosting for imbalanced classification. In 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS). IEEE, 1–5.

[33] Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017). 502–518.

[34] Sara Rosenthal and Kathleen McKeown. 2016. Social proof: The impact of author traits on influence detection. In Proceedings of the First Workshop on NLP and Computational Social Science. 27–36.

[35] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. SemEval-2015 Task 10: Sentiment Analysis in Twitter. 451-463. In Proc. of the 9th International Workshop on Semantic Evaluation.

[36] Andrew P Sage, RM HARALICK, HE KOENIG, MH MICKLE, CP NEUMAN, HL OESTREICHER, and F DICESARE. 1979. IEEE Transactions on Systems, Man, and Cybernetics. (1979).

[37] Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). 70–74.

[38] Meenakshi Tripathi and Sushant Upadhyaya. 2021. Conference Proceedings of ICDLAIR2019. Vol. 175. Springer Nature.

[39] Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. 2010. Predicting elections with twitter: What 140 characters reveal about political sentiment. In Proceedings of the International AAAI Conference on Web and Social Media, Vol. 4.

[40] Hao Wang, Doğan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In Proceedings of the ACL 2012 system demonstrations. 115–120.

[41] Sabih Bin Wasi, Rukhsar Neyaz, Houda Bouamor, and Behrang Mohit. 2014. CMUQ@ Qatar: Using rich lexical features for sentiment analysis on Twitter. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014). 186–191.

[42] Zhihua Zhang, Guoshun Wu, and Man Lan. 2015. Ecnu: Multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 561–567.