Project Manajemen Data dan Informasi

Analisis Ujaran Kebencian Indonesia di Twitter/X Pasca Debat Capres Kedua Tahun 2024

Analisis ini berupaya menyelidiki sifat dan prevalensi ujaran kebencian di beberapa tweet yang menjadi trending setelah Debat Presiden Indonesia kedua pada tahun 2024 di Twitter/X. Memanfaatkan berbagai model pembelajaran mesin, antara lain K-Nearest Neighbors (KNN), Logistic Regression, SVM, Naive Bayes, Decision Tree, dan Random Forest, penelitian ini bertujuan untuk mengetahui apakah tweet yang

Daftar Isi

Abstrak

 Pendahuluan • 1. Persiapan

• 2. Pengumpulan Data • 3. Pengolahan Data

• 5. Machine Learning

• 6. Prediksi Kesimpulan Referensi

Abstrak

• 4. Eksplorasi, Analisis dan Visualisasi Data

from wordcloud import WordCloud from sklearn.feature_extraction.text import TfidfVectorizer from sklearn.model_selection import train_test_split from sklearn.metrics import classification_report, accuracy_score, precision_score, recall_score, f1_score, confusion_matrix from sklearn.neighbors import KNeighborsClassifier from sklearn.linear_model import LogisticRegression from sklearn.svm import SVC from sklearn.naive_bayes import ComplementNB, BernoulliNB from sklearn.tree import DecisionTreeClassifier $from \ sklearn.ensemble \ import \ Random Forest Classifier$ from sklearn.model_selection import GridSearchCV warnings.filterwarnings("ignore") [nltk_data] Downloading package stopwords to C:\Users\Rivaldi\AppData\Roaming\nltk_data... [nltk_data] [nltk_data] Package stopwords is already up-to-date! 1.2. Membuat Fungsi Setelah mengimpor semua *library* yang diperlukan, tahap selanjutnya adalah membuat fungsi-fungsi yang akan membantu dalam analisis, berikut beberapa fungsi tersebut. 1.2.1. Fungsi pengelola teks Ada beberapa fungsi yang dibuat untuk mengolah teks seperti mengubah teks menjadi huruf kecil, menghilangkan karakter yang tidak perlu seperti url, rt, user, emoji, dll. Lalu ada juga fungsi untuk menghilangkan teks non alfanumerik, normalisasi kata alay, menghilangkan stop word, spasi berlebih, dan melakukan stemming. factory = StemmerFactory() stemmer = factory.create_stemmer() def lowercase(text): return text.lower() def remove_unnecessary_char(text): text = re.sub('\n',' ',text) text = re.sub('rt',' ',text) text = re.sub('user',' ',text) text = re.sub('url',' ', text) $text = re.sub('((www\.[^\s]+)|(https?://[^\s]+)|(http?://[^\s]+))',' ', text)$ text = re.sub(' +', ' ', text) return text

In [11]: nltk.download('stopwords') Out[11]: In [4]: def remove_unicode(text): text = $re.sub(r'\bx[a-fA-F0-9]{2}\b', '', text)$ $text = re.sub(r'\bx([a-fA-F0-9]{2})', '', text)$ return text def remove_nonaplhanumeric(text): text = re.sub(' $\lceil ^0-9a-zA-Z \rceil + '$, ' ', text) return text alay_dict_map = dict(zip(df2["alay"], df2["arti"])) def normalize_alay(text): return ' '.join([alay_dict_map[word] if word in alay_dict_map else word for word in text.split(' ')]) def remove_stopword(text): text = ' '.join(['' if word in df3.stopword.values else word for word in text.split(' ')])
text = re.sub(' +', ' ', text) text = text.strip() return text def stemming(text): return stemmer.stem(text) def remove_extra_spaces(text): $text = re.sub(r'\s+', ' ', text)$ text = text.strip() return text In []: def preprocess(text): text = lowercase(text) text = remove_nonaplhanumeric(text) text = remove_unnecessary_char(text) text = normalize_alay(text) text = stemming(text) text = remove_stopword(text) text = remove_unicode(text) text = remove_extra_spaces(text) return text 1.2.2. Fungsi penghasil confusion matrix Fungsi ini dibuat untuk melihat hasil prediksi dari machine learning untuk mencocokkan apakah hasil prediksi tersebut akurat atau tidak untuk keperluan evaluasi.

In [8]: def plot_confusion_matrix(y, y_predict): cm = confusion_matrix(y, y_predict) ax = plt.subplot() sns.heatmap(cm, annot=True, ax=ax, fmt='d') ax.set_xlabel('Predicted labels') ax.set_ylabel('True labels') ax.set_title('Confusion Matrix') ax.xaxis.set_ticklabels(['Bukan Ujaran Kebencian', 'Ujaran Kebencian']) ax.yaxis.set_ticklabels(['Bukan Ujaran Kebencian', 'Ujaran Kebencian']) plt.savefig('charts/confusion_matrix.png', bbox_inches='tight') plt.show() 1.2.3. Fungsi training model Fungsi ini dibuat untuk melakukan pelatihan pada data dengan menggunakan beberapa model sekaligus seperti KNN, Logistic Regression, SVM, Complement Naive Bayes, Bernoulli Naive Bayes, Decision Tree, dan Random Forest. Setelah pelatihan, fungsi tersebut akan mengembalikan modul dengan hasil akurasi tertinggi sekaligus menghasilkan hasil prediksi dalam bentuk confusion matrix. Selain itu, fungsi tersebut juga akan mengembalikan hasil evaluasi seluruh model dalam bentuk dataframe. In [10]: def vector_model(data, category, ngram): X = data['Tweet'].fillna(' ') Y = data[category] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=42) vector = TfidfVectorizer(ngram_range=ngram, stop_words=df3['stopword'].tolist()) X_train_scal = vector.fit_transform(X_train) X_test_scal = vector.transform(X_test) X_test_scal_new = vector.transform(df_debat["Tweet"]) models = { 'KNN': KNeighborsClassifier(n_neighbors=5), 'Logistic Regression': LogisticRegression(), 'SVM': SVC(kernel='rbf'), 'Complement NB': ComplementNB(), 'Bernoulli NB': BernoulliNB(), 'Decision Tree': DecisionTreeClassifier(criterion='entropy', min_samples_split=2, random_state=42),

'Random Forest': RandomForestClassifier(n_estimators=105, min_samples_split=2, random_state=42) } best_accuracy = 0 best_model = None scores = {'Model': [], 'Accuracy': [], 'Precision': [], 'Recall': [], 'F1 Score': []} for model_name, model in models.items(): model.fit(X_train_scal, Y_train) Y_pred = model.predict(X_test_scal) accuracy = accuracy_score(Y_test, Y_pred) if accuracy > best_accuracy: best_accuracy = accuracy best_model = model accuracy = accuracy_score(Y_test, Y_pred) precision = precision_score(Y_test, Y_pred) recall = recall_score(Y_test, Y_pred) f1 = f1_score(Y_test, Y_pred) scores['Model'].append(model_name) scores['Accuracy'].append(accuracy) scores['Precision'].append(precision) scores['Recall'].append(recall) scores['F1 Score'].append(f1) scores_df = pd.DataFrame(scores) if best model: print(f"Best Model: {best_model} with Accuracy: {best_accuracy}") best_model_instance = best_model

best_model_instance.fit(X_train_scal, Y_train)

plot_confusion_matrix(Y_test, Y_pred_best)

Ada 4 dataset yang digunakan di awal dan dimuat ke dalam 4 variabel.

df4 = pd.read_excel("dataset/tweet_capres_2024.xlsx")

- disaat semua cowok berusaha melacak perhatia... 1

41. Kadang aku berfikir, kenapa aku tetap perc...

RT USER: USER siapa yang telat ngasih tau elu?...

3 USER USER AKU ITU AKU\n\nKU TAU MATAMU SIPIT T... 0

2.2. Dataset Kamus Alay

USER USER Kaum cebong kapir udah keliatan dong...

alay

pakcikdahtua

t3tapjokowi

3x

pakcikmudalagi

2.3. Dataset Stopword Indonesia

0 anakjakartaasikasik anak jakarta asyik asyik

print("No best model found")

2. Pengumpulan Data

2.1. Dataset Ujaran Kebencian

df1 = pd.read_csv("dataset/tweet_hs.csv") df2 = pd.read_csv("dataset/kamus_alay.csv")

df1.drop("Unnamed: 0",axis=1, inplace=True)

else:

df1.head()

df2.head()

df3.head()

stopword

ada

adalah

adanya adapun agak

df4.head()

4

0

2

4

0

2

Unnamed: 0

1

Out[13]:

In [14]:

Out[14]:

In [15]:

Out[15]:

In [16]:

Out[16]:

In [42]:

Out[6]:

Out[63]:

Y_pred_best = best_model_instance.predict(X_test_scal)

return best_model, best_accuracy, scores_df, X_test_scal_new

In [3]: #url: https://www.kaggle.com/datasets/ilhamfp31/indonesian-abusive-and-hate-speech-twitter-text?select=data.csv

Dataset pertama terdiri dari data tweet yang diambil pada tanggal 20 Maret 2018 hingga 10 September 2018 yang sudah diberi label sebagai ujaran kebencian atau bukan.

Dataset kedua berisi istilah-istilah alay yang sering digunakan oleh netizen beserta artinya. Dataset ini bertujuan untuk melakukan normalisasi data tweet pada dataset pertama.

Dataset keempat berisi tweet yang diambil pada 9 Januari 2024 setelah debat presiden kedua. Data yang diambil berupa 30 tweet terpopuler pada tanggal tersebut.

Terdapat 5561 data yang terindikasi ujaran kebencian. Untuk menyeimbangkan rasio tersebut, akan diambil 5561 data tweet yang berlabel bukan ujaran kebencian.

Preprocessing dilakukan pada dataset pertama dan keempat untuk menormalkan dan menghilangkan karakter yang tidak diperlukan yang nantinya dapat mempengaruhi hasil prediksi dengan machine

Proses penyeimbangan dilakukan untuk menyeimbangkan frekuensi ujaran kebencian dan bukan ujaran kebencian dengan rasio 50:50. Proses penyeimbangan dimulai dengan melihat jumlah dataset yang

plt.pie(comparation, shadow=True, explode=(0, 0.1), autopct='%1.1f%%', labels=["Bukan ujaran kebencian", "Ujaran kebencian"], colors=colors, startangle=270)

Bukan ujaran kebencian

df3 = pd.read_csv("dataset/stopwordbahasa.csv", header=None).rename(columns={0: 'stopword'})

Tweet HS

arti

pak cik sudah tua

pak cik muda lagi

2.4. Dataset Tweet Setelah Debat Kedua Calon Presiden 2024

"Saudara-saudara ada pula yang nyinggung-nying... Allhamdulilah Jabar 80% Prabowo tapi balihonya...

BREAKING NEWS: Sumber Data Yang Disampaikan An...

Seorang pemimpin tidak akan pernah merendahkan...

AHAHAHAHAHAHAHA TEMEN KELAS GW YANG PRO PRA...

3. Pengolahan Data

df1.to_csv("dataset/cleaned_tweets.csv")

3.2. Menyeimbangkan Data

df1 = df.drop(['Tweet'], axis=1)

diberi label ujaran kebencian.

df1.sum()

df.head()

2

4

5561 dtype: int64

df4.to_csv("dataset/cleaned_tweet_capres_2024.csv")

df = pd.read_csv("dataset/cleaned_tweets.csv") df.drop("Unnamed: 0", axis=1, inplace=True)

0 cowok usaha lacak perhati gue lantas remeh per...

 $df_HS_1 = df[df['HS'] == 1]$

df_HS_bal.shape

(11122, 2)

plt.show()

Ujaran kebencian

hs = df.loc[df["HS"]==1]

bambang yumbang pilih dasar susilo bambang

plt.axis("off")

iya agama_{turun} islam bangsat

partai komunis

cobarakyatgoblok

In [42]: nhs = df.loc[df["HS"]==0]

plt.axis("off")

nama cina <u>∵</u>

kepal

wu milik asing dewan wak

df_debat.head()

plt.axis("off")

plt.show()

an

In [13]:

Bukan Ujaran Kebencian

Ujaran Kebencian

scores_df

2

4

5

1 Logistic Regression

Complement NB

Bernoulli NB

Decision Tree

viz_scores = scores_df.copy()

plt.figure(figsize=(10, 4))

plt.xlabel('Models') plt.ylabel('Values')

plt.tight_layout()

plt.show()

1.0

0.8

0.6

0.4

0.2

0.0

In [22]:

In [23]:

6. Prediksi

 $df4["HS"] = Y_pred$

plt.show()

Not Hate Speech

Kesimpulan

meningkatkan akurasi prediksi.

Social Media (ICWSM), pages 512-515.

https://doi.org/10.1016/j.procs.2020.03.201

Linguistic Association, 10.

Computer Science and Information Systems (ICACSIS), pages 233-238.

Technologies, 13(4), e202348. https://doi.org/10.30935/ojcmt/13603

Referensi

KNN

In [45]: comparation = df4["HS"].value_counts() colors = ["#fc8a77", "#a2d7d8"]

Y_pred = best_model.predict(X_test_scal_new)

plt.title("Hate Speech vs Not Hate Speech Comparison")

3.3%

plt.savefig('charts/pie_chart_debat_comp.png', bbox_inches='tight')

Values

Out[14]:

True labels

Out[12]:

0

4

plt.show()

bangun

prabowo kerja bilang ad

plt.show()

bikin

telat tau edan sarap gue gaul cigax jifla cal ...

41 kadang pikir percaya tuhan jatuh kali kali ...

kaum cebong kafir lihat dongok dungu haha

 $df_HS_bal = pd.concat([df_HS_1, df_HS_0], axis=0)$

 $df_HS_0 = df[df['HS'] == 0].iloc[:5561]$

df_HS_bal.to_csv("dataset/HS.csv")

comparation = df["HS"].value_counts()

plt.title("Ujaran Kebencian vs Bukan Ujaran Kebencian")

42.2%

text = ' '.join(hs['Tweet'].fillna('').astype(str))

plt.imshow(whs, interpolation="bilinear")

plt.savefig('charts/pie_chart_hate_vs_non.png', bbox_inches='tight')

Ujaran Kebencian vs Bukan Ujaran Kebencian

colors = ["#6BBCD1", "#f69c9c"]

ku tau mata sipit lihat

Setelah proses penyeimbangan dilakukan, ekspor dataset untuk tujuan prediksi.

4. 4. Eksplorasi, Analisis dan Visualisasi Data

4.1. Perbandingan Ujaran Kebencian vs Bukan Ujaran Kebencian

Membandingkan jumlah ujaran kebencian dan bukan ujaran kebencian dari data menggunakan diagram lingkaran.

57.8%

4.2. Kata-kata yang Paling Sering Digunakan untuk Indikasi Ujaran Kebencian

whs = WordCloud(width=1000, height=800, background_color="white").generate(text)

ahok isi cebong rezim

gant hancur presider

whs = WordCloud(width=1000, height=800, background_color="white").generate(text)

plt.savefig('charts/non_hate_speech_wordcloud.png', bbox_inches='tight')

· ekonomi

daerah salah

df_debat = pd.read_csv("dataset/cleaned_tweet_capres_2024.csv")

Tweet

whs = WordCloud(width=1000, height=800, background_color="white").generate(text)

df_debat.drop("Unnamed: 0",axis=1, inplace=True)

saudara saudara singgung singgung tanah tanah ... allhamdulilah jabar 80 prabowo baliho doang ma...

istirahat news sumber data anies debat calon p... orang pimpin rendah pijak orang puji bullyan l...

text = ' '.join(df_debat['Tweet'].fillna('').astype(str))

plt.savefig('charts/debat_wordcloud.png', bbox_inches='tight')

selamat

kalahnyata tana bangungoblok yaksa bansos hati goblok bansos bansos hati goblok bansos bansos hati goblok bansos bansos hati goblok bansos banso

hasil prediksi direpresentasikan menggunakan confusion matrix.

Best Model: SVC() with Accuracy: 0.8309859154929577

Confusion Matrix

Predicted labels

drama sannangisin

badandukung

best_model, best_accuracy, scores_df, X_test_scal_new = vector_model(data_HS2, 'HS', (1,2))

286

1395

Ujaran Kebencian

Recall F1 Score

plt.bar(x=[pos + i * 0.1 for pos in x], height=values, width=0.1, label=metric)

SVM

Hasil prediksi menunjukkan, dari seluruh tweet, 96,7% tweet terindikasi sebagai ujaran kebencian

Hate Speech vs Not Hate Speech Comparison

Metrik Untuk Setiap Model

Complement NB Bernoulli NB

Karena model terbaik setelah melakukan pelatihan adalah SVM, maka model ini digunakan untuk memprediksi tweet pasca debat capres kedua tahun 2024

plt.pie(comparation, shadow=True, explode=(0, 0.1), autopct='%1.1f%%', labels=["Hate Speech", "Not Hate Speech"], colors=colors, startangle=180)

Hate Speech

Berdasarkan analisis terhadap 30 tweet populer pasca debat presiden kedua tahun 2024, sekitar 96,7% dari tweet tersebut diidentifikasi sebagai ujaran kebencian. Kata-kata yang sering digunakan dalam tweet ini termasuk "Prabowo", "Anies", "Jokowi", dan berbagai istilah kasar. Menariknya, istilah-istilah ini selaras dengan istilah-istilah yang sering muncul dalam kumpulan data pelatihan tahun 2018, yang

Untuk meningkatkan analisis di masa depan, menambah kumpulan data pelatihan dengan data dari debat presiden pertama dan kedua untuk mendapatkan menawarkan pola penulisan tweet yang baru. Selain itu, penggunaan model neural network seperti LSTM dibandingkan model machine learning dapat meningkatkan kemampuan untuk membedakan pola penulisan dalam tweet, sehingga berpotensi

• Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In International AAAI Conference on Web and

• Ika Alfina, Rio Mulia, Mohamad Ivan Fanany, and Yudo Ekananta. 2017. Hate speech detection in the indonesian language: A dataset and preliminary study. In International Conference on Advanced

• Tumasjan, A., Andranik, S., Sprenger, T., Sandner, P., & Welpe, I. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. Word: Journal of the International

• Kurniawan, S., Gata, W., Puspitawati, D. A., -, N., Tabrani, M., & Novel, K. (2019). Perbandingan Metode Klasifikasi Analisis Sentimen Tokoh Politik Pada Komentar Media Berita Online. Jurnal RESTI

• Omran, E., Al Tararwah, E., & Al Qundus, J. (2023). A comparative analysis of machine learning algorithms for hate speech detection in social media. Online Journal of Communication and Media

menampilkan kata "Jokowi" dan "Prabowo" serta kata-kata kasar yang serupa. Hal ini menunjukkan penggunaan pola kata yang serupa secara terus-menerus selama bertahun-tahun.

• Tansa Trisna Astono Putri. 2018. Analisis dan deteksi hate speech pada sosial twitter berbahasa indonesia. Master's thesis, Faculty of Computer Science, Universitas Indonesia.

• Ansari, M. Z., Aziz, M. B., Siddiqui, M. O., Mehra, H., & Singh, K. P. (2020). Analysis of Political Sentiment Orientations on Twitter. Procedia Computer Science, 167, 1821-1828.

• Komnas HAM. 2015. Buku Saku Penanganan Ujaran Kebencian (Hate Speech). Komisi Nasional Hak Asasi Manusia, Jakarta.

• I Dewa Putu Wijana and Muhammad Rohmadi. 2010. Sosiolinguistik: Kajian, Teori, dan Analisis. Pustaka Pelajar, Yogyakarta.

• Muhammad Okky Ibrohim and Indra Budi. 2019. Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter.

(Rekayasa Sistem Dan Teknologi Informasi), 3(2), 176 - 183. https://doi.org/10.29207/resti.v3i2.935

Models

Decision Tree Random Forest

Accuracy Precision Recall

F1 Score

0.882653 0.103407 0.185126 0.818449 0.832636 0.825481

0.829863 0.833831 0.831843

0.817501 0.782979 0.879857 0.828596

0.677433 0.952779 0.772251 0.751793 0.814704 0.781985

Dari visualisasi yang ditampilkan, kata-kata yang paling sering muncul adalah Prabowo dan Anies (calon presiden), Jokowi, dan lain sebagainya.

1200

1000

800

600

400

Dengan menggunakan dataset yang sudah diseimbangkan, pelatihan dilakukan menggunakan fungsi pelatihan model. Hasil pelatihan menunjukkan model terbaik adalah SVM dengan akurasi 83% dengan

2 ahahahahahahaha teman kelas gue pro prabowo...

plt.imshow(whs, interpolation="bilinear")

saudarā

5. Machine Learning

data_HS2 = pd.read_csv('dataset/HS.csv')

1378

278

Bukan Ujaran Kebencian

Model Accuracy Precision

0.543602

0.823494

0.830986

0.748876

viz_scores.set_index('Model', inplace=True)

values = viz_scores[metric].values

metrics = viz_scores.columns.tolist() for i, metric in enumerate(metrics):

plt.title('Metrik Untuk Setiap Model')

x = range(len(viz_scores))

Random Forest 0.820497 0.815882 0.829050 0.822413

Visualisasi metrik menggunakan grafik batang untuk setiap model

plt.xticks([i + 0.15 for i in x], viz_scores.index) plt.legend(bbox_to_anchor=(1.02, 1), loc='upper left')

plt.savefig('charts/metrics_bar_chart.png', bbox_inches='tight')

Logistic Regression

Tabel metrik dari semua model.

KNN

babe hektar

Dari visualisasi yang ditampilkan, kata yang paling sering muncul adalah orang, gue, Indonesia, dan lain sebagainya,

4.4. Kata-kata yang Paling Sering Digunakan pada Tweet Setelah Debat Kedua Calon Presiden 2024

Visualisasi menggunakan wordcloud untuk melihat kata apa yang paling sering muncul di tweet populer pasca debat capres kedua.

deh 🌬 negara 📕 dengar masuk 🧸 pakai jokowi 💂 iya bodoh

a

budayaallah

asing

pimpin Cina negara berantas korupsi hari ulam

text = ' '.join(nhs['Tweet'].fillna('').astype(str))

plt.imshow(whs, interpolation="bilinear")

banget

Dari visualisasi yang ditampilkan, kata-kata yang paling sering muncul adalah Jokowi (nama presiden), cebong, ganti presiden, dan lain sebagainya.

^{hina} pakai

Visualisasi menggunakan wordcloud untuk melihat kata apa yang paling sering muncul di tweet yang terindikasi bukan ujaran kebencian.

4.3. Kata-kata yang Paling Sering Digunakan untuk Indikasi Bukan Ujaran Kebencian

plt.savefig('charts/hate_speech_wordcloud.png', bbox_inches='tight')

Dari visualisasi yang diberikan, sebagian besar data bukan merupakan ujaran kebencian yaitu 57,8%, sisanya merupakan ujaran kebencian.

Visualisasi menggunakan wordcloud untuk melihat kata apa yang paling sering muncul di tweet yang terindikasi ujaran kebencian.

Tweet HS

3.1. Preprocessing Data

tetap jokowi

tiga kali

Dataset ketiga berisi stopwords yang akan dicocokkan dengan tweet pada dataset pertama yang akan dihapus.

Tweet

learning nantinya. Pemrosesan awal dilakukan dengan fungsi pemrosesan teks yang telah dideklarasikan sebelumnya.

Setelah itu, ekspor dataset yang telah diproses sebelumnya untuk tujuan prediksi menggunakan machine learning.

df1["Tweet"] = df1["Tweet"].fillna('').astype(str).apply(lambda x: preprocess(x)) df4["Tweet"] = df4["Tweet"].fillna('').astype(str).apply(lambda x: preprocess(x))