

Analisis Sentimen Menggunakan Metode Neural Network dan LSTM

Team members:

1. Arief Budiman
2. Muhidin Jaariya
3. Rommy



C O N T E N T S

01.

Pendahuluan

02.

Metode Penelitian

03.

Kesimpulan

PENDAHULUAN





Latar belakang

Analisis sentimen merupakan proses penting dalam menganalisis opini, perasaan, dan sikap yang terkandung dalam teks. Dalam era digital saat ini, jumlah data teks yang dihasilkan melalui media sosial, ulasan pelanggan, dan sumber lainnya semakin meningkat. Untuk mengatasi kompleksitas dan volume besar data tersebut, pendekatan yang digunakan adalah jaringan saraf tiruan (neural network) dan LSTM telah terbukti berhasil dalam analisis sentimen.



Rumusan masalah

Dalam konteks analisis sentimen dengan menggunakan neural network dan LSTM, terdapat beberapa rumusan masalah yang dapat diajukan, antara lain:

1. Bagaimana cara menerapkan analisis sentimen menggunakan Neural Network dan LSTM untuk mengklasifikasikan teks ke dalam tiga kategori sentimen: positif, negatif, dan netral?
2. Seberapa akurat dan efektifkah metode analisis sentimen berbasis Neural Network dan LSTM dalam mengklasifikasikan sentimen pada teks



Tujuan penelitian

1. Menerapkan metode analisis sentimen berbasis Neural Network dan LSTM untuk mengklasifikasikan teks ke dalam tiga kategori sentimen: positif, negatif, dan netral.
2. Mengevaluasi akurasi dan efektivitas metode analisis sentimen berbasis Neural Network dan LSTM dalam memprediksi sentimen pada teks,
3. Memberikan pemahaman yang lebih baik tentang penggunaan Neural Network dan LSTM dalam analisis sentimen dan potensinya untuk mendukung pengambilan keputusan berbasis data.

Metode Penelitian



1. Neural Network

Data Asli

	Text	Sentimen
0	warung ini dimiliki oleh pengusaha pabrik tahu...	positive
1	mohon ulama lurus dan k212 mmbri hujjah partai...	neutral
2	lokasi strategis di jalan sumatera bandung . t...	positive
3	betapa bahagia nya diri ini saat unboxing pake...	positive
4	duh . jadi mahasiswa jangan sombong dong . kas...	negative
...
10995	tidak kecewa	positive
10996	enak rasa masakan nya apalagi kepiting yang me...	positive
10997	hormati partai-partai yang telah berkoalisi	neutral
10998	pagi pagi di tol pasteur sudah macet parah , b...	negative
10999	meskipun sering belanja ke yogya di riau junct...	positive
11000 rows × 2 columns		

Stopwords NLTK

	Text	Sentimen	text_clean	text_filter
0	warung ini dimiliki oleh pengusaha pabrik tahu...	positive	warung ini dimiliki oleh pengusaha pabrik tahu...	warung dimiliki pengusaha pabrik puluhan terke...
1	mohon ulama lurus dan k212 mmbri hujjah partai...	neutral	mohon ulama lurus dan k212 mmbri hujjah partai...	mohon ulama lurus k212 mmbri hujjah partai diw...
2	lokasi strategis di jalan sumatera bandung . t...	positive	lokasi strategis di jalan sumatera bandung t...	lokasi strategis jalan sumatera bandung nya ny...
3	betapa bahagia nya diri ini saat unboxing pake...	positive	betapa bahagia nya diri ini saat unboxing pake...	betapa bahagia nya unboxing paket barang nya b...
4	duh . jadi mahasiswa jangan sombong dong . kas...	negative	duh jadi mahasiswa jangan sombong dong kas...	duh mahasiswa sombong kasih kartu kuning belaj...
...
10995	tidak kecewa	positive	tidak kecewa	kecewa
10996	enak rasa masakan nya apalagi kepiting yang me...	positive	enak rasa masakan nya apalagi kepiting yang me...	enak masakan nya kepiting menyenangkan memilih...
10997	hormati partai-partai yang telah berkoalisi	neutral	hormati partai partai yang telah berkoalisi	hormati partai partai berkoalisi
10998	pagi pagi di tol pasteur sudah macet parah , b...	negative	pagi pagi di tol pasteur sudah macet parah b...	pagi pagi tol pasteur macet parah bikin jengkel
10999	meskipun sering belanja ke yogya di riau junct...	positive	meskipun sering belanja ke yogya di riau junct...	belanja yogya riau junction kali lihat foodlif...

11000 rows × 4 columns

Feature Extraction

```
df_vektor = df.text_clean.tolist()
```

```
count_vect = CountVectorizer()
```

```
# melakukan fitting dan transformasi pada dokumen  
count_vect.fit(df_vektor)
```

```
# melihat hasil representasi bag of words  
X = count_vect.fit_transform(df_vektor)
```

Train - Test

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 40)
```

Model NN

```
GridSearchCV
GridSearchCV(estimator=Pipeline(steps=[('algoritma', MLPClassifier())]),
              param_grid={'algoritma__activation': ['relu', 'tanh', 'logistic'],
                           'algoritma__alpha': [0.1, 0.01, 1],
                           'algoritma__early_stopping': [True],
                           'algoritma__hidden_layer_sizes': [{1}, {10}],
                           'algoritma__learning_rate_init': [0.01]})
  estimator: Pipeline
    Pipeline(steps=[('algoritma', MLPClassifier())])
      MLPClassifier
        MLPClassifier()
```

Model Neural Network

```
{'algoritma__activation': 'logistic',  
 'algoritma__alpha': 0.01,  
 'algoritma__early_stopping': True,  
 'algoritma__hidden_layer_sizes': {10},  
 'algoritma__learning_rate_init': 0.01}
```

Evaluasi Model (Data Training)

	precision	recall	f1-score	support
negative	0.91	0.96	0.94	2775
neutral	0.95	0.89	0.92	915
positive	0.97	0.96	0.97	5110
accuracy			0.95	8800
macro avg	0.95	0.94	0.94	8800
weighted avg	0.95	0.95	0.95	8800

Menggunakan data training

Evaluasi Model (Data Testing)

	precision	recall	f1-score	support
negative	0.79	0.85	0.82	661
neutral	0.82	0.70	0.76	233
positive	0.92	0.91	0.91	1306
accuracy			0.87	2200
macro avg	0.84	0.82	0.83	2200
weighted avg	0.87	0.87	0.87	2200

Menggunakan data testing

2. Long Short-Term Memory (LSTM)

Import package

```
import pandas as pd
import matplotlib.pyplot as plt

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.regularizers import l1
from tensorflow.keras.regularizers import l2
from tensorflow.keras.regularizers import L1L2
from tensorflow.keras.models import load_model

from sklearn.model_selection import train_test_split
import pickle

import re
```

Import Data

```
data = pd.read_csv('train_preprocess.csv', sep='\t', header = None)
df = data.rename(columns={0: 'text', 1: 'label'})
df
```

✓ 0.1s

	text	label
0	warung ini dimiliki oleh pengusaha pabrik tahu...	positive
1	mohon ulama lurus dan k212 mmbri hujjah partai...	neutral
2	lokasi strategis di jalan sumatera bandung . t...	positive
3	betapa bahagia nya diri ini saat unboxing pake...	positive
4	duh . jadi mahasiswa jangan sombong dong . kas...	negative
...
10995	tidak kecewa	positive
10996	enak rasa masakan nya apalagi kepiting yang me...	positive
10997	hormati partai-partai yang telah berkoalisi	neutral
10998	pagi pagi di tol pasteur sudah macet parah , b...	negative
10999	meskipun sering belanja ke yogya di riau junct...	positive

11000 rows × 2 columns

CLEANSING DATA

```
category = pd.get_dummies(df.label)
df_baru = pd.concat([df, category], axis=1)
df_baru = df_baru.drop(columns='label')

df_baru['text'] = df_baru['text'].replace('\n', ' ').str.lower()

def fun(x):
    y = re.sub(r'^a-zA-Z0-9. ', '', x)
    return y

df_baru['text_new'] = df_baru['text'].apply(lambda x : fun(x))

df_baru['text_new'] = df_baru['text_new'].replace(' ', ' ')
df_baru['text_new'] = df_baru['text_new'].replace(' ', ' ')

df_baru.head()
```

MagicPython

	text	negative	neutral	positive	text_new
0	warung ini dimiliki oleh pengusaha pabrik tahu...	0	0	1	warung ini dimiliki oleh pengusaha pabrik tahu...
1	mohon ulama lurus dan k212 mmbri hujjah partai...	0	1	0	mohon ulama lurus dan k212 mmbri hujjah partai...
2	lokasi strategis di jalan sumatera bandung . t...	0	0	1	lokasi strategis di jalan sumatera bandung . t...
3	betapa bahagia nya diri ini saat unboxing pake...	0	0	1	betapa bahagia nya diri ini saat unboxing pake...
4	duh . jadi mahasiswa jangan sombong dong . kas...	1	0	0	duh . jadi mahasiswa jangan sombong dong . kas...

PENAMAAN VARIABEL X, Y DAN PEMBUATAN TOKENIZER

```
x = df_baru['text_new'].values  
y = df_baru[['negative', 'neutral', 'positive']].values
```

✓ 0.0s

```
tokenizer = Tokenizer(num_words=5000, oov_token='x')  
tokenizer.fit_on_texts(x)  
sekuens_x = tokenizer.texts_to_sequences(x)  
padded_x = pad_sequences(sekuens_x)
```

✓ 0.7s

```
pickle.dump(tokenizer, open("feature_New_lstm.sav", "wb"))
```

✓ 0.0s

SPLIT TRAIN , VALIDATION AND TEST

```
x_train, x_test, y_train, y_test = train_test_split(padded_x, y, test_size=0.2, random_state = 4)  
  
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=0.2, random_state = 4)
```

MODEL LSTM

```
%%time

model = tf.keras.Sequential([
    tf.keras.layers.Embedding(input_dim=100000, output_dim=128),
    tf.keras.layers.LSTM(32, dropout=0.2),
    tf.keras.layers.Dense(8, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])

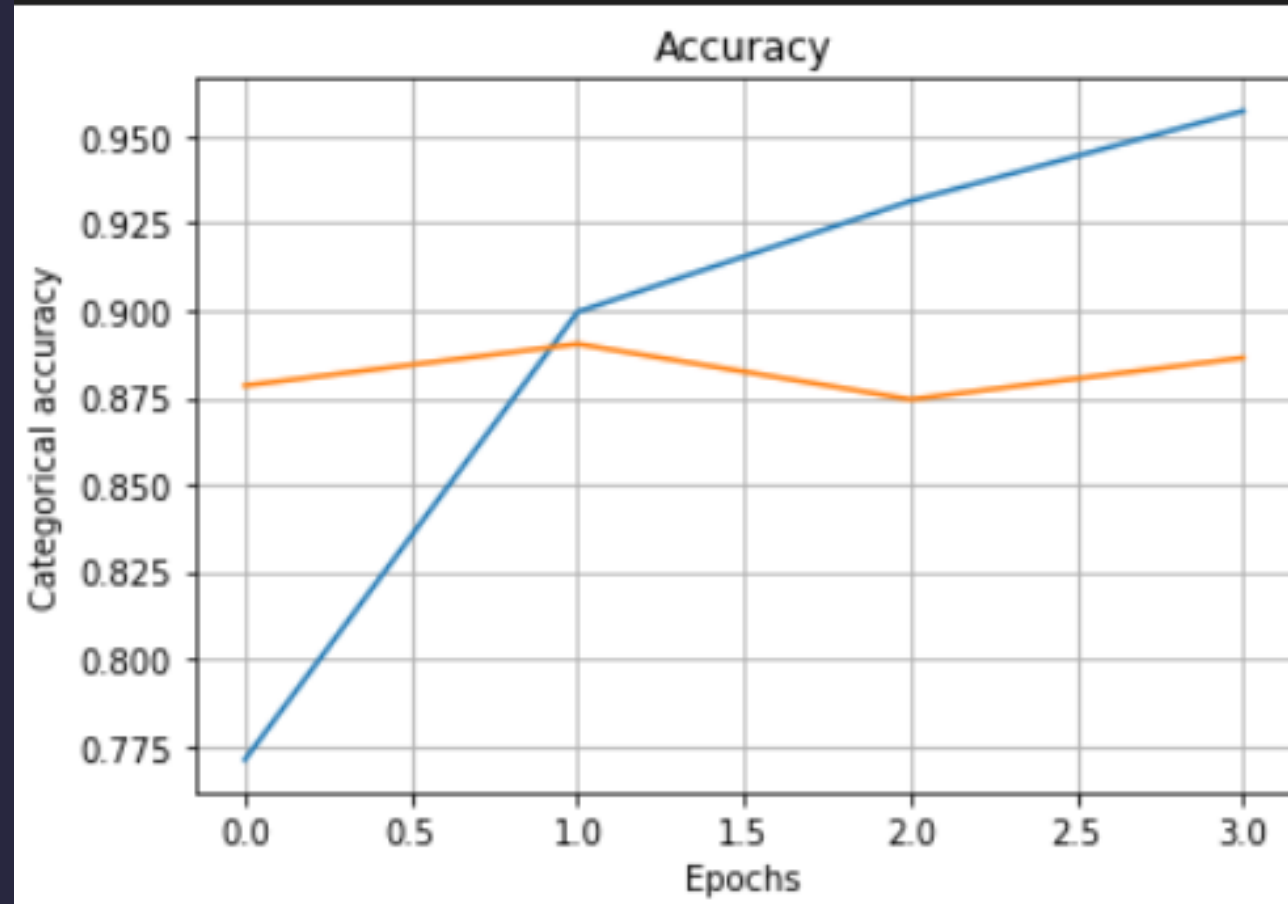
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('val_categorical_accuracy') > 0.90 ):
            self.model.stop_training = True

callbacks = myCallback()

optimizer = keras.optimizers.Adam(learning_rate=0.001)

model.compile(loss='categorical_crossentropy',
              optimizer=optimizer,
              metrics=['categorical_accuracy'])

num_epochs = 4
fit = model.fit(x_train,
                y_train,
                batch_size = 16,
                epochs=num_epochs,
                validation_data=(x_val, y_val),
                callbacks = [callbacks]
)
```



AKURASI

```
train_loss, train_acc = model.evaluate(x_train, y_train)
val_loss, val_acc = model.evaluate(x_val, y_val)
test_loss, test_acc = model.evaluate(x_test, y_test)

print('\nTrain: %.3f, val: %.3f, test: %.3f,' % (train_acc, val_acc, test_acc))
```

✓ 7.1s

```
220/220 [=====] - 5s 16ms/step - loss: 0.0768 - categorical_accuracy: 0.9741
55/55 [=====] - 1s 14ms/step - loss: 0.3503 - categorical_accuracy: 0.8864
69/69 [=====] - 1s 13ms/step - loss: 0.4142 - categorical_accuracy: 0.8659

Train: 0.974, val: 0.886, test: 0.866,
```


3. API

Homepage

[Go to docs page](#)

No	Text	Sentiment	Model
1	Dekat dengan hotel saya menginap, hanya ditempuh jalan kaki, di sini banyak sekali pilihan makanannya, tempat yang luas, dan menyenangkan	positive	LSTM
2	Banyak hal yang dapat kita lakukan untuk mengisi waktu luang.	neutral	LSTM
3	Iya itu yang bikin saya kesal. Ini bukan kesalahan sang pelamar tapi jelas 100% kesalahan pihak yang di atas yang entah bagaimana cara mereka berkordinasi. Nyesek saya itu walaupun ini kasus teman saya yang di lampung.	negative	NN
4	Saya baru saja bertemu teman saya yang bekerja di trans tv	neutral	NN

Click the link of „Go to docs page“ to redirect swagger page

API Documentation for Data Processing and Modeling ^{1.0.0}

[Base URL: 127.0.0.1:5000]

[/docs.json](#)

Dokumentasi API untuk Data Processing dan Modeling

DELETE

GET /delete_all

GET

GET /get_all

POST

POST /manual_input_lstm

POST /manual_input_neural_network

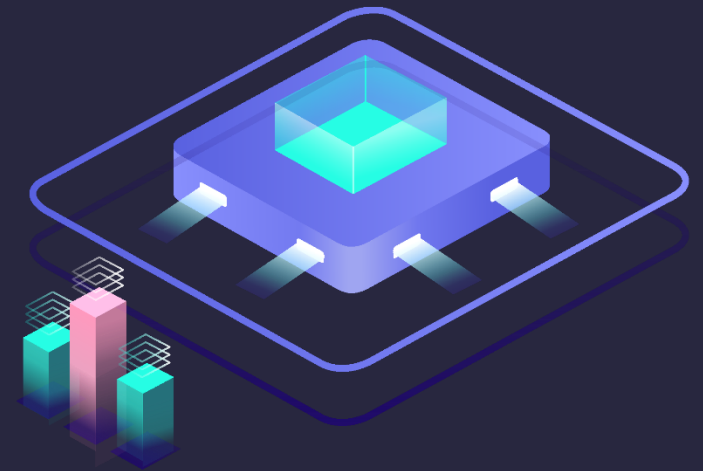
Upload

POST /upload_lstm

POST /upload_neural_network

[Powered by [Flasgger](#) 0.9.5]

Kesimpulan



1. Model LSTM memiliki performa yang baik dengan akurasi yang tinggi pada data pelatihan, validasi, dan pengujian.
2. Model NN juga memberikan hasil yang baik dengan akurasi yang tinggi pada data pengujian dan metrik evaluasi yang seimbang untuk setiap kelas.
3. Kedua model menunjukkan kemampuan yang baik dalam mengklasifikasikan data, tetapi LSTM memiliki keunggulan dalam memahami data sequential seperti teks atau time series.
4. Pilihan model tergantung pada sifat dan karakteristik data serta tujuan spesifik dari masalah yang ingin diselesaikan.

THANKS

