

PROJECT UAS
KECERDASAN BUATAN (AI)
CHAT BOT SEDERHANA



Dosen Pengampu :

Yuyun Umaidah, M.Kom.

Disusun Oleh :

(Ilham Arif Farabi, M.Addien Gilang Saputra)

(2210631170074,2210631170027)

KELAS (D)
TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SINGAPERBANGSA KARAWANG
2024

SOAL

**Buatkan Project Kecerdasan Buatan dengan tema bebas
(yang sudah dipelajari selama kuliah)**

JAWABAN

Latar Belakang

Chatbot adalah program komputer yang dirancang untuk mensimulasikan percakapan dengan pengguna manusia melalui Internet. Dalam beberapa tahun terakhir, chatbot telah meningkat popularitasnya dan telah digunakan di berbagai domain seperti layanan pelanggan, pendidikan, dan hiburan. Pengembangan chatbot melibatkan penggunaan teknologi artificial intelligence (AI) untuk memahami dan merespons bahasa alami manusia. Salah satu pendekatan umum yang digunakan dalam pengembangan chatbot adalah

Metode atau Tahapan

1) Persiapan Data

- Data yang digunakan dalam pelatihan chatbot berasal dari maksud.file json, terdiri dari pola dan daftar tanggapan yang diklasifikasikan dalam beberapa tag.
- Kata-kata dalam pola diberi token menggunakan NLTK dan kemudian diturunkan menggunakan LancasterStemmer.

2) Pemrosesan Data

- Setiap pola telah diubah ke bentuk "bag of words" dan representasi binari dari masing-masing kata unik dalam pola tersebut telah diciptakan.
- Label (tag) juga diubah menjadi bentuk biner dengan metode one-hot encoding.

3) Pembangunan Model Neural Network

- TFLearn digunakan untuk membangun jaringan saraf yang arsitekturnya terdiri dari tiga lapisan: lapisan masukan, dua hidden layers masing-masing dengan 8 neuron, dan output layer dengan aktivasi softmax untuk prediksi klasifikasi.
- Model dilatih pada data pelatihan yang telah diproses sebelumnya.

4) Pelatihan Model

- Model dilatih dengan 1000 epoch dan batch size 8 untuk memastikan bahwa model cukup terlatih untuk mengenali pola dalam data.
- Model yang telah dilatih disimpan ke dalam file 'modelnya.tflearn'.

5) Penggunaan Model

- Saat digunakan, model akan memprediksi tanggapan yang sesuai berdasarkan input pengguna menggunakan fungsi 'bag_of_words' untuk mengubah input menjadi bentuk yang dapat dipahami oleh model.
- Fungsi chat mengatur antarmuka percakapan dengan pengguna, memungkinkan pengguna untuk berinteraksi dengan chatbot.

Code

```
import nltk
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()

import numpy
import tflearn
import tensorflow
from tensorflow.python.framework import ops
ops.reset_default_graph()
import random

nltk.download('punkt')

import json
with open('/content/intents.json') as file:
    data = json.load(file)

words = []
labels = []
docs_x = []
docs_y = []

for intent in data['intents']:
    for pattern in intent['patterns']:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)
        docs_x.append(wrds)
        docs_y.append(intent["tag"])

    if intent["tag"] not in labels:
        labels.append(intent["tag"])

words = [stemmer.stem(w.lower()) for w in words if w != "?"]
words = sorted(list(set(words)))

labels = sorted(labels)

training = []
output = []

out_empty = [0 for _ in range(len(labels))]

for x, doc in enumerate(docs_x):
    bag = []

    wrds = [stemmer.stem(w.lower()) for w in doc]
```

```

for w in words:
    if w in wrds:
        bag.append(1)
    else:
        bag.append(0)

output_row = out_empty[:]
output_row[labels.index(docs_y[x])] = 1

training.append(bag)
output.append(output_row)

training = numpy.array(training)
output = numpy.array(output)

len(output[0])

ops.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(output[0]), activation="softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)

model.fit(training, output, n_epoch=5000, batch_size=8, show_metric=True)
model.save("/content/model.tflearn")
model.load("/content/model.tflearn")

def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]

    s_words = nltk.word_tokenize(s)
    s_words = [stemmer.stem(word.lower()) for word in s_words]

    for se in s_words:
        for i, w in enumerate(words):
            if w == se:
                bag[i] = 1

    return numpy.array(bag)

def chat():

```

```

def chat():
    print("Start talking with the bot (type quit to stop)!")
    while True:
        inp = input("You: ")
        if inp.lower() == "quit":
            break

        results = model.predict([bag_of_words(inp, words)])
        results_index = numpy.argmax(results)
        tag = labels[results_index]

        for tg in data["intents"]:
            if tg['tag'] == tag:
                responses = tg['responses']

        print(random.choice(responses))

chat()

```

Output Code

```
Start talking with the bot (type quit to stop)!
You: hai
Halo. Bagaimana perasaanmu hari ini?
You: baik
Ceritakan lebih banyak
You: aku baik
Tentu! Ceritakan apa yang kamu perlukan.
You: siapa namamu?
Saya Pandora. Saya adalah agen percakapan yang dirancang untuk meniru seorang terapis. Jadi, bagaimana perasaan Anda hari ini?
You: perasaan saya baik
Bisakah Anda jelaskan lebih lanjut?
You: tidak
Bisakah Anda jelaskan lebih lanjut?
You: jangan memaksa
Silakan lanjutkan.
```

Hasil dan Pembahasan

- 1) **Keakuratan Model:** Dengan pelatihan yang dilakukan hingga 1000 epoch, model neural network ini diharapkan memiliki tingkat akurasi yang tinggi dalam mengenali dan mengelompokkan input pengguna berdasarkan pola yang telah dilatih. Hal ini dapat dilihat dari metrik yang ditampilkan selama pelatihan.
- 2) **Fleksibilitas dan Responsivitas:** Chatbot ini dirancang untuk bisa merespons berbagai input pengguna dengan tanggapan yang telah ditentukan dalam file intents.json. Model menggunakan probabilitas untuk memilih respon yang paling tepat berdasarkan input yang diberikan.
- 3) **Kelebihan Chat Bot Sederhana:**
 - Sederhana dan Mudah Diimplementasikan : Implementasi chatbot ini relatif sederhana dan dapat dengan mudah dipahami oleh pemula. Dengan menggunakan pustaka seperti NLTK dan TFLearn, pengembang dapat dengan cepat membangun dan melatih model chatbot.
 - Pembelajaran Mendalam dengan Neural Network : Penggunaan neural network untuk pelatihan memungkinkan chatbot untuk mengenali pola yang kompleks dalam data dan memberikan tanggapan yang lebih akurat dibandingkan dengan metode yang lebih sederhana.

- Penggunaan Python dan Pustaka Populer : Implementasi ini menggunakan Python, bahasa pemrograman yang populer dan memiliki komunitas yang besar, sehingga banyak sumber daya dan dukungan yang tersedia. Pustaka seperti NLTK dan TFLearn juga didukung dengan baik dan sering digunakan dalam pengembangan AI dan NLP.

4) Kekurangan Chat Bot Sederhana:

- Keterbatasan Dataset : Chatbot ini hanya sebaik dataset yang digunakan untuk melatihnya. Jika dataset `intents.json` tidak mencakup berbagai variasi pola bahasa pengguna atau berbagai konteks yang relevan, chatbot akan kesulitan memberikan tanggapan yang akurat atau sesuai.
- Keterbatasan dalam Menangani Ambiguitas : Chatbot mungkin mengalami kesulitan dalam menangani ambiguitas atau variasi linguistik dalam input pengguna.

NAMA KELOMPOK : PANDORA