

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА

кафедра інформаційних систем та технологій

ЗВІТ

із лабораторної роботи № 4

з дисципліни “**Технології програмування об’єктів лінгвістичної**
предметної галузі”

на тему: “Автоматична генерація тексту та створення чат-ботів”

Варіант № **2-05**

Виконав:

Студент групи №2

Кличлієв Кирило

Перевірив:

доц. Костіков М. П.

Київ — 2023

Мета лабораторної роботи

Метою лабораторної роботи №4 є отримання базових навичок а автоматичній генерації тексту, а також створенні чатботів. Окремі завдання:

- Створити функцію, що приймає на вхід текст в якості аргументу й визначає залежності між словами за ланцюгом Маркова
- Створити функцію генерації тексту та протестувати її
- Реалізувати та протестувати чатбот, використовуючи бібліотеку ChatterBot
- З'ясувати, які є шляхи практичного застосування ланцюгів Маркова в NLP

Дані з індивідуального варіанту №2-05

Файл: "bronte-eyre.txt"

N: 17

Мова: шведська

Середовище розробки

Мова програмування: Python

Назва та версія IDE: Visual Studio Code 1.78.2

Назва та версія ОС: Linux Mint 21 Vanessa

Мова ОС: англійська

Хід роботи

1. Створюємо новий консольний проєкт мовою Python, що при запуску виводить ім'я, прізвище виконавця роботи і номер лабораторної роботи:

```
print('Кличієв Кирило\nГрупа №2\nЛабораторна робота №4')
```

✓ 0.0s

Кличієв Кирило
Група №2
Лабораторна робота №4

2. Встановлюємо та імпортуємо до проєкту бібліотеки NLTK, ChatterBot (версію 1.0.0) та інші допоміжні/службові бібліотеки:

```
import chatterbot
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer, ChatterBotCorpusTrainer
import nltk
from collections import defaultdict
import collections
import random
collections.Hashable = collections.abc.Hashable

print(
    f'chatterbot version: {chatterbot.__version__}\n'
    f'nltk version: {nltk.__version__}'
)
```

✓ 0.1s Python

chatterbot version: 1.0.0
nltk version: 3.7

3. Створюємо функцію, яка на вхід приймає текст (змінну рядкового типу), визначає залежності між словами за ланцюгом Маркова (для кожного слова – всі слова, які йдуть після нього) та повертає їх як словник (ключ – слово, значення – список наступних слів).

```
def markov_chain(text):
    words = text.split(' ')
    my_dict = defaultdict(list)
    for current_word, next_word in zip(words[0:-1], words[1:]):
        my_dict[current_word].append(next_word)
    my_dict = dict(my_dict)
    return my_dict
```

✓ 0.0s Python

4. Створюємо функцію генерації тексту, яка на вхід як аргумент приймає створений у минулому завданні словник і число слів для генерації, після чого генерує текст заданої довжини з урахуванням залежностей зі словника та виводить його в консоль.

```
def generate_sentence(chain, word_count):
    cur_word = random.choice(list(chain.keys()))
    sentence = cur_word.capitalize()
    for i in range(word_count-1):
        next_word = random.choice(chain[cur_word])
        sentence += ' ' + next_word
        cur_word = next_word
    sentence += '.'
    return sentence
```

✓ 0.0s Python

5. Використовуємо створені функції, задавши для створення словника текстовий файл за індивідуальним варіантом (bronte-eyre.txt), а для генерації тексту – число слів N (17):

```
with open('bronte_eyre.txt') as file:
    text = file.read()
    test_dict = markov_chain(text)

print(f'Згенерований текст: {generate_sentence(test_dict, 17)}')
```

✓ 0.1s

Згенерований текст: Kindred: those of raising me when the hour the habit of their presence in token that had.

6. Реалізовуємо чат-бота з використанням з використанням бібліотеки ChatterBot за зразком із лекції №9: задаємо списки фраз small_talk та math_talk, проводимо навчання бота на цих фразах і перевіряємо його роботу в діалозі:

```
logger = logging.getLogger().setLevel(logging.CRITICAL)
collections.Hashable = collections.abc.Hashable

my_bot = ChatBot(
    name="SwedBot",
    read_only = True,
    logic_adapters=["chatterbot.logic.MathematicalEvaluation",\
        "chatterbot.logic.BestMatch"]
)

small_talk = ["Hej!",
    "Hej",
```

```

        "Hallå",
        "Hur mår du?",
        "Hur står det till?",
        "Jag mår bra",
        "Bra, och du?",
        "Alltid bra",
        "Jag är okej",
        "Skönt att höra det",
        "Jag mår bra",
        "Skönt att höra det",
        "Jag mår fantastiskt",
        "Utmärkt, skönt att höra det",
        "Inte så bra",
        "Tråkigt att höra det",
        "Varifrån kommer du?",
        "Jag kommer från Sverige.",
        "Vad heter du?",
        "Jag är SwedBot. Fråga mig en matematikfråga, snälla."
    ]

math_talk1 = ['Sinus av dubbel vinkel?', '  $\sin(2x)=2\sin x \cos x$ ']

math_talk2 = ['Formel för linjär regression', 'Y=a + bX +  $\epsilon$ ']

math_talk3 = ['Cosinuslikhet', 'Cos(x, y) = x . y / ||x|| * ||y||']

math_talk4 = ['Skriv en formel för en cirkelradie', 'D/2']

math_talk5 = ['Grundläggande sannolikhetsformel', 'P(A) = n(A)/n(S)']

list_trainer = ListTrainer(my_bot)
for item in (small_talk, math_talk1, math_talk2, math_talk3, math_talk4,
math_talk5):
    list_trainer.train(item)

print('\nДиалог з ботом')
for i in range(10):
    user_input= input('User: ')
    print(f'Bot: {my_bot.get_response(user_input)}')
```

Результат роботи коду:

```

List Trainer: [#####] 100%
List Trainer: [#####] 100%
List Trainer: [#####] 100%
List Trainer: [#####] 100%
List Trainer: [#####] 100%
List Trainer: [#####] 100%

Діалог з ботом
User: Hej
Bot: Hej
User: Hur mår du?
Bot: Hur står det till? Jag mår bra
User: Jag är okej
Bot: Skönt att höra det
User: Varifrån kommer du?
Bot: Jag kommer från Sverige.
User: Vad heter du?
Bot: Jag är SwedBot. Fråga mig en matematikfråga, snälla.
User: Cosinuslikhet
Bot:  $\cos(x, y) = x \cdot y / ||x|| * ||y||$ 
User: Skriv en formel för en cirkelradie
Bot:  $D/2$ 
User: Grundläggande sannolikhetsformel
Bot:  $P(A) = n(A)/n(S)$ 
User: Formel för linjär regression
Bot:  $Y = a + bX + \epsilon$ 
User: Sinus av dubbel vinkel?
Bot:  $\sin(2x) = 2\sin x \cos x$ 

```

7. Проводимо навчання чат-бота на корпусі мовою за індивідуальним варіантом (шведська мова) та аналогічно перевіряємо його роботу в діалозі (мінімум – 10 фраз користувача).

```

logger = logging.getLogger().setLevel(logging.CRITICAL)
collections.Hashable = collections.abc.Hashable

my_bot = ChatBot('SwedBot')

corpus_trainer = ChatterBotCorpusTrainer(my_bot)
corpus_trainer.train('chatterbot.corpus.swedish')

print('\nДіалог з ботом')
for i in range(10):
    user_input = input('User: ')
    print(f'Bot: {my_bot.get_response(user_input)}')

```

Результати роботи коду (два діалоги з ботом, натренованому на корпусі шведської мови):

```
return yaml.load(data_file)

Training ai.yml: [#####] 100%
Training conversations.yml: [#####] 100%
Training food.yml: [#####] 100%
Training greetings.yml: [#####] 100%
Training sports.yml: [#####] 100%

Діалог з ботом
User: Hej
Bot: Hej
User: Hur mår du?
Bot: De är mitt namn
User: Vad heter du?
Bot: De är mitt namn
User: Är "de" ditt pronomen?
Bot: Vad är ai
User: Jag tror att du vet bättre
Bot: De är mitt namn
User: Det vet jag redan. Och mitt namn är Kyrylo. Trevligt att träffas
Bot: python
User: Vad menar du med
Bot: I'm also good.
User: Jag förstår inte det här språket, tyvärr
Bot: Golf
User: Jag förstår dig inte. Kan du säga varifrån du kommer?
Bot: That is good to hear
User: Skriv på svenska
Bot: In the face of ambiguity, refuse the temptation to guess.
klychiieafx@svarthamar:~/Desktop/lab-4$ █
```

```
Training ai.yml: [#####] 100%
Training conversations.yml: [#####] 100%
Training food.yml: [#####] 100%
Training greetings.yml: [#####] 100%
Training sports.yml: [#####] 100%

Діалог з ботом
User: Hej
Bot: Hej
User: Hur står det till?
Bot: Tell me about your self.
User: svenska please
Bot: Vad är ai
User: ett intressant område att utforska, eller hur?
Bot: De är mitt namn
User: Trevligt att träffas! Var kommer du ifrån?
Bot: De är mitt namn
User: Varifrån kommer du?
Bot: De är mitt namn
User: ok
Bot: Att en och samma spelare gör 3 mål under samma match och helst under samma halvlek/period.
User: Vad pratar du om?
Bot: What good news?
User: svenska please
Bot: Vad är ai
User: Du är AI
Bot: AI står för Artificial Intelligence och betyder att ett antal utvecklare har skapat en maskin som kan tänka på själv
klychiieafx@svarthamar:~/Desktop/lab-4$ █
```

7. Написати у звіті відповідь на питання “Варіанти (приклади) застосування моделі Маркова в лінгвістиці.”

Ланцюги Маркова та моделі, створені спираючись на цю математичну теорію, широко використовуються в області штучного інтелекту, зокрема й в наступних напрямках NLP:

1. Машинний переклад. Модель Маркова може використовуватися при побудові систем статистичного машинного перекладу. Вона дозволяє враховувати залежності між словами та фразами у вхідній (source) та цільовій (target) мовах, а також імовірності перекладу на основі статистичних даних
2. Автодоповнення тексту та виправлення помилок. Модель, створена на основі ланцюгів Маркова, може моделювати ймовірність появи наступного слова на основі контексту, що допомагає пропонувати правильні варіанти заповнення чи виправлення тексту
3. Генерація тексту. Ланцюги Маркова можна використовувати для генерації нового тексту на основі заданого корпусу. Аналізуючи ймовірності переходу між словами або n-грамами в навчальних даних, можна побудувати ланцюг Маркова для генерації зв'язних і контекстуально релевантних речень.
4. Розпізнавання мови. Моделі Маркова, зокрема НММ (hidden Markov model), широко використовуються в системах автоматичного розпізнавання мови. Моделі фіксують імовірнісні зв'язки між фонемами або акустичними ознаками в мовному сигналі. Оцінюючи найімовірнішу послідовність фонем або слів на основі спостережуваних акустичних даних, НММ можуть перетворювати розмовну мову в текст.

Інші задачі, що можуть бути вирішені за допомогою ланцюгів Маркова: виявлення іменованих сутностей (NER - named entity recognition), частиномовна розмітка (POS tagging).

Висновки

Отже, у ході виконанні лабораторної роботи №4 було розглянуто базові шляхи автоматичної генерації текстів, використовуючи ланцюги Маркова, а також алгоритм створення чатботів за допомогою бібліотеки chatterbot. Було створено дві функції:

1. Перша приймає текст в якості аргументу і визначає залежності між словами за моделлю Маркова
2. Друга на основі визначених лексичних зв'язків у тексті генерує новий текст

Було реалізовано простенький чатбот для шведської мови, що може вести примітивний діалог.

Також було з'ясовано, що головними напрямками в NLP, в яких практично застосовують ланцюги Маркова, є машинний переклад, автодоповнення тексту, виправлення помилок, генерація тексту, розпізнавання мови, виявлення іменованих сутностей і частиномовна розмітка.