Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет інфокомунікацій
(повна назва)

Кафедра інформаційно-мережної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Вилучення тексту з Інтернету на основі навчання машин ML
(тема)

Виконав:
студент 2 курсу, групи ІМІм-21-2
Шалатов В.О.
(прізвище, ініціали)

Спеціальність 172 «Телекомунікації та радіотехніка»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма «Інформаційно-мережна інженерія»
(повна назва освітньої програми)

Керівник доц. Кривенко С.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Безрук В.М.
(підпис) (прізвище, ініціали)

2023 р.

Не містить відомостей, заборонених до відкритого публікування

Студент _____

Керівник_____

Харківський національний університет радіоелектроніки

Факультет _____інфокомунікацій_____
(повна назва)

Кафедра _____інформаційно-мережної інженерії_____
(повна назва)

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність _____172 «Телекомунікації та радіотехніка»_____
(код і повна назва)

Тип програми _____освітньо-наукова_____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____«Інформаційно-мережна інженерія»_____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___»_____20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____Шалатову Василю Олеговичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи Вилучення тексту з Інтернету на основі навчання машин ML
_____
_____
затверджена наказом університету від 17 березня 2023р. № 275 Ст_
2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2023р.
3. Вихідні дані до роботи: _____
Провести аналіз доступних сервісів хмарних обчислень для застосування в технологіях навчання машин. Основні послуги, що надаються хмарними системами. Мови реалізації моделі – Python3, HTML.
4. Перелік питань, що потрібно опрацювати в роботі: _____
навчання машин: основні поняття та методи;
обробка природної мови NLP;
методи вилучення тексту з інтернету;
модель використання Beautiful Soup;
результати роботи моделі.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) 16 слайдів у форматі PowerPoint _____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	доц. Кривенко С.А.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	*Ознайомлення із завданням. Уточнення ТЗ.*	*21.03*	вик
2	*Підбір літератури за темою роботи.*	*21.03-28.03*	вик
3	*Виконання розділу 1*	*29.03-04.04*	вик
4	*Виконання розділу 2*	*05.04-11.04*	вик
5	*Виконання розділу 3*	*12.04-28.04*	вик
6	*Виконання розділів 4, 5*	*29.04-05.05*	вик
7	*Оформлення презентаційного матеріалу, підготовка до захисту у ЕК*	*06.05-18.05*	вик

Дата видачі завдання 14 березня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Кривенко С.А
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка випускної магістерської кваліфікаційної роботи містить: 94 стор., 64 рис., 25 джерел.

ОБРОБКА ПРИРОДНЬОЇ МОВИ, МАШИННЕ НАВЧАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, AWS.

Об’єкт дослідження – модель одного з етапів конвеєру навчання машин стосовно обробки природної мови NLP.

Предмет дослідження – дослідження інструментів зменшення долі недиференційованих важких робіт при вилученні тексту з інтернету, які насправді не допомагають бізнесу зосередитися на їхній основній цінності для клієнтів, але забирають у них багато часу та ресурсів.

Мета роботи – формулювання задачі обробки природної мови в загальному вигляді, аналіз сучасних методів вилучення тексту з інтернету, розробка моделі одного з етапів конвеєру навчання машин, оцінка результатів роботи моделі.

Методи досліджень – методи сервісів AWS.

ABSTRACT

The explanatory note of the final master's qualification work contains: 94 pages, 64 figures, 25 sources.

NATURAL LANGUAGE PROCESSING, MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, AWS.

The object of research is a model of one of the stages of machine learning pipeline related to natural language processing (NLP).

The subject of research – the investigation of tools to reduce the amount of undifferentiated heavy work in web scraping, which do not actually contribute to the business's core value for customers but consume significant time and resources.

The purpose of the work is formulating the natural language processing task in a general form, analyzing contemporary methods of web scraping, developing a model for one of the stages of the machine learning pipeline, and evaluating the model's performance.

Research methods – AWS service methods.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

AGI – Artificial General Intelligence – сильний штучний інтелект

AI – Artificial Intelligence – штучний інтелект

API – Application Programming Interface – інтерфейс прикладного програмування

AWS – Amazon Web Services – дочірня компанія Amazon.com, що надає платформу хмарних обчислень в оренду приватним особам, компаніям та урядам на основі платної підписки

HTML – HyperText Markup Language – мова розмітки гіпертексту

ML – Machine Learning –навчання машин

NLP – Natural Language Processing – обробка природної мови

URL – Uniform Resource Locator – єдиний вказівник на ресурс

TF-IDF – Term Frequency and Inverse Document Frequency – частота термінів та інверсна частота документа

ВСТУП

Актуальною темою навчання машин ML є обробка природної мови NLP. На сьогоднішній день обробка природної мови (NLP) є темою, яка досліджується та розробляється останнє десятиліття. Здатність аналізувати, обчислювати та розуміти просту людську мову за допомогою комп'ютерів і навчання машин була впроваджена в різні програми та платформи. Обробка природної мови, використовує обчислювальні алгоритми для автоматичного аналізу та інтерпретації людської мови. Завдяки оцінці структури мови, системи навчання машин ML здатні обробляти великі обсяги слів, фраз і речень.

В роботі розглянуто можливості розробленої на кафедрі моделі одного з етапів конвеєру навчання машин стосовно обробки природної мови NLP.

Мета роботи полягає у дослідженні інструментів зменшення долі недиференційованих важких робіт при вилученні тексту з інтернету, які насправді не допомагають бізнесу зосередитися на їхній основній цінності для клієнтів, але забирають у них багато часу та ресурсів.

У першому розділі розглянуті основні понятя та методі машинного навчання. Другий розділ присвячено формулюванню задачі обробки природної мови в загальному вигляді. В третьому розділі аналізуються сучасні методи вилучення тексту з інтернету. Розробка моделі одного з етапів конвеєру навчання машин є предметом четвертого розділу. Оцінка результатів роботи моделі представлена в п'ятому розділі.

Показано ефективність застосування мови програмування Python для розв'язання задач обробки природної мови, аналізу сучасних методів вилучення тексту з інтернету, розробки моделі конвеєру навчання машин, оцінки результатів роботи моделі.

Метою кваліфікаційної роботи є дослідження інструментів зменшення долі недиференційованих важких робіт при вилученні тексту з Інтернету, які насправді

не допомагають бізнесу зосередитися на їхній основній цінності для клієнтів, але забирають у них багато часу та ресурсів.

Завдання роботи полягає у формулюванні задачі обробки природної мови в загальному вигляді, аналізі сучасних методів вилучення тексту з Інтернету, розробці моделі одного з етапів конвеєра навчання машин та оцінці результатів роботи моделі.

Кваліфікаційна робота складається з п’яти розділів, 13 рисунків, 3 додатків та 23 джерел літератури.

1 НАВЧАННЯ МАШИН: ОСНОВНІ ПОНЯТТЯ ТА МЕТОДИ

Навчання машин (Machine Learning) є однією з найбільш широко використовуваних галузей науки в сучасному світі. Воно забезпечує можливість комп'ютерам навчатися на основі даних, замість того, щоб бути просто програмованими заздалегідь. У цьому тексті розглядаються основні поняття та методи навчання машин.

Основні поняття навчання машин:

Дані: навчання машин базується на аналізі та обробці даних, які можуть бути представлені у вигляді чисел, тексту, зображень та інших форматів. Дані можуть бути зібрані з різних джерел, таких як бази даних, веб-сторінки, сенсори, смартфони та інші джерела.

Модель: модель - це математичне вираження, яке описує залежність між вхідними даними та вихідними результатами. Наприклад, модель може передбачити, який продукт буде куплений користувачем на основі його попередніх покупок та інших даних.

Навчання: навчання - це процес, під час якого модель навчається на основі набору даних, з метою покращення точності її прогнозування. Під час навчання, модель знаходить залежності між вхідними даними та вихідними результатами, що дозволяє їй зробити кращі прогнози на нових даних. Тестування: тестування - це процес перевірки точності прогнозування моделі на нових, раніше не використовуваних даних. Це допомагає визначити, наскільки добре модель виконує своє завдання.

Основні методи навчання машин:

Навчання з учителем: цей метод використовується для розв'язання задач класифікації та регресії. Модель навчається на основі пари вхідних даних та вихідного результату. Наприклад, якщо потрібно передбачити ціну на нерухомість на основі її параметрів, модель буде навчатися на основі пари "параметри нерухомості" та "ціна".

Навчання без учителя: цей метод використовується для задач кластеризації та зменшення розмірності даних. Він базується на тому, щоб знайти схожість між різними елементами даних та згрупувати їх у кластери. Наприклад, якщо потрібно згрупувати користувачів на основі їх поведінки на сайті, модель буде навчатися на основі схожості між їх діями та інших факторів.

Навчання з підкріпленням: цей метод використовується для навчання моделей, які мають здатність приймати рішення та взаємодіяти з навколишнім середовищем. Модель навчається на основі того, які дії повинна виконувати, щоб максимізувати деякий показник успішності. Наприклад, якщо потрібно навчити робота виконувати певну задачу, модель буде навчатися на основі того, які дії приводять до успішного виконання цієї задачі.

Оптимізація: оптимізація - це процес пошуку параметрів моделі, які максимізують її точність на тестових даних. Цей процес може використовувати різні алгоритми, такі як градієнтний спуск, щоб знайти найкращі параметри. Ці поняття та методи навчання машин є основними, і їх застосовують в різних сферах, таких як бізнес, медицина, транспорт, наука та багато інших. Навчання машин стає все більш популярним, оскільки дозволяє отримувати більш точні та швидкі результати в порівнянні з традиційними методами аналізу даних.

1.1 Алгоритми класифікації

Алгоритми класифікації є одним з найпоширеніших методів для вилучення текстової інформації з Інтернету на основі навчання машин. Ці алгоритми можуть використовуватися для класифікації текстів на підставі їхньої тематики або наявності певних ключових слів. Одним з прикладів таких алгоритмів є наївний Байесівський класифікатор, який використовується для віднесення текстів до певних категорій на основі ймовірностей входження певних слів у текст. Іншим прикладом алгоритмів класифікації є алгоритм к-найближчих сусідів, який використовується для визначення класу нового тексту на основі аналізу схожості

його з уже існуючими текстами у навчальній вибірці. Цей алгоритм може бути використаний для автоматичного розпізнавання тексту на підставі його вмісту та структури та генерації додаткового тексту на основі цього.

Одним з найбільш ефективних методів застосування алгоритмів класифікації для вилучення інформації з Інтернету є розпізнавання спаму. Це може бути використано для фільтрації небажаних електронних листів, коментарів та повідомлень на веб-сайтах. Алгоритми класифікації також можуть бути використані для розпізнавання емоцій, що відображені у тексті, які можуть бути важливими для аналізу відгуків користувачів та вимог споживачів. Іншим методом використання алгоритмів класифікації є визначення авторства тексту, що дозволяє відрізняти текст, написаний однією особою, від тексту, написаного іншою. Це може бути корисно для визначення авторства новин або інформаційних повідомлень. Алгоритми класифікації також можуть бути використані для автоматичного розпізнавання мови, що дозволяє автоматично визначати мову тексту та перекладати його на інші мови. Це може бути корисно для міжнародних компаній та веб-сайтів, які мають користувачів з різних країн та мов.

Узагальнюючи, алгоритми класифікації є потужним інструментом для вилучення текстової інформації з Інтернету на основі навчання машин. Вони можуть бути використані для класифікації текстів на підставі їхньої тематики, визначення авторства тексту, визначення мови тексту та розпізнавання спаму. Для досягнення найкращих результатів, важливо вибрати підходящий алгоритм та налаштувати його параметри відповідно до потреб користувача.

1.2 Зведення до ключових слів

Зведення до ключових слів - це метод вилучення інформації з текстів, який полягає в ідентифікації та виділенні ключових слів, які найкраще описують зміст тексту. Цей метод може бути застосований до будь-якого типу тексту, від простих новин до складних наукових статей. Одним з найпоширеніших алгоритмів

зведення до ключових слів є TextRank, який використовує графову модель для визначення важливості слів у тексті. Цей алгоритм оцінює вагу кожного слова на основі його взаємодії з іншими словами у тексті, що дозволяє виділити ключові слова, які найкраще описують зміст тексту. Іншими словами, слова, які найбільш інформативні та сильно пов'язані з контекстом тексту.

Існують інші алгоритми зведення до ключових слів, такі як TF-IDF та RAKE. Алгоритм TF-IDF оцінює важливість слова на основі частоти його вживання у тексті та величини інверсії документної частоти (IDF), яка відображає, наскільки рідкісним є це слово у текстах корпусу загалом. RAKE, з іншого боку, використовує алгоритм розбиття тексту на фрази та визначення їхньої ваги на основі частоти вживання та наявності важливих слів у них.

Зведення до ключових слів може бути корисним методом для швидкого збіру та аналізу інформації з Інтернету. Наприклад, цей метод може бути використаний для автоматичного створення опису статей або категоризації веб-сторінок за ключовими словами. Крім того, зведення до ключових слів може бути корисним для знаходження новин та трендів у певній тематиці на основі аналізу ключових слів, які найчастіше зустрічаються у пов'язаних з цією тематикою текстах.

Ще одним застосуванням зведення до ключових слів є поліпшення пошукових запитів. Користувачі Інтернету часто вводять запити у пошукових системах, що містять занадто багато слів або неясних понять, що ускладнює пошук потрібної інформації. За допомогою зведення до ключових слів, можна автоматично створювати більш точні та специфічні запити, що збільшує ефективність пошуку.

Зведення до ключових слів також може бути корисним для аналізу тексту в академічних дослідженнях та наукових статтях. Для великих текстів, наприклад, дисертацій та наукових звітів, зведення до ключових слів може допомогти авторам швидше зорієнтуватися в змісті своїх робіт та зосередитися на найважливіших аспектах дослідження.

Окрім того, зведення до ключових слів може бути корисним для маркетингу та SEO (пошукової оптимізації) в Інтернеті. Ключові слова можуть бути використані для підвищення видимості веб-сторінок у пошукових системах, а також для привертання уваги потенційних клієнтів до продуктів або послуг.

Однак, важливо зазначити, що зведення до ключових слів не є повноцінним замінником для ручного аналізу тексту. Алгоритми зведення до ключових слів можуть пропустити важливі деталі та нюанси тексту, які можуть бути виявлені тільки в процесі ретельного прочитання. Тому, зведення до ключових слів можна використовувати як допоміжний інструмент для аналізу тексту, але не як єдиний джерело інформації.

1.3 Аналіз методів обробки природної мови(NLP)

Розпізнавання мови є однією з ключових задач у сфері обробки природної мови. Цей метод має на меті розуміти та інтерпретувати людську мову, яка може бути дуже складною через наявність синонімів, амбігвітності та інших факторів. Для досягнення цієї мети використовуються алгоритми навчання машин, які навчають моделі розуміти та інтерпретувати мову. Такі моделі можуть бути використані для перекладу мов, створення голосових помічників, аналізу соціальних мереж та багато іншого.

Класифікація текстів - це метод обробки природної мови, який використовується для автоматичної класифікації текстів на певні категорії. Для досягнення цієї мети використовуються алгоритми навчання машин, які навчають моделі класифікувати тексти з використанням навчальних даних. Класифікація текстів може бути використана для автоматичного сортування електронної пошти, аналізу настроїв відгуків користувачів та багато іншого.

Екстракція інформації - це метод обробки природної мови, який використовується для автоматичного виділення корисної інформації з текстів. Цей метод може бути використаний для виділення імен, дат, адрес та іншої інформації

з текстів. Для досягнення цієї мети використовуються алгоритми навчання машин, які навчають моделі виділяти корисну інформацію з текстів.

Розпізнавання мови є однією з ключових задач у сфері обробки природної мови. Цей метод має на меті розуміти та інтерпретувати людську мову, яка може бути дуже складною через наявність синонімів, амбігвітності та інших факторів. Для досягнення цієї мети використовуються алгоритми навчання машин, які навчають моделі розуміти та інтерпретувати мову. Такі моделі можуть бути використані для перекладу мов, створення голосових помічників, аналізу соціальних мереж та багато іншого.

Розпізнавання іменованих сутностей є методом обробки природної мови, який використовується для виявлення та виділення іменованих сутностей з тексту. Цей метод дозволяє автоматично визначати та класифікувати різні типи іменованих сутностей, такі як люди, місця, організації, події тощо. Для досягнення цієї мети використовуються алгоритми навчання машин, які навчають моделі розпізнавати та класифікувати іменовані сутності з використанням навчальних даних. Розпізнавання іменованих сутностей може бути використане для багатьох завдань, таких як аналіз новин, класифікація документів, підготовка даних для пошукових систем та багато іншого. Наприклад, цей метод може бути використаний для визначення того, хто є автором статті, яка містить інформацію про деякий продукт. Також, розпізнавання іменованих сутностей може бути використане для створення автоматичних тезаурусів та визначення популярних тем у текстах.

Сумаризація тексту - це метод обробки природної мови, який використовується для автоматичного стиснення текстів з метою виділення найважливішої інформації. Цей метод може бути використаний для створення коротких оглядів новин, резюме та іншої інформації з великої кількості тексту. Для досягнення цієї мети використовуються алгоритми навчання машин, які навчають моделі зводити текст до короткого та зрозумілого формату.

Сумаризація тексту може бути використана для багатьох завдань, таких як автоматичне створення коротких оглядів новин для швидкого перегляду,

автоматичне стиснення документів для зручності читання та збереження часу, створення резюме для відбору кандидатів на роботу та багато іншого. Крім того, сумаризація тексту може бути використана для автоматичного створення заголовків та описів для веб-сторінок та інших матеріалів, що допомагає збільшити їх привабливість та ефективність.

1.4 Метод навчання з вчителем

Метод навчання з вчителем для вилучення тексту з Інтернету складається з декількох етапів.

Першим етапом є збір та підготовка навчальних даних. Це може включати в себе збір веб-сторінок, які містять необхідну інформацію, а також підготовку цих даних для використання в навчанні. Наприклад, це може включати в себе збір тексту з HTML-коду веб-сторінок та очищення від зайвих тегів та символів.

Другий етап - це побудова моделі навчання машин для вилучення тексту. Для цього можна використовувати алгоритми навчання машин, такі як різні види нейронних мереж, алгоритми класифікації та кластеризації даних, та інші. Під час цього етапу, модель навчається розпізнавати та виділяти інформацію з тексту веб-сторінок.

Третій етап - це тестування та налаштування моделі. Під час цього етапу модель тестується на нових навчальних даних, щоб оцінити її точність та ефективність. Якщо модель потребує додаткової настройки, цей етап дозволяє внести необхідні зміни до алгоритмів та параметрів моделі.

Останній етап включає в себе використання навченої моделі для вилучення потрібної інформації з веб-сторінок. Модель може бути інтегрована в програму або сервіс, який забезпечує автоматичне вилучення тексту з веб-сторінок. Користувачі можуть задати критерії для вилучення тексту, такі як ключові слова, регулярні вирази або інші параметри. Після того, як модель зібрала необхідну інформацію з веб-сторінок, вона може бути оброблена та збережена у відповідному форматі,

наприклад, у текстовому файлі або базі даних. Ця інформація може бути використана для різноманітних цілей, таких як аналіз даних, збір статистики, пошук інформації, моніторинг веб-сайтів та інше.

Завдяки використанню навчання машин та методу навчання з вчителем, процес вилучення тексту з Інтернету може бути значно автоматизований та оптимізований. Це дозволяє збільшити ефективність та точність процесу та скоротити час, який потрібен для збору інформації з Інтернету.

1.5 Метод навчання без вчителя

Метод навчання без вчителя для вилучення тексту з Інтернету (unsupervised learning for web text extraction) - це підхід до автоматичного збору даних з Інтернету, який не потребує навчальних даних. В основі методу лежить використання алгоритмів навчання машин, які здатні аналізувати веб-сторінки та вилучати з них необхідну інформацію. Основна ідея методу полягає в тому, щоб використовувати статистичні аналізи та алгоритми класифікації для визначення того, які частини веб-сторінок містять корисну інформацію, а які - ні. Наприклад, для збору даних з сайту можна використовувати алгоритми кластеризації. У цьому методі спочатку збирається набір текстів, з якого потрібно вилучити інформацію. Далі, за допомогою алгоритмів кластеризації (наприклад, K-means), ці тексти розділяються на декілька кластерів. Після цього в кожному кластері можна виділити найбільш типові для нього слова і фрази, що вказують на тематику цього кластеру. Наприклад, якщо в одному кластері багато текстів про футбол, то найчастіше зустрічаються слова "гол", "команда", "чемпіонат" і т.д.

Далі, за допомогою методу "пошуку ключових слів", можна знайти у кожному кластері найбільш важливі для нього слова і фрази. Наприклад, якщо в одному кластері найчастіше зустрічається слово "комп'ютер", то ключовими словами для цього кластеру можуть бути "програмне забезпечення", "процесор", "оперативна пам'ять" і т.д. За допомогою отриманих ключових слів можна створити

правила для вилучення тексту з Інтернету. Наприклад, якщо потрібно вилучити текст про комп'ютери, то можна шукати текст, в якому зустрічаються ключові слова для кластера з темою "комп'ютери".

Отже, метод кластеризації та пошуку ключових слів може бути ефективним для вилучення тексту з Інтернету без використання вчителя. Однак, слід зазначити, що цей метод може бути недостатньо точним, іноді вилучати зайву інформацію, або пропускати важливі дані.

Окрім методу кластеризації та пошуку ключових слів, існують і інші методи навчання без вчителя для вилучення тексту з Інтернету. Наприклад, можна використовувати методи тематичного моделювання, які дозволяють виявляти теми, які присутні в текстах, і виділяти ключові слова для кожної теми.

Методи тематичного моделювання - це група алгоритмів ного навчання машин, які дозволяють виявляти теми, які присутні в текстах, і виділяти ключові слова для кожної теми. Ці методи є важливим інструментом для аналізу текстових даних, таких як статті, блоги, соціальні медіа та інші джерела. Один з найпоширеніших методів тематичного моделювання - це Latent Dirichlet Allocation (LDA). LDA є ймовірнісною моделлю, яка дозволяє виявляти теми в наборі текстових документів. Основна ідея LDA полягає в тому, що кожен документ можна розглядати як комбінацію декількох тем, причому кожна тема містить певну кількість слів, які в ній найчастіше зустрічаються. Завдання LDA полягає в тому, щоб визначити, які теми є присутніми в колекції документів, і які слова є ключовими для кожної теми. Інший метод тематичного моделювання - це Non-negative Matrix Factorization (NMF). NMF є матричним методом, який дозволяє виділяти теми з матриці термів-документів. У NMF кожен документ можна розглядати як комбінацію декількох тем, а кожна тема представляється як вектор, що складається з ключових слів, що характеризують тему. Завдання NMF полягає в тому, щоб знайти матриці тем та ключових слів, які дозволять найкраще описати дані. Інші методи тематичного моделювання включають Probabilistic Latent Semantic Analysis (PLSA), Hierarchical Dirichlet Process (HDP), Structural Topic

Models (STM) та інші. Кожен з цих методів має свої переваги та недоліки і може бути використаний в залежності від конкретного завдання та набору даних.

Крім вищезгаданих методів тематичного моделювання, існують й інші підходи до аналізу тем у текстах. Наприклад, Topic Modeling based on Evolutionary Multi-objective Optimization (TME-MO) використовує методи оптимізації для пошуку найкращих тематичних моделей, що забезпечують баланс між точністю та складністю моделі. Також існує метод Topic Over Time (TOT), який дозволяє виявляти зміни тем у текстових колекціях з плином часу. Крім того, використання тематичного моделювання можна комбінувати з іншими методами аналізу текстових даних, наприклад, з аналізом емоцій або з кластеризацією документів на основі їхніх тематик. В цілому, методи тематичного моделювання дозволяють автоматизувати процес аналізу текстових даних та забезпечують можливість відкривати нові зв'язки та підходи до їхнього вивчення.

Також можна використовувати методи зведення розмірності, які дозволяють зменшити кількість ознак у текстах, що сприяє більш ефективному використанню алгоритмів класифікації.

Методи зведення розмірності - це група алгоритмів навчання машин, які дозволяють зменшити кількість ознак у текстах з метою поліпшення ефективності алгоритмів класифікації. Зазвичай, у текстах кількість ознак досить велика, що може призвести до перенавчання (overfitting) та низької точності класифікації. Зведення розмірності дозволяє зменшити кількість ознак, що полегшує класифікацію та покращує результати. Один з найпоширеніших методів зведення розмірності - це Principal Component Analysis (PCA). PCA є статистичним методом, який дозволяє знайти нові ознаки, які є лінійними комбінаціями початкових ознак, із збереженням максимальної дисперсії. Основна ідея PCA полягає в тому, що велика частина інформації в даних може бути описана за допомогою меншої кількості ознак. Завдання PCA полягає в тому, щоб знайти ці нові ознаки, які дозволять найкраще описати дані. Інший метод зведення розмірності - це t-SNE (t-Distributed Stochastic Neighbor Embedding). t-SNE є нелинійним методом, який

дозволяє знайти нові ознаки, які візуалізують дані в низькорозмірному просторі зберігаючи відносної близькість між об'єктами. t-SNE дозволяє зображувати дані у двох або трьох вимірах, що полегшує їх візуалізацію та дозволяє здійснювати класифікацію на основі візуальної інтерпретації даних.

Крім того, існують такі методи зведення розмірності, як Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), Factor Analysis (FA) та інші. Кожен з цих методів має свої переваги та недоліки, і їх вибір залежить від конкретної задачі та даних, з якими має справу дослідник. LDA - це метод зведення розмірності, який використовується для знаходження нових ознак, що максимально розділяють класи. LDA використовується для знаходження нових ознак шляхом проектування даних на площину з меншою розмірністю, яка максимально розділяє класи. Це досягається шляхом знаходження власних векторів матриці розподілу даних, які відповідають найбільш значущим напрямкам розділення класів. Нові ознаки обираються згідно зі значеннями дискримінантної функції, яка визначає, наскільки добре окремі класи розділяються в проекції даних. Основна ідея полягає в тому, щоб знайти нові ознаки, які дозволять максимально розділити класи в даних. LDA є корисним методом для класифікації, оскільки дозволяє знайти нові ознаки, які мають максимальний роздільний потенціал. LDA використовується не тільки для зведення розмірності, але і для покращення роздільної здатності моделей класифікації, зокрема, для зменшення перенавчання (overfitting) та покращення універсальності моделі. LDA широко використовується в біоінформатиці, медичній діагностиці та інших галузях, де важливо класифікувати об'єкти за допомогою важливих ознак. Один з головних недоліків LDA полягає в тому, що він працює тільки для задач з дискретними класами, тобто для задач класифікації. Крім того, якщо кількість ознак вихідних даних велика, то LDA може бути дуже витратним з точки зору обчислювальних ресурсів.

ICA - це метод зведення розмірності, який дозволяє знайти незалежні компоненти в даних. ICA є корисним методом, якщо вихідні ознаки корельовані між собою та/або мають складні залежності. Основна ідея полягає в тому, щоб

знайти нові ознаки, які є незалежними від початкових ознак. Один з основних застосувань ICA полягає в розрізненні джерел. Наприклад, у задачах обробки звуку ICA може допомогти виділити різні голоси або звукові ефекти, що містяться в змішаному сигналі. У фінансовій аналітиці ICA можна використовувати для виділення різних факторів, що впливають на ціни акцій. Недоліком ICA є те, що він припускає, що незалежність між ознаками є лінійною, тобто можуть існувати складні залежності між ознаками, які не можуть бути виявлені цим методом. Також ICA може бути відчутливим до шуму в даних, що може призвести до появи спотворень у незалежних компонентах.

FA - це метод зведення розмірності, який дозволяє знайти латентні змінні, які пояснюють спільну варіацію в даних. FA є корисним методом, якщо ознаки взаємозв'язані та залежать від деяких латентних змінних. Основна ідея полягає в тому, щоб знайти нові ознаки, які відображають латентні змінні, які пояснюють спільну варіацію в даних. Це може бути корисно, наприклад, в задачах, пов'язаних з аналізом соціальних мереж або оцінкою ментального здоров'я. У таких випадках ознаки можуть бути пов'язані з різними аспектами поведінки або емоцій, але за допомогою FA можна знайти латентні змінні, які є спільними для багатьох ознак. FA може бути використаний для визначення кількості латентних змінних, які необхідні для пояснення певного відсотка варіації в даних. Крім того, FA може бути застосований для виявлення груп ознак, які взаємозв'язані між собою та можуть бути пов'язані зі специфічними аспектами даних. Узагальнюючи, FA - це корисний метод зведення розмірності, який може допомогти виявити латентні змінні, які пояснюють спільну варіацію в даних та дозволяють замінити велику кількість початкових ознак меншою кількістю латентних змінних.

Кожен з цих методів має свої переваги та недоліки, і їх вибір залежить від конкретної задачі та даних, з якими має справу дослідник. Отже, хоча LDA, ICA та FA є корисними методами зведення розмірності, вони також мають свої недоліки та обмеження. Наприклад, LDA не працює добре, якщо класи перетинаються або якщо кількість ознак значно більше, ніж кількість спостережень. ICA може бути

обмеженим, якщо деякі незалежні компоненти не є суттєвими для аналізу даних. FA може бути непрацездатним, якщо немає латентних змінних, які пояснюють велику частину варіації в даних. Окрім цього, важливо мати на увазі, що зведення розмірності може призвести до втрати інформації. Тому, перед використанням будь-якого методу зведення розмірності, необхідно ретельно проаналізувати дані та вибрати метод, який найкраще підходить для конкретної задачі, має найменші недоліки та зберігає найбільшу кількість інформації.

Важливо зазначити, що методи навчання без вчителя для вилучення тексту з Інтернету мають свої обмеження та використовуються з обережністю. Наприклад, вони можуть бути неефективними в тих випадках, коли дані, які потрібно вилучати, знаходяться в складних форматах, таких як відео або зображення. Також вони можуть бути недостатньо точними в тих випадках, коли текст містить складну граматику або ідіоматичні вирази. Оскільки методи навчання без вчителя для вилучення тексту з Інтернету ґрунтуються на статистичному аналізі, вони не завжди можуть правильно розуміти семантику тексту, що може призвести до неправильного вилучення або пропуску деякої інформації. У загальному, метод навчання без вчителя для вилучення тексту з Інтернету є корисним інструментом для автоматичного збору даних з Інтернету, але перед його використанням слід ретельно проаналізувати його можливості та обмеження, і дотримуватися належних процедур перевірки та перевірки якості.

1.6 Навчання з підкріпленням

Навчання з підкріпленням є однією з найбільш захоплюючих та перспективних галузей навчання машин, оскільки вона дозволяє машинам вчитися на основі взаємодії з навколишнім середовищем, подібно до того, як люди вчаться через взаємодію з реальним світом. У навчанні з підкріпленням агент здійснює дії в середовищі і отримує винагороду або штраф, залежно від того, наскільки його дії призводять до досягнення поставленої мети. Цей вид навчання можна

використовувати в різних областях, таких як автономна навігація, керування роботами, управління виробництвом та багато іншого. У процесі навчання з підкріпленням агент збирає досвід, який потім використовується для вдосконалення його поведінки. Для цього використовуються різні методи, такі як Q-навчання, Policy Gradient та Actor-Critic. Кожен з цих методів має свої переваги та недоліки, тому вибір методу залежить від конкретної задачі та умов навчання. Навчання з підкріпленням є потужним інструментом для навчання машин здійснювати рішення в реальному світі. Воно дозволяє агенту вчитися самостійно і здобувати досвід, який може бути використаний для розв'язання нових задач в майбутньому. Одним з ключових елементів навчання з підкріпленням є визначення функції винагороди. Ця функція повинна визначати, які дії агента є правильними та призводять до досягнення поставленої мети. Це може бути дуже складно, оскільки функція винагороди повинна враховувати багато факторів, таких як час, кількість взаємодій з середовищем та багато іншого. Важливо також забезпечити баланс між винагородою та штрафом, щоб агент не зосереджувався тільки на отриманні винагороди та не ігнорував можливі негативні наслідки своїх дій. Іншим важливим аспектом навчання з підкріпленням є здатність агента до узагальнення. Це означає, що агент повинен вміти застосовувати набуті знання та досвід до нових ситуацій, які він не зустрічав раніше. Для цього використовуються різні методи, такі як навчання на рівномірно випадкових діях та навчання на прикладах. Одним з переваг навчання з підкріпленням є його здатність до самонавчання. Агент може навчитися вирішувати складні задачі, які люди можуть вирішувати тільки за допомогою багатьох років навчання та досвіду. Наприклад, навчання з підкріпленням використовується для навчання автономних автомобілів, де агент повинен приймати рішення в реальному часі на основі даних з різних датчиків та камер. Однак, навчання з підкріпленням має свої недоліки. Один з найбільш очевидних недоліків полягає у високих вимогах до обчислювальної потужності та часу. Навчання з підкріпленням може вимагати великої кількості обчислювальних ресурсів та тривати дуже довго. Крім того, іншим недоліком є можливість навчання

агента до небажаної поведінки. Це може статися, якщо функція винагороди неправильно налаштована або якщо навчальні дані містять неадекватні або небажані приклади. Ще одним важливим аспектом навчання з підкріпленням є етика. Навчання з підкріпленням може використовуватися для створення автономних систем, які мають великий вплив на життя людей. Тому важливо враховувати етичні аспекти та відповідальність при розробці та застосуванні таких систем. У підсумку, навчання з підкріпленням є потужним методом навчання машин, який може допомогти автономним системам приймати рішення в реальному часі та навчитися вирішувати складні задачі. Однак, для успішного навчання з підкріпленням необхідно враховувати важливі аспекти, такі як винагорода, узагальнення, обчислювальна потужність, етика та відповідальність.

1.7 Метод вилучення тексту з веб-сторінок «Крок за кроком»

"Крок за кроком" - це метод вилучення тексту з веб-сторінок, який полягає в ітеративному вилученні не-текстових елементів до тих пір, поки не буде отримано бажаний текст. Цей метод дозволяє ефективно вилучати текст з веб-сторінок, які містять багато не-текстових елементів, таких як зображення, відео, реклама та інші.

Основна ідея методу полягає в тому, що зазвичай текст на веб-сторінках розташований в середині блоків або контейнерів, оточених не-текстовими елементами. Тому, щоб отримати бажаний текст, потрібно спочатку вилучити всі не-текстові елементи на сторінці. Після цього можна використовувати різні алгоритми вилучення тексту для вилучення бажаного тексту. Крок за кроком метод може бути особливо корисним для вилучення тексту з веб-сторінок, які містять складну структуру або динамічний контент, такі як соціальні мережі або новинні сайти. Такі сторінки можуть містити велику кількість не-текстових елементів, які ускладнюють процес вилучення тексту. Застосування методу "крок за кроком" може допомогти покращити ефективність вилучення тексту з таких сторінок та збільшити точність отриманих результатів.

У цілому, метод "крок за кроком" є потужним інструментом для вилучення тексту з веб-сторінок на основі навчання машин, який може бути використаний для різноманітних завдань, таких як збір інформації з Інтернету, аналіз даних та багато інших. Додатково можна сказати, що метод "крок за кроком" зазвичай використовується в поєднанні з іншими методами вилучення тексту з веб-сторінок, такими як метод "базової лінії" або "згорнутий байесовський класифікатор". Крім того, для покращення результатів використання методу "крок за кроком" можна застосовувати різні техніки обробки даних, такі як токенізація, стемінг та інші. Недоліком методу "крок за кроком" є те, що він може займати багато часу на виконання, особливо якщо сторінка містить велику кількість не-текстових елементів. Крім того, зміни в структурі сторінки або динамічний контент можуть ускладнити процес вилучення тексту. Однак, метод "крок за кроком" залишається одним з найпоширеніших методів вилучення тексту з веб-сторінок, оскільки він дозволяє ефективно вилучати текст з більшості веб-сторінок і є достатньо гнучким, щоб використовувати разом з іншими методами для досягнення кращих результатів.

2 ОБРОБКА ПРИРОДНОЇ МОВИ NLP

2.1 Актуальність теми

Актуальною темою навчання машин ML (Machine Learning) є обробка природної мови NLP (Natural Language Processing) [1].

В якості прикладу NLP можна розглянути Amazon Alexa. Amazon Alexa використовує NLP для створення розмови з користувачами. По-перше, пристрій Amazon, наприклад Echo, записує ваші слова. Запис вашого виступу надсилається на сервери Amazon для більш ефективного аналізу. Amazon розбиває вашу фразу на окремі звуки. Потім він підключається до бази даних, яка містить вимову різних слів, щоб знайти слова, які найбільше відповідають поєднанню окремих звуків. Потім Alexa визначає важливі слова, щоб зрозуміти завдання та виконати відповідні функції. Наприклад, якщо Alexa помічає такі слова, як надворі або температура, вона відкриває програму погоди. Нарешті сервери Amazon надсилають інформацію на ваш пристрій, і Alexa говорить.

NLP — це широкий термін для загального набору бізнес або обчислювальних проблем, які можна вирішити за допомогою навчання машин ML. Системи NLP передували ML. Перетворення мовлення в текст на старому мобільному телефоні та програми зчитування з екрана є прикладами рішень NLP. Зараз багато систем NLP використовують певну форму навчання машин. NLP працює з ієрархічною структурою мови. Слова знаходяться на найнижчому рівні ієрархії. Група слів утворює словосполучення. Наступним рівнем в ієрархії є група фраз, які утворюють речення, і, зрештою, речення передають ідеї.

Системи NLP стикаються з кількома серйозними проблемами.

Мова не точна. Часто ті самі слова чи фрази можуть мати кілька значень. Наприклад, розглянемо термін образ. Образ – зовнішній вигляд і образ – ікона. Слова можуть мати різні значення, які базуються на інших словах, які їх оточують.

Слова, які оточують слова, є контекстом. Слова також можуть мати різне значення залежно від флексії. Наприклад, фраза О, справді? може передати здивування, незгоду або багато інших значень, залежно від поєднання контексту та флексії.

Деякі з основних проблем для NLP включають.

Виявлення структури тексту. Одним із першочергових завдань будь-якої програми NLP є розбиття тексту на значущі одиниці, такі як слова, фрази та речення.

Позначення даних – після того, як система перетворить текст на дані, наступним завданням є застосування міток, які представляють різні частини мови. Для кожної мови потрібна інша схема маркування, яка відповідає граматиці мови.

Контекст представлення. Вирішення цієї проблеми – це те, де навчання машин може мати значний вплив. Оскільки значення слова залежить від контексту, будь-якій системі NLP потрібен спосіб представлення контексту. Це великий виклик через велику кількість контекстів. Перетворити контекст у форму, зрозумілу комп’ютерам, важко.

Застосування граматики. Незважаючи на те, що граматика визначає структуру мови, застосування граматики майже нескінченне. Робота з варіаціями в тому, як люди використовують мову, є головною проблемою для систем NLP

Можна застосовувати NLP до широкого кола проблем. Деякі з найпоширеніших програм включають: пошукові програми (наприклад, Google і Bing); перекладацькі послуги; аналіз настроїв для маркетингових або політичних кампаній; соціальні дослідження, що базуються на аналізі ЗМІ; інтерфейси людина машина (такі як Alexa); програми для керування контентом; чат-боти для імітації людської мови в додатках.

Для розв’язання основних проблем для NLP можна застосувати класичний конвеєр навчання машин. Першим етапом програми NLP є завантаження та обробка тексту. Цей етап можна розглядати як такий, що складається з трьох під етапів. По-перше, необхідно отримати дані з джерел даних. Наприклад, можна отримати текст із веб-сайтів або інших веб-ресурсів. Цей процес відомий як веб-

збирання. Якщо виконується завантаження даних з документів, необхідно перетворити їх у форму, яку вимагає застосований компонент завантаження. Для більшості реальних програм є бажання автоматизувати процес вилучення. Витягнувши текст, його можна завантажити в конвеєр перетворення. Цей процес можна виконати за допомогою бібліотек Python, але також можна автоматизувати процес за допомогою Amazon Texttract. Нарешті, можна перетворити текст у числове представлення для використання обраної моделі навчання машин ML.

2.2 Висновки до другого розділу

Обробка природної мови (NLP) є дисципліною, яка займається розробкою обчислювальних алгоритмів для автоматичного аналізу та обробки людської мови. Вона оцінює структуру мови та дозволяє системам навчання машин (ML) обробляти великі набори слів, фраз і речень. Навчання машин, в свою чергу, є підгалуззю ширшої області інформатики, відомої як штучний інтелект (AI). Штучний інтелект охоплює створення і розвиток систем, які можуть виконувати завдання, характерні для людей.

Одним з важливих застосувань NLP є аналіз пошукових запитів, взаємодія людини з машиною, а також маркетингові та соціальні дослідження. Наприклад, голосовий помічник Amazon Alexa використовує NLP для відповіді на запитання користувачів. Обробка природної мови також включає розуміння і аналіз письмового тексту.

Висновок полягає в тому, що обробка природної мови (NLP) є важливою галуззю навчання машин, яка знайшла широке застосування в різних сферах. Її розвиток сприяє взаємодії між людиною та комп'ютерами, дозволяючи машинам аналізувати та розуміти людську мову.

3 МЕТОДИ ВИЛУЧЕННЯ ТЕКСТУ З ІНТЕРНЕТУ

Існують потужні бібліотеки та фреймворки Python Beautiful Soup, Selenium або Scrapy. Вони не задовольняють усі потреби веб-збирання, а отже, важливо знати, який інструмент слід використовувати для конкретної роботи.

Нижче аналізуються відмінності між Beautiful Soup, Scrapy та Selenium.

3.1 Простота використання

Пакет Python Beautiful Soup [2] призначений для синтаксичного аналізу документів HTML і XML (зокрема, що мають неправильну розмітку, тобто незакриті теги, названий так на честь тегу soup). Він створює дерево синтаксичного аналізу для аналізованих сторінок, яке можна використовувати для вилучення даних із HTML, що корисно для веб-збирання.

Пакет Beautiful Soup пропонує всі рудиментарні інструменти, необхідні для сканування веб-сторінок, і це особливо корисно для інструментів, які зменшують долю недиференційованих важких робіт при вилученні тексту з інтернету. Він допомагає бізнесу зосередитися на їхній основній цінності для клієнтів, та не забирає у людей, які мають мінімальний досвід роботи з Python, багато часу та ресурсів.

Єдине застереження полягає в тому, що завдяки своїй простоті Beautiful Soup не такий потужний, як Scrapy або Selenium. Програмісти з досвідом розробки можуть освоїти як Scrapy, так і Selenium, але для початківців створення першого проекту може зайняти багато часу, якщо вони вирішають використовувати ці фреймворки замість Beautiful Soup.

Щоб отримати вміст тегу заголовка на example.com за допомогою Beautiful Soup, необхідно використовувати такий код (рис. 3.1.)

```
url = "https://example.com/"
res = requests.get(url).text
soup = BeautifulSoup(res, 'html.parser')
title = soup.find("title").text
print(title)
```

Рисунок 3.1 – Beautiful Soup

Щоб досягти подібних результатів за допомогою Selenium, необхідно написати відповідний код (рис. 3.2.)

```
url = "https://example.com"
driver = webdriver.Chrome("path/to/chromedriver")
driver.get(url)
title = driver.find_element(By.TAG_NAME, "title").get_attribute('text')
print(title)
```

Рисунок 3.2 – Selenium

Файлова структура проекту Scrapy складається з кількох файлів, що додає його складності. Наступний код бере назву з example.com (рис. 3.3.)

```
import scrapy

class TitleSpider(scrapy.Spider):
    name = 'title'
    start_urls = ['https://example.com']

    def parse(self, response):
        yield {
            'name': response.css('title'),
        }
```

Рисунок 3.3 – код, який бере назву з example.com

Якщо є необхідність отримувати дані зі служби, яка пропонує офіційний API, було б розумним рішенням використовувати API замість розробки програми веб-збирання.

3.2 Швидкість сканування

З цих трьох опцій Scrapy є явним переможцем, коли мова йде про швидкість. Це тому, що він підтримує паралельні обчислення за замовчуванням. Використовуючи Scrapy, можна надіслати кілька HTTP-запитів одночасно, і коли сценарій завантажить HTML-код для першого набору запитів, він буде готовий надіслати наступний пакет.

За допомогою Beautiful Soup можна використовувати бібліотеку потоків для надсилання одночасних HTTP-запитів, і для цього необхідно опанувати багато поточність. На Selenium неможливо досягти паралельної обробки без запуску кількох екземплярів браузера.

Якщо оцінювати ці три інструменти веб-збирання за швидкістю, Scrapy є найшвидшим, за ним йдуть Beautiful Soup і Selenium.

3.3 Використання пам’яті

Selenium — це API для автоматизації браузера, який знайшов своє застосування в області веб-збирання. Коли використовується Selenium для сканування веб-сайту, він породжує безголовий екземпляр браузера, який працює у фоновому режимі. Це робить Selenium ресурсномістким інструментом порівняно з Beautiful Soup і Scrapy.

Оскільки останні працюють повністю в командному рядку, вони використовують менше системних ресурсів і пропонують кращу продуктивність, ніж Selenium.

3.4 Вимоги до незалежності

Beautiful Soup — це набір інструментів аналізу, які допомагають видобувати дані з файлів HTML і XML. Він поставляється ні з чим іншим. Необхідно

використовувати такі бібліотеки, як запити або urllib, щоб робити HTTP-запити, вбудовані аналізатори для аналізу HTML/XML і додаткові бібліотеки для реалізації проксі-серверів або підтримки бази даних.

Scrapy, з іншого боку, поставляється з цілим shebang. Існують інструменти для надсилання запитів, аналізу завантаженого коду, виконання операцій із витягнутими даними та зберігання зібраної інформації. Можна додати інші функції до Scrapy за допомогою розширень і проміжного програмного забезпечення.

За допомогою Selenium завантажується веб-драйвер для браузера, який необхідно автоматизувати. Щоб реалізувати інші функції, наприклад зберігання даних і підтримку проксі-сервера, знадобляться сторонні модулі.

3.5 Якість документації

Загалом, документація кожного проекту добре структурована та описує кожен метод на прикладах. Але ефективність документації проекту значною мірою залежить і від читача.

Документація Beautiful Soup набагато краща для початківців, які починають працювати з веб-збиранням. Selenium і Scrapy, без сумніву, мають детальну документацію, але технічний жаргон може застати багатьох новачків зненацька.

Якщо спеціаліст має досвід роботи з концепціями та термінологією програмування, будь-яку з трьох документацій буде легко прочитати.

3.6 Підтримка розширень

Scrapy — це найбільш розширюваний фреймворк Python для сканування веб-сайтів. Він підтримує проміжне програмне забезпечення, розширення, проксі-сервери тощо та допомагає розробити сканер для великомасштабних проектів.

Можна створити надійні та ефективні сканери, реалізувавши проміжне програмне забезпечення в Scrapy, яке в основному є хуками, які додають користувальницькі функції до механізму фреймворку за замовчуванням. Наприклад, HttpErrorMiddleware піклується про помилки HTTP, тому павукам не потрібно мати справу з ними під час обробки запитів.

Проміжне програмне забезпечення та розширення є ексклюзивними для Scrapy, але можна досягти подібних результатів із Beautiful Soup і Selenium, використовуючи додаткові бібліотеки Python.

3.7 Відтворення JavaScript

Selenium має один варіант використання, де він перевершує інші бібліотеки веб-збиранням, а саме – копіювання веб-сайтів із підтримкою JavaScript. Хоча можна очищати елементи JavaScript за допомогою проміжного програмного забезпечення Scrapy, робочий процес Selenium є найпростішим і найзручнішим з усіх.

Використовується веб-переглядач, щоб завантажити веб-сайт, взаємодіяти з ним за допомогою клацань і натискань кнопок, а коли є вміст, який потрібно скопіювати на екрані, його можна витягнути за допомогою селекторів CSS і XPath Selenium.

Beautiful Soup може вибирати елементи HTML за допомогою селекторів XPath або CSS. Однак він не пропонує функціональність для сканування елементів, відтворених JavaScript, на веб-сторінці

4 МОДЕЛЬ ВИКОРИСТАННЯ BEAUTIFUL SOUP

Модель використовує Beautiful Soup [1], щоб видобувати заголовки, авторів, резюме, опубліковані дані та гіперпосилання з публікацій блогу. Щоб потім витягнутий текст можна було б використати в подальших завданнях NLP, таких як виділення теми, аналіз настроїв, перетворення тексту в мовлення або переклад.

Повідомлення в блозі, яке аналізувалося, є блогом навчання машин AWS [2].

За допомогою веб-браузера була відкрита сторінка AWS Machine Learning. Використовувався режим інспектора браузера, щоб дізнатися структуру сторінки. У Mozilla FireFox і Google Chrome можна відкрити інспектор, натиснувши CTRL+SHIFT+C. Якщо використовується інший браузер, необхідно звертатися до документації браузера.

Були переглянуті різні елементи веб-сторінки, переміщаючи вказівник на сторінку. Переміщенням вказівника на наступні елементи було визначено, чи можна знайти теги, які використовуються для ідентифікації інформації: заголовок публікації в блозі; автор; дата публікації; короткий текст; гіперпосилання на публікацію в блозі.

Покрокова методика пошуку тегів наведена нижче.

Код статусу HTTP дорівнює 200, це передумова виконання наступних кроків.

Вміст зі сторінки content був завантажений в об’єкт soup.

Всю сторінка доступна для перегляду за допомогою функції soup.prettify().

Примітка. Вміст зі сторінки блогів AWS може бути довгим. Щоб перейти до наступного завдання, необхідно прокручувати блокнот JupyterLab вниз.

До всіх елементів сторінки можна отримати доступ за допомогою крапкової нотації (.). Таким чином, щоб переглянути заголовок, можна використовувати soup.title. Якщо потрібен лише текст, можна використовувати текстовий елемент soup.h2.text.

Best Egg досяг утричі швидшого навчання моделі ML за допомогою автоматичного налаштування моделі Amazon SageMaker.

Коли використовувався інспектор для пошуку тегів на сторінці блогів AWS, було виявлено, що вміст публікації в блозі впорядковано organized/categorized/marked позначено тегами <article>, які вказують на окрему одиницю вмісту.

Заголовок можна знайти на soup.article.h2.span.

Щоб відобразити лише текст, використалась властивість text.

Дата публікації статті знайдена за допомогою: soup.article.time.text.

Далі короткий зміст статті витягнутий за допомогою: soup.article.section.p.text.

Прізвище автора вказано у нижньому колонтитулі. Допис у блозі може мати кількох авторів. Однак спочатку було отримано лише першого автора: soup.article.footer.span.prettify().

Гіперпосилання на повний текст статті є останньою інформацією, яка була знайдена: soup.article.a[‘href’].

Тепер коли були визначили всі відповідні елементи. Можна знайти всі статті за допомогою функції find_all().

Визначивши формат даних, можна додати результати до масиву:

Далі був завантажений масив у фрейм даних pandas.

Стовпець published тобто значення дати й часу були перетворені за допомогою метода to_datetime().

Ширину стовпця було налаштовано для pandas і відображені перші п’ять рядків фрейму даних.

Тепер, коли дані знаходяться у фреймі даних pandas, їх можна використовувати у наступних завданнях NLP.

5 РЕЗУЛЬТАТИ РОБОТИ МОДЕЛІ

Вся сторінка [3] доступна для перегляду за допомогою функції soup.prettify() (рис. 5.1)

```
[73]: print(soup.prettify())

<!DOCTYPE html>
<html class="no-js aws-lng-en_US" data-aws-assets="https://a0.awsstatic.com" data-css-v
ersion="1.0.537" data-js-version="1.0.661" data-static-assets="https://a0.awsstatic.com
" lang="en-US" xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <meta charset="utf-8"/>
  <meta content="width=device-width, initial-scale=1" name="viewport"/>
  <title>
   AWS Machine Learning Blog
  </title>
```

Рисунок 5.1 – перегляд HTML-коду всієї сторінки через функцію soup.prettify()

Примітка. Вміст зі сторінки блогів AWS може бути довгим. Щоб перейти до наступного завдання, необхідно прокручувати блокнот JupyterLab вниз.

До всіх елементів сторінки можна отримати доступ за допомогою крапкової нотації (.). Таким чином, щоб переглянути заголовок, можна використовувати soup.title. Якщо потрібен лише текст, можна використовувати текстовий елемент soup.h2.text(рис. 5.2).

```
[76]: print(soup.h2.text)

Best Egg achieved three times faster ML model training with Amazon SageMaker Automatic
Model Tuning

When you used the inspector to search for tags on the AWS Blogs page, you might have found that
blog-post content is organized/categorized/marked with <article> tags, which indicate a self-
contained unit of content.
```

Рисунок 5.2 – текстовий елемент soup.h2.text

Коли використовувався інспектор для пошуку тегів на сторінці блогів AWS, було виявлено, що вміст публікації в блозі впорядковано organized/categorized/marked позначено тегами <article>, які вказують на окрему одиницю вмісту.

Заголовок можна знайти на soup.article.h2.span(рис.5.3).

```
[92]: print(soup.article.h2.prettify())
```

```
<h2 class="lb-bold blog-post-title">
 <a href="https://aws.amazon.com/blogs/machine-learning/best-egg-achieved-three-times-faster-ml-model-training-with-amazon-sagemaker-automatic-model-tuning/" property="url" rel="bookmark">
  <span property="name headline">
   Best Egg achieved three times faster ML model training with Amazon SageMaker Automatic Model Tuning
  </span>
 </a>
</h2>
```

Рисунок 5.3 – заголовок з soup.article.h2.span

Щоб відобразити лише текст, використалась властивість text(рис. 5.4)

```
[93]: print(soup.article.h2.span.text)
```

```
Best Egg achieved three times faster ML model training with Amazon SageMaker Automatic Model Tuning
```

Рисунок 5.4 – використання властивості text

Дата публікації статті знайдена за допомогою: soup.article.time.text(рис. 5.5).

```
[94]: print(soup.article.time.text)
```

```
26 JAN 2023
```

Рисунок 5.5 – пошук дати публікації через soup.article.time.text

Далі короткий зміст статті витягнутий за допомогою: soup.article.section.p.text(рис. 5.6).

```
[95]: print(soup.article.section.p.text)
```

```
This post is co-authored by Tristan Miller from Best Egg. Best Egg is a leading financial confidence platform that provides lending products and resources focused on helping people feel more confident as they manage their everyday finances. Since March 2014, Best Egg has delivered $22 billion in consumer personal loans with strong credit performance, welcomed […]
```

Рисунок 5.6 – використання soup.article.section.p.text

Прізвище автора вказано у нижньому колонтитулі. Допис у блозі може мати кількох авторів. Однак спочатку було отримано лише першого автора: soup.article.footer.span.prettify() (рис. 5.7).

```
[96]: print(soup.article.footer.span.prettify())

<span property="author" typeof="Person">
 <span property="name">
  Tristan Miller
 </span>
</span>
```

Рисунок 5.7 – застосування soup.article.footer.span.prettify()

Гіперпосилання на повний текст статті є останньою інформацією, яка була знайдена: soup.article.a[‘href’](рис.5.8).

```
[97]: print(soup.article.a['href'])

https://aws.amazon.com/blogs/machine-learning/best-egg-achieved-three-times-faster-ml-model-training-with-amazon-sagemaker-automatic-model-tuning/
```

Рисунок 5.8 – остання інформація, яка була знайдена через soup.article.a[‘href’]

Тепер коли були визначили всі відповідні елементи. Можна знайти всі статті за допомогою функції find_all()(рис.5.9).

```
[111]: for article in soup.find_all('article'):
           print('========================================')
           print(article.h2.span.text)
           authors = article.footer.find_all('span', {"property":"author"})
           print('by', end=' ')
           for author in authors:
               if author.span != None:
                   print(author.span.text, end=', ')
           print(f'on {article.time.text}')
           print(article.section.p.text)
           print(article.a['href'])
```

```
========================================
Best Egg achieved three times faster ML model training with Amazon SageMaker Automatic
Model Tuning
by Tristan Miller, Ajjay G, Ganapathi Krishnamoorthi, Hariharan Suresh, Valerio Perrone
, on 26 JAN 2023
This post is co-authored by Tristan Miller from Best Egg. Best Egg is a leading financi
al confidence platform that provides lending products and resources focused on helping
people feel more confident as they manage their everyday finances. Since March 2014, Be
st Egg has delivered $22 billion in consumer personal loans with strong credit performa
nce, welcomed […]
https://aws.amazon.com/blogs/machine-learning/best-egg-achieved-three-times-faster-ml-m
odel-training-with-amazon-sagemaker-automatic-model-tuning/
========================================
Build a loyalty points anomaly detector using Amazon Lookout for Metrics
by Dhiraj Thakur, on 25 JAN 2023
Today, gaining customer loyalty cannot be a one-off thing. A brand needs a focused and
```

Рисунок 5.9 – пошук за допомогою функції find_all()

Визначивши формат даних, можна додати результати до масиву:

Далі був завантажений масив у фрейм даних pandas.

Стовпець published тобто значення дати й часу були перетворені за допомогою метода to_datetime().

Ширину стовпця було налаштовано для pandas і відображені перші п’ять рядків фрейму даних(рис.5.10).

```
[117]: pd.options.display.max_rows
       pd.set_option('display.max_colwidth', None)
       df.head()
```

[117]:

	title	authors	published	summary	link
0	Best Egg achieved three times faster ML model training with Amazon SageMaker Automatic Model Tuning	Tristan Miller, Ajjay G, Ganapathi Krishnamoorthi, Hariharan Suresh, Valerio Perrone	2023-01-26	This post is co-authored by Tristan Miller from Best Egg. Best Egg is a leading financial confidence platform that provides lending products and resources focused on helping people feel more confident as they manage their everyday finances. Since March 2014, Best Egg has delivered $22 billion in consumer personal loans with strong credit performance, welcomed [...]	https://aws.amazon.com/blogs/machine-learning/best-egg-achieved-three-times-faster-ml-model-training-with-amazon-sagemaker-automatic-model-tuning/
1	Build a loyalty points anomaly detector using Amazon Lookout for Metrics	Dhiraj Thakur	2023-01-25	Today, gaining customer loyalty cannot be a one-off thing. A brand needs a focused and integrated plan to retain its best customers—put simply, it needs a customer loyalty program. Earn and burn programs are one of the main paradigms. A typical earn and burn program rewards customers after a certain number of visits or spend. [...]	https://aws.amazon.com/blogs/machine-learning/build-a-loyalty-points-anomaly-detector-using-amazon-lookout-for-metrics/
	Explain text classification model predictions	Pinak Panigrahi	2023-01-	Model explainability refers to the process of relating the prediction of a machine learning (ML) model to the input feature values of an instance in humanly understandable terms. This field is often	https://aws.amazon.com/blogs/machine-learning/explain-text-classification-

Рисунок 5.10 – перші п’ять рядків фрейму даних у фреймі даних pandas

Тепер, коли дані знаходяться у фреймі даних pandas, їх можна використовувати у наступних завданнях NLP.

ВИСНОВКИ

Інтернет повний необроблених даних. Вилучення тексту з інтернету допомагає перетворити ці дані на значущу інформацію, яку можна використати з користю. Selenium, мабуть, є найбезпечнішим вибором, якщо ви хочете скинути веб-сайт за допомогою javascript або потрібно активувати деякі елементи на екрані перед вилученням даних.

Scrapy — це повноцінний фреймворк для сканування веб-сторінок для всіх потреб, незалежно від того, чи є завдання написати маленький сканер чи великомасштабний сканер, який постійно сканує Інтернет для оновлених даних.

Пакет Beautiful Soup пропонує всі рудиментарні інструменти, необхідні для сканування веб-сторінок, і це особливо корисно для інструментів, які зменшують долю недиференційованих важких робіт при вилученні тексту з інтернету. Він допомагає бізнесу зосередитися на їхній основній цінності для клієнтів, та не забирає у людей, які мають мінімальний досвід роботи з Python, багато часу та ресурсів.

Який би фреймворк чи бібліотека не використовувалась, легко почати вивчати сканування інтернету за допомогою мови програмування Python.

СПИСОК ЛІТЕРАТУРИ

[1] What exactly is Alexa? Where does she come from? And how does she work? [Online]. Available: https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/

[2] Sadavarte, Sanket Sanjay, and Eliane Bodanese, “Pregnancy Companion Chatbot Using Alexa and Amazon Web Services”, IEEE Pune Section International Conference (PuneCon), pp. 1- 5. IEEE, 2019.

[3] Gobinda G. Chowdhury. “Natural Language Processing”. In: Annual Review of Information Science and Technology, 2003. – C. 51-89.

[4] J. Le. (2018). “The 7 NLP Techniques That Will Change How You Communicate in the Future (Part I)” [Online]. Available: https://heartbeat.fritz.ai/the-7-nlp-techniques-that-will-change-how-youcommunicate-in-the-future-part-i-f0114b2f0497/

[5] Ted Briscoe. Introduction to Linguistics for Natural Language Processing. Tech. rep. 2013, pp. 1–37.

[6] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu. Deep Learning Based Text Classification: A Comprehensive Review // 2005.

[7] Liddy, E.D. 2001. Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. NY. Marcel Decker, Inc.

[8] Jurafsky D, Martin J H (2002) Speech and Language Processing – An Intro to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson Education Asia, ISBN 81-7808-594-1

[9] R. Socher, “Recursive deep learning for natural language processing and computer vision” Stanford University, 2014, pp. 8-120.

[10] Daniel Jurafsky. Classification: Naive Bayes, Logistic Regression, Sentiment / Daniel Jurafsky, James H. Martin // Speech and Language Processing / Daniel

Jurafsky, James H. Martin., 2015. – С. 1–28.

[11] Бенгфорт Бенджамин, Билбро Ребекка, Охеда Тони Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка. – СПб.: Питер, 2019. – 368 с.

[12] Хобсон Лейн, Ханнес Хапке, Коул Ховард Обработка естественного языка в действии. – СПб.: Питер, 2020. – 576 с

[13] Manaswi N.K. (2018) Developing Chatbots. In: Deep Learning with Applications Using Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-3516-4_11

[14] What is Amazon Transcribe? [Електронний ресурс] – Режим доступу до ресурсу: https://docs.aws.amazon.com/transcribe/latest/dg/transcribe-whatis.html

[15] Большакова Е.И., Воронцов К.В., Ефремова Н.Э., Клышинский Э.С., Лукашевич Н.В., Сапин А.С. Автоматическая обработка текстов на естественном языке и анализ данных: навч. посіб. – М.: НИУ ВШЭ, 2017. – 269 с.

[16] What is Amazon Polly? [Електронний ресурс] – Режим доступу до ресурсу: https://docs.aws.amazon.com/polly/latest/dg/what-is.html

[17] What is Amazon Translate? [Електронний ресурс] – Режим доступу до ресурсу: https://docs.aws.amazon.com/translate/latest/dg/what-is.html

[18] Klopfenstein, L., Delpriori, S., Malatini, S., Bogliolo, A.: The rise of bots: a survey of conversational interfaces, patterns, and paradigms. In: Proceedings of the 2017 Conference on Designing Interactive Systems, pp. 555–565. Association for Computing Machinery (2017).

[19] Poibeau T., Saggion H., Piskorski J., Yangarber R. Multi-source, Multilingual Information Extraction and Summarization. Theory and Applications of Natural Language Processing. Springer, Berlin, Heidelberg, 2013. – 257 с.

[20] What is Amazon Comprehend? [Електронний ресурс] – Режим доступу до ресурсу: https://docs.aws.amazon.com/comprehend/latest/dg/what-is.html

[21] M. Bates (1995). “Models of natural language understanding” [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC40721/

[22] Abdul-Kader, S., & Woods, J. (2015). Survey on Chatbot Design Techniques in Speech Conversation Systems. International Journal of Advanced Computer Science and Applications, 6(7). http://doi.org/10.14569/ijacsa.2015.060712

[23] J. Chai and J.Lin, “The role of natural language conversational interface in online sales: a case study,” International Journal of Speech Technology., vol. 4, pp. 285–295, Nov. 2001.