Міністерство освіти і науки України

Департамент освіти і науки Дніпропетровської облдержадміністрації

Дніпропетровське територіальне відділення МАН України

Дніпровський ліцей №3 Дніпровської міської ради

Наукове товариство «Інтелектуал»

Відділення: Інформаційні технології

Секція: Системи та технології

штучного інтелекту

АНАЛІЗ ПРИРОДНЬОЇ МОВИ, РОЗРОБЛЕННЯ ЧАТ-БОТІВ.

Роботу виконав:

Козенко Артур Олександрович,

учень 11-А класу

Дніпровський ліцей №3 ДМР

Науковий керівник:

Вовк Тетяня Вітальевна,

учитель інформатики,

вчитель вищої

кваліфікаційної

категорії,

вчитель-методист

Дніпро - 2024

Зміст:

[ВСТУП 3](#_Toc182582484)

[РОЗДІЛ І 5](#_Toc182582485)

[МОЖЛИВОСТІ ТА СФЕРИ ЗАСТОСУВАННЯ АНАЛІЗУ ПРИРОДНЬОЇ МОВИ.МЕТОДИ НАВЧАННЯ МОДЕЛЕЙ 5](#_Toc182582486)

[1.1. Сфери застосування аналізу природньої мови у сьогоденні(NLP). 5](#_Toc182582487)

[1.2. Складність аналізу природньої мови. 6](#_Toc182582488)

[1.3. Методи створення та навчання моделей, які можуть аналізувати природню мову. Графематичний аналіз. 7](#_Toc182582489)

[1.4. Звідки беруться дані для начання ШІ. Скрита функція капчі 10](#_Toc182582490)

[РОЗДІЛ ІІ 11](#_Toc182582491)

[ПРОЕКТУВАННЯ ФУНКЦІОНАЛУ ДЛЯ ЧАТ-БОТУ 11](#_Toc182582492)

[2.1. Підготовка технічного функціоналу чат-бота 11](#_Toc182582493)

[РОЗДІЛ ІІІ 14](#_Toc182582494)

[СТВОРЕННЯ ЧАТ-БОТУ ТА ДОДАВАННЯ ДО НЬОГО ЗАЗДЕЛЕГІТЬ ПІДГОТОВЛЕНОГО ФУНКЦІОНАЛУ 14](#_Toc182582495)

[3.1. Плюси та недоліки чат-ботів 14](#_Toc182582496)

[3.2. Інструкція створення власного чат-бота у телеграмі 14](#_Toc182582497)

[3.3. Додаємо до чат-боту функціонал з можливостю аналізу природньої мови використовуючи бібліотеку aiogram 15](#_Toc182582498)

[3.3.0. Вибір фреймворка для чат-боту 15](#_Toc182582499)

[3.3.1. Створення базового ехо-бота 16](#_Toc182582500)

[3.3.2. Створення команди /help і заміна клавіатури 16](#_Toc182582501)

[3.3.3. Обробка всіх команд. Застосування заготовлених функцій 17](#_Toc182582502)

[3.3.4 Удосконалення проекту 18](#_Toc182582503)

[ВИСНОВОК 19](#_Toc182582504)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 20](#_Toc182582505)

[Додатки 21](#_Toc182582506)

# ВСТУП

На сьогоднішній день люди не вже не уявляють свого життя без гаджетів. Вони допомагають нам розслабитися, відпочити після роботи та навіть навчатися. Виробники додатків, як для смартфонів, так і для інших пристроїв вже давно змагаються у гонці за нашу увагу. Додають новий функціонал та можливості і намагаються бути кращими за своїх конкурентів. І тут не обійшлося без аналізу природньої мови (NLP - Natural Language Processing). Технології продовжують розвиватися і все більше компаній вже використовує цю технологію. Важливість аналізу природної мови визначається його потенціалом у забезпеченні машинам здатності розуміти, інтерпретувати та взаємодіяти з людьми через природну мову. Такий підхід розкриває широкі можливості в розвитку інноваційних систем та додатків, що полегшують повсякденне життя та вирішують складні завдання.

Навчені аналізу та обробки природньої мови моделі вже давно створюють нам стрічку «рекомендації» у Instagram, TikTok і в подібних соціальних мережах. Штучний інтелект підбирає для нас музику, друзів по інтересам та навіть рекламу. Але в аналізу природньої мови є і інші сфери застосування. Нижче список можливостей NLP:

* Переклад тексту;
* Аналіз вмісту повідомлення:
  + Настрій повідомлення (позитивне, негативне, нейтральне тощо);
  + Аналіз на токсичність повідомлення;
  + Переказ тексту;
  + Пошук ключової інформації;
* Класифікація тексту.
* Генерація тексту:
  + Генерації зображешь по опису;
  + Опис наданих зображень;
  + Вигадування історій;
  + Перефразування тексту, заміна слів на синоніми;
* Пошук помилок у тексті та виправлення їх.
* Автоматизування (повне або часткове) чат-ботів:
  + Голосовий помічник;
  + Чат-бот що відповідає на питання клієнтів або за проханням з’єднує з оператором;

Наприклад голосовий помічник «Siri» від Apple – яскравий приклад використання технологій аналізу природньої мови. В цій науковій роботі я покажу приклад використання даних технологій за допомогою телеграм боту.

Телеграм – це месенджер, який завоював довіру мільйонів користувачів своєю безпековою, швидкою та функціональною платформою. Зазвичай чат-боти у Telegram використовуються як засіб спілкування, розваги, а також в бізнесі для автоматизації клієнтського обслуговування та взаємодії з користувачами. Сприятлива атмосфера платформи Телеграм дозволяє створювати не тільки зручні, але й потужні інструменти для взаємодії з користувачами. Інтерфейс месенджера, з його простотою та можливістю впровадження різноманітних медіа-елементів, створює ідеальне середовище для реалізації різних завдань та функцій чат-ботів. Але перед практикою, давайте трохи пройдемося по теоретичній базі, а саме розглянемо конкретні сфери застосування аналізу природньої мови, розглянемо складнощі аналізу, методи створення та навчання моделей та ознайомимося із сервісом Hugging Face.

Метою даної наукової роботи є дослідження сфер використання аналізу природньої мови, методинавчання ШІ для NLP, а також створення власного чат-боту у телеграмі з можливостями аналізу природньої мови.

Об’єкт дослідження: використання аналізу природньої мови (NLP) у створенні чат-ботів.

Предмет дослідження: використовуючи можливості телеграму, створити власного чат-бота з використанням аналізу природної мови як основного функціоналу.

Завдання:

* + Розглянути сфери застосування аналізу природньої мови.
  + Ознайомитися з методами сворення та начання ШІ.
  + Спроєктувати та створити функціонал для чат-боту.
  + Створити сам чат-бот і додати до нього раніше розроблений і підготовлений функціонал.

Методи дослідження:

* Метод аналізу та синтезу;
* Метод «теорія-практика»;
* Розробка та тестування.

# РОЗДІЛ І

# МОЖЛИВОСТІ ТА СФЕРИ ЗАСТОСУВАННЯ АНАЛІЗУ ПРИРОДНЬОЇ МОВИ.МЕТОДИ НАВЧАННЯ МОДЕЛЕЙ

1.1. Сфери застосування аналізу природньої мови у сьогоденні(NLP).

Аналіз природньої мови (далі просто NLP - Natural Language Processing) використовується у багатьох сферах для аналізу, розуміння, перекладу та генерації людської мови. Розглянемо детальніше, як NLP використовується в різних сферах.

Клієнтське обслуговування – NLP використовуюється для автоматизованої підтримки клієнтів через онлайн-чати та голосові асистенти. Частіше за все таким користуються банки або підприємства, що пропонують різні послуги чи продають речі, наприклад магазини одягу чи побутової техніки.

Аналіз текстів - NLP використовується для аналізу текстових документів, включаючи фінансові звіти, заявки на кредити та шахрайство. Такими методами також користуються в більшості банки та подібні організації пов’язані зі сферою інвестицій.

Одна з найважливіших сфер застосування NLP – це діагностика та лікування. ШІ може аналізувати медичні записи, симптоми пацієнтів та наукові публікації для діагностики захворювань та розробки лікувальних планів. Це якраз та ситуація, де NLP рятує та впливає на людські життя. Також не мало важлива роль ­аналізу природньої мови - медична документація. Автоматизовані системи можуть перетворювати мовлення медичних фахівців у структуровані дані для полегшення обміну інформацією та зниження ризику помилок.

NLP також використувується у науці для аналізу наукових статей, виявлення зв'язків та трендів у дослідженнях. Також NLP допомагає в екстрагуванні інформації з великої кількості текстів та створенні корисних даних для досліджень.

Відповідно до сказаного вище, можна прийти до висновку, що NLP допомагає в навчені і це буде вірна думка. NLP може адаптувати зміст навчальних матеріалів до індивідуальних потреб учнів, а також ШІ може самотужки оцінювати рівень знань учнів та автоматизовано генерувати звіти щодо успішності та навчання.

А також, раніше згадана екстракція знань застосовується в сфері торгівлі та фінансів. Системи NLP допомагають в екстрагуванні інформації з великої кількості текстів та створенні корисних даних для досліджень. Ще один корисний спосіб застосування аналізу природньої мови в сфері торгівлі – це аналіз ринків. Він використовується для аналізу фінансових новин, соціальних медіа та інших джерел для прогнозування трендів на фінансових ринках. І так само в трейдингу: Деякі компанії використовують NLP для розуміння та аналізу текстової інформації для прийняття рішень щодо трейдингу на фондовому ринку.

Остання сфера, яка буде згадана сьогодні – сфера розваг. Багато популярних месенджерів та програм використовують NLP задля покращення свого продукту, зробити якого більш цікавим, зручним та корисним. Голосові асистенти та чат-боти, у яких використовується NLP для розуміння та відповіді на запитання користувачів у віртуальних асистентах та ігрових середовищах.

Персоналізація контенту використовується для рекомендацій змісту, такого як фільми, музика та книги, на основі історії перегляду та інших персональних даних. Найкращі та всім відомі приклади – це додатки TikTok, Instagram, Facebook, Twitter, Youtube. Окрім того, що персоналізація контенту допомагає підпирати цікаві відео та пости для вас, так ще за допомогою NLP можуть підбиратися «можливо знайомі», фільтри та аккаунти.

Див. у додатку А , Рисунок А.1. - Статистика використання аналізу природньої мови у різних сферах.

Отже, аналіз природньої мови застосовується у дуже багатьох сферах,пов'язаниз із бізнесом, розвагами, начанняим, лікуванням тощо. NLP покращує продуктивності, допомагає у зростанні ефективності та покращенні користувацького досвіду. Вона дозволяє комп'ютерам розуміти та взаємодіяти з людьми більш природним способом, що має значний потенціал для впливу на багато аспектів нашого повсякденного життя. Сьогоднішній світ вже не уявити без використання ШІ.

1.2. Складність аналізу природньої мови.

Не дивлячись на сучасні можливості та потужність комп’ютерів, аналіз природньої мови залишається непростою задачею, яка потребую великого вкладу як фінансів, так і часу. Річ у тому, якщо казати просто, то штучний інтелект всю інформацію сприймає як числа. Фото, текст, голос – це все для нього всього лише набір нулів та одиниць. Тому наша мова для нього не є природньою. Розглянемо основні складності в розумінні та аналізу нашої мови.

1. Неоднозначність слів. Всі чули ще зі школи про ононіми -це слова, які пишуться однаково, але, залежно від контексту, мають різне значення, як от ключ (ключ яким відкриваються двері, ключ журавлів, джерело води і інструмент яким закручуються гайки) або коса (інструмент у сільськогосподарстві та вид зачіски).
2. Спрощення. Люди звикли для спрощення, або щоб уникнути тавтології, використовувати тире. Наприклад: «*я вважаю, що найкраща страва – це борщ, а вона – піца*». Тут у другій частині речення упущено і замінено за допомогою тире фразу «вважає, що найкраща страва». Так зроблено, тому що нам, людям, і так зрозуміло, що мова йшла про те, *що моя улюблена страва – це борщ, а її - це піца*. Але штучний інтелект без потрібної підготовки, може подумати, що малось на увазі «я полюбляю борщ, а вона піца (тобто вона є піцею, а не людиною)»
3. Розрізнення знаків пунктуації. Особливо при обробці голосу буває важливо розуміти інтонацію та розрізняти знаки пунктуації. Я думаю, усі чули про такий приклад: «*Помилувати неможна стратити!*». Тут упущена кома, і, залежно від того, де вона стоїть, буде різний посил. «*Помилувати, неможна стратити!*» (залишити живим) або ж «*Помилувати неможна, стратити!*» (стратити). Тобто навіть ціле речення може нести в собі різний посил. Тому треба бути дуже обережними у подібних ситуаціях і намагатися краще підготувати ваш ШІ, якщо ви будете використовувати для розпізнавання та генерації голосу.
4. Особливості української мови (та деяких інших). Навіть для людини треба багато часу щоб вивчити нову мову. Окрім значень слів, майже у всіх мовах є різні закінчення, суфікси, префікси, чергування звуків тощо. Наприклад в українській мові є відмінки, рід, особа, число (множина, однина), відміна у іменників. Вид, час, особа у дієслова тощо.

Тобто аналіз природньої мови є складною задачею через низку причин, таких як неоднозначність слів, спрощення, розрізнення знаків пунктуації та особливості української мови. Також можуть виникати складнощі з фонетикою, коли людина не дуже чітко каже слова (наприклад іноземець вчить іншу мову і ще не має правильного акценту). Це все ускладнює процес розпізнавання та генерації тексту і голосу за допомогою штучного інтелекту.

1.3. Методи створення та навчання моделей, які можуть аналізувати природню мову. Графематичний аналіз.

Розглянемо декілька найпоширеніших методів створення та навчання ШІ для аналізу природньої мови, а потім розберемо алгоритм токенізації та принцип графематичного аналізу.

* Deep Learning або методи глибокого навчання – це глибокі нейронні мережі, що є одним з основних інструментів у області NLP. Методи такого навчання, такі як рекурентні нейронні мережі (RNN), згорткові нейронні мережі (CNN) та трансформери, широко використовуються для різних задач обробки природної мови (NLP), включаючи машинний переклад, розпізнавання мови, аналіз настроїв та інші.
* Одні із найпоширеніших методів навчання ШІ для NLP та не тільки - методи звичайного навчання з учителем і без нього. Вони використовуються для навчання моделей на основі набору даних, який містить приклади вхідних даних та відповідні вихідні мітки. Данні розбиваються на дві вибірки (як правило на 75% і 25% відповідно): одна для навчання, а інша – для тесту, тому що важливо, щоб при тестуванні моделі вона отримувала незнайомі для неї дані. А методи без учителя можуть використовуватися для виявлення структури в даних, навіть якщо немає попередньо заданих відповідей. Тобто шукати закономірності і схиляючись на попередній досвід видавати на вихід певну інформацію
* Методи передового навчання з підкріпленням абл Reinforcement Learning. Вони використовуються для навчання моделей приймати послідовні дії в середовищі з метою максимізації деякого скалярного сигналу винагороди. Ці методи допомагають у вирішенні задач, де потрібно приймати рішення на основі тексту або взаємодіяти з ним. Саме те, без чого не обійтись у аналізі природньої мови.
* Методи передбачення зв'язків та аналізу тексту. Ці методи використовуються для виявлення та аналізу зв'язків між різними елементами тексту, такими як іменовані сутності, ключові слова, семантичні відношення тощо.

З методами трохи розібрались. Тепер розглянемо принцип Графематичного аналізу. Графематичний аналіз це такий алгоритм, який допомагає розбити текст на окремі графеми (найменші одиниці письма), такі як літери та графічні символи, з метою подальшого їх аналізу.

1. Токенізація. На початку текст розбивається на окремі графеми (слова, знаки пунктуації, емодзі тощо). Нижче поданий алгоритм розбиття на графеми:
   1. Розбиття на абзаци. Тут все просто, просто шукаємо отступи і розбиваємо текст.
   2. Розбиття на речення. На цьому етапі вже задіюється спеціально навченій ШІ, що буде розбивати на речення. Просто розбиття за допомогою розділення тексту в місцях, де є крапка – буде не завжди правильне рішення. Іноді крапка – це і є частина слова, назви фірми, а також скорочення. Наприклад фраза «інститут ім. І.Франка.» в такому випадку розіб’ється на 3 речення. Тому і слід використовувати вже навчені моделі для покращення результату.
   3. Розбиття на окремі слова та знаки пунктуації. Для того, щоб у нас виходили «чисті» слова, без лапок, крапок і так далі, потрібно не тільки відокремлювати речення на місцях пробілів, але й відокремлювати кожний знак пунктуації.
   4. Розділення на частини слів. У сучасних методах навчання ШІ не рідко використовують розбиття на частини слів. Наприклад слово «геологорозвідувальний» буде розбито на «геолого|розвідувальний». Найпопулярніші алгоритми які використовують таке розділення, це BPE, WordPiece, Unigram, SentencePiece.
   5. \*\*для деяких алгоритмів, також використовують розбиття на окремі символи.
2. Нормалізація. Отримані в результаті токенізації графеми можуть бути нормалізовані до стандартної форми, наприклад, переведені до нижнього регістру або видалені деякі непотрібні символи.
3. Морфологічний аналіз. Як ми розбирали в пункті 1.2. – мови мають різні закінчення, суфікси і префікси, але при цьому можуть нести один і той самий посил. Розглянемо способи морфологічного аналізу природньої мови:
   1. Лематизація – процес приведення словоформи до леми (її нормальній формі)
   2. Стематизація – це інший спосіб спрощення слів, але тут вже кожне слово в тексті замінюється на основу слова.
   3. Визначення частини мови – іменники, прикметники, дієслова, дієприкметники тощо.
   4. Постановка в задану форму. Це треба вже в тих випадках, коли наш ШІ повинен генерувати текст для людей, який буде для нас природнім і легко читатися.
   5. Знаяття неоднозначності. Цим також займається морфологічний аналіз. Тему неоднозначності ми згадували в 1.2 1).
4. Останній етап який розгялнемо – це синтаксичний аналіз. Це процес розбору тексту для визначення структури синтаксичних відношень між словами та фразами. Основна мета синтаксичного аналізу полягає в тому, щоб зрозуміти синтаксичну структуру речення, визначити ролі слів у відношенні до один одного та виявити зв'язки між ними. Для цих цілеей існують такі методи, як: метод засобів контекстно-вільної граматики (CFG), методи засновані на статистиці, методи на основі машинного навчання.

1.4. Звідки беруться дані для начання ШІ. Скрита функція капчі

Прочитавши параграфи вище, не складно догадатися, що навіть для навчання ШІ нерідко використовуються інші ШІ. А також, для того щоб навчити його аналізу природньої мови, треба мати велику вибірку даних.

Найчастіший джерело даних це соціальні мережі такі як Facebook, Instagram, TikTok та інші популярні додатки, які збирають інформацію користувачів, їхні пости і коментарі для навчання ШІ.

Всі ми звикли бачити капчу при вході в аккаунт, яка просить підтвердити, що ви не робот, ввівши текст з картинки, обравши світлофори, тощо. Річ у тому, що вона також використовується для збору даних. Вводячи символи з картинки, ви не тільки підтверджуєте, що ви не робот, але і допомагаєте ШІ навчитися розпізнавати тяжко зрозумілий текст. За допомогою цього він потім може розшифровувати написи зі старих документів, які вже навіть людина розібрати не може. Тепер, коли ШІ давно вміє знаходити текст на картинках, капчі стали по типу: знайдіть всі світлофори, автобуси, велосипеди, схожі силуети тощо. Це все допоможе використовувати ШІ на камерах та в автомобілях для розпізнавання важливих речей на дорозі і в інших сферах життя.

# РОЗДІЛ ІІ

# ПРОЕКТУВАННЯ ФУНКЦІОНАЛУ ДЛЯ ЧАТ-БОТУ

2.1. Підготовка технічного функціоналу чат-бота

Ми вже розібралися з теоретичною частиною аналізу природньої мови. Тепер перейдемо до написання функціоналу для чат боту. Напишемо спочатку окремо функції, що безпосередньо будуть відповідати за аналіз природньої мови і видачі результатів, а в наступному розділі будемо створювати телеграм бота і просто «насаджувати» функціонал. Нам пощастило, тому що зараз в інтернеті вже є багато навченних моделей, які ми можемо використовувати у своїх програмах. Оскільки навчання моделей з нуля потребує багато часу і даних, я буду використовувати моделі від Hugging Face.

Hugging Face - це платформа, яка спеціалізується на розробці та розповсюдженні готових моделей для аналізу природньої мови, зображень, ауідо та відео з відкритим кодом. Ця платформа надає доступ до широкого спектру передових моделей, включаючи BERT, GPT, RoBERTa та інші, які були навчені на великих наборах даних.

Як я вже казав раніше, використання готових моделей з Hugging Face дозволяє нам значно економити час та ресурси, оскільки ми можемо використовувати вже навчені та оптимізовані моделі замість власного навчання з нуля. Крім того, це дозволяє нам використовувати найновіші досягнення в галузі NLP без необхідності великої експертизи у цій області.

Додамо наступний функціонал, до чат боту:

1. Визначення настрою повідомлення (Sentimental Analyse)

Використовуючи модель «cointegrated/rubert-tiny-sentiment-balanced» створемо функцію, яка, аналізуючи наданий текст, буде видавати одну з трьох відповідей: positive, negative, meutral. Нащастя, на офіційному сайті Hugging Face разом із моделями часто надано інструкцію користування. Зазвичай алгоритм використання моделей NLP від Hugging Face виглядає наступним чином:

1. Імпортуємо потрібні модулі із бібліотеки transformers (попередньо завантаживши її. Це можна зробити у термінали використавши команду “pip install transformers”).

1. Завантажуємо модель та токенайзер (він потрібен для того перетворити дані так, щоб модель могла їх якісно обробити)
2. Додаткові дії з даними/моделлю (функціонально)
3. За допомогою токенайзера готуємо дані для моделі і безпосередньо надаємо їй їх.
4. Модель обробляє попередньо приготовані для неї дані і видає результат.
5. Отримавши результат, дістаємо потрібну нам інформацію та використовуємо у своєму коді.

Але цей алгоритм частіше правдивий для тих моделей, які розраховані на небагато мов і прості в використанні. Нажаль, на разі порівняно дуже мало моделей заточених під українську мову. А ті, які підтримуються її, можуть дуже не точно працювати. По можливості, я буду використовувати ті, які можуть працювати з українскьою. Інакше – англійська.

Див додаток Б, Рисунок Б. 1. - код з реалізацією аналізу настрою повідомлення.

1. Класифікація на токсичність тексту (Toxicity Classificator)

Використовуючи модель «cointegrated/rubert-tiny-sentiment-balanced», створемо функцію, яка, аналізуючи наданий текст, буде видавати одну з двох відповідей: kindly/toxicity. Така функція може бути корисна, наприклад, для перевірки коментарів у соціальних мережах, сайтах тощо.

Код з реалізаціїю Див додаток Б, Рисунок Б. 2.

1. Вставлення слова у контекст

Використовуючи модель «bert-base-multilingual-uncased», створимо функцію, яка, аналізуючи наданий текст, може підбирати для вибраного місця слово, яке вписується у контекст. На реальній практиці є ясркавий приклад схожого функціоналу – Т9 у смартфонах, який над клавіатурою пропонує слова. Також таку модель можна використовувати, наприклад, при обробці якихось важливих та пошкоджених документів. Можливо, ШІ зможу вгадати слово, яке неможливо розібрати.

Будемо позначати місце, де треба вставити слово як «?». Код реалізації і результат див. у додатку Б, Рисунок Б. 3; Рисунок Б. 4.

1. Відповідь на питання по наданому контексту

Використовуючи модель «timpal0l/mdeberta-v3-base-squad2», стоврему функцію, яка зможе аналізувати наданий текст та надавати ту інформацію, яка в ньому написана. Це дуже корисна модель, оскільки дозволяй економити час та сили. Просто надаємо текст і задаємо по ньому питання.

Код реалізації і результат див. у додатку Б, Рисунок Б. 5; Рисунок Б. 6. Зауважте, що в цьому випадку ми використовуємо функцію pipeline.

1. Опис зображень (Оскільки для того, щоб описати зображення, теж використовуютсья моделі, навчані окрім того що розпізнавати об’єкти на фото, так ще й правильно описувати їх в зрозумілій для нас формі)

Використовуючи модель «Salesforce/blip-image-captioning-large», створемо функцію, яка на вході буде отримувати зображення, а на виході будемо мати опис цього зображення. Тут вже окрім NLP використовуються технології комп’ютерного зору.

Якщо хочете скористатися компьюерним зором, не забудьте завантажити модуль pillow (pip install pillow).

Код реалізації, результат і описуване зображення див. у додатку Б, Рисунок Б. 7; Рисунок Б. 8; Рисунок Б. 9.

# РОЗДІЛ ІІІ

# СТВОРЕННЯ ЧАТ-БОТУ ТА ДОДАВАННЯ ДО НЬОГО ЗАЗДЕЛЕГІТЬ ПІДГОТОВЛЕНОГО ФУНКЦІОНАЛУ

## 3.1. Плюси та недоліки чат-ботів

Перед створенням чат-боту, перелічимо плюси та недоліки чат-боту.

Перше, що приходить на думку – це автоматизація рутинних завдань, а також цілодобова підтримка, що дає змогу економити людські ресурси і перенаправляти їх на більш складні завдання. Це також може зекономити гроші компанії, оскільки чат боту не треба платити кожен місяць зарплату і він може спілкуватися одночасно з тисячами клієнтів, заміняючи десятки, а то і сотні працівників. Спілкування з чат-ботом – це зручно та швидко. Йому, зазвичай, не треба часу на «подумати» і він швидко видає відповідь. Ще один плюс чат-ботів – це можливість автоматизовано збирати дані клієнтів, а також впровадити ШІ, який допоможе, наприклад, підібрати послугу, підібрану спеціально для даного користувача.

Але майже до кожного плюсу можна дібрати відповідний недостаток. По-перше, чат бот має обмежений функціонал і якщо запитання користувача не стандартне або потребує більш складного аналізу, частіше за все, спілкування зводиться до справжнього оператора-консультанта. По-друге, чат-боти потребують постійного технічного обслуговування. Наприклад: додати новий функціонал, полагодити збій, затрати на сервер тощо. І для вирішення цих всіх задач потрібно наймати спеціалістів, які розбираються у даних сферах.

Отже, чат-бот це всього лише робот, який не має емоціонального інтелекту і часто не взмозі вирішувати складніші завдання за рутинні, а також іноді можуть бути збої, тому він потребує догляду. Але не дивлячись на усі мінуси, як і раніше, чат-боти залишаються зручнимм і вигіднимм для використання. Вони мають попит і їх постійно використовує багато популярних компаній на практиці.

## 3.2. Інструкція створення власного чат-бота у телеграмі

Перед тим, як переходити до написання коду, треба створити самого бота і отримати спеціальний токен для взаємодії з API телеграму.

1. Пишемо @BotFather (<https://t.me/BotFather>) команду ‘/help’.

Див. додаток В, Рисунок В. 1. (скріншот переписки)

1. Обираємо команду /newbot.

Див. додаток В, Рисунок В. 2. (скріншот переписки)

1. Вказуємо ім’я, яке буде відображатися у бота біля аватарки.

Див. додаток В, Рисунок В. 3. (скріншот переписки)

1. Вказуємо юзернейм, по якому можна буде знаходити бота та звертатися до нього. Ім’я повинно бути написано англійськими літерами і закінчуватися на «bot» (наприклад, назвемо його proanalyser2000\_bot).

Див. додаток В, Рисунок В. 4. (скріншот переписки)

1. Отримуємо посилання на бота та токен, який буде нам потрібен для функціювання чат-боту. Важливо! Нікому не показуйте його, в інакшому випадку стороння особа зможе керувати вашим чат-ботом.

Див. додаток В, Рисунок В. 5. (скріншот переписки)

Також за допомогою даного бота-батька можна редагувати ваші телеграм-боти: змінювати ім’я, фото, опис і так далі. Всі функції можна подивитися за допомогою команди ‘/help’.

## 3.3. Додаємо до чат-боту функціонал з можливостю аналізу природньої мови використовуючи бібліотеку aiogram

### 3.3.0. Вибір фреймворка для чат-боту

Найпопулярніші бібліотеки для створення телеграм ботів на мові програмування Python – це telebot і aiogram.

telebot вважається трохи легшим і популярним серед початківців, через те, що він є досить простим у використанні і має зрозумілу документацію. Він має основні функції для створення ботів, такі як обробка повідомлень, відповіді на команди, зміна клавіатури та інше.

aiogram, з іншої сторони, є більш потужним і розширеним фреймворком для створення телеграм-ботів на Python`і. Він базується на асинхронному програмуванні, що робить його швидшим та ефективнішим у використанні для великих проектів. Крім того, aiogram має більше можливостей, таких як робота з клавіатурами, опитуваннями, відправка медіафайлів, інтеграція з базами даних та іншими сервісами.

Я обрав aiogram через його більшу функціональність та можливості для розширення проекту у майбутньому. Він надає більше контролю над розробкою та забезпечує більшу гнучкість у реалізації різних функціональностей у телеграм-боті. Насправді, aiogram i telebot - схожі у синтаксисі. Якщо ви вже володієте мовою програмування, то обидві бібліотеки для вас не будуть занадто складними. Головне - прагнення розвиватися і покращувати свої навички.

Якщо ви не знайомі з асинхронністю, перед написанням бота на цій бібліотеці, рекомендую ознайомитися з іншою бібліотекою під назвою async.

### 3.3.1. Створення базового ехо-бота

Проаналізуємо код базовго ехо-бота (див. додаток Г, Рисунок Г. 1.) :

1. Імпортуємо потрібні функції/класи з модуля aiogram.
2. Створюємо екземпляр класа Bot (вказавши отриманий TOKEN\_API).
3. Створюємо на основі нашого Бота екземпляр класу Dispatcher.
4. По бажанню створюємо функцію on\_startup, яка буде визиватися при запуску бота.
5. Асинхронна функція echo з декоратором, завдяки якому вона буде спрацьовувати при будь-якому повідомленню від користувача і відповідати цим самим повідомленням (звідси і назва – ехо-бот).
6. Параметри запуску бота.

Результат тестування див. додаток Г, Рисунок Г. 2.

Якщо хочете, щоб бот просто відправляв ехо-повідомлення без «відмітки» вас, замініть функцію message.reply() на message.answer() (див. додаток Г, Рисунок Г. 3.).

Зауважте, що після зміни коду, бота треба перезапускати, щоб зміни застосувались.

### 3.3.2. Створення команди /help і заміна клавіатури

У минулому підпункті, наш чат-бот реагував на всі види повідомлень. Але message\_handler (Обробник повідомлень), також може реагувати лише на певні. Для цього йому треба вказати відповідний параметр:

* сommands = [“start”,] – ця функція буде переходити в виконання тільки у випадку, якщо користувач викличе у чаті команду /start (навіть якщо повідомлення має додатково текст через пробіл після команди).
* content\_types = [“text”, ] – реагує тільки на тестові повідомлення
* lambda x: ”Україна” in x - так, можна навіть використовувати лямбда функції та фільтри (як ті, які надаються фреймворком, так і ті, що створите самі).

Замінемо нашу ехо-функцію на іншу, яка буде реагувати на команди /start, /help. Код реалізації та тестування див. у додатку Г, Рисунок Г. 4; Рисунок Г. 5.

Тепер прийшов час зробити вивід нашого функціоналу, а також заміни клавіатури для зручного викрику функцій однім кліком по кнопці.

В aiogram є декілька видів кнопок:

* KeyboardButton (які «насаджуються» на екземпляр клавіатури ReplyKeyboardMarkup. Ці кнопки будуть завжди доступні у додатковій клавіатурі, яку можна закрити та відкрити у будь-який момент.)

Див. приклад у додатку Г, Рисунок Г. 6.

* InlineButton (з’являються прямо у чаті, прив’язані до одного повідомлення та підправляють на сервер callback запрос)

Див. приклад у додатку Г, Рисунок Г. 7.

Нас цікавить перший варіант. Як для прикладу, додамо лише кнопку /help, щоб у будь-який момент можна було знову переглянути команди. Код у файлі config.py див. у додатку Г, Рисунок Г. 8.

Імпортуємо кнопку у головний файл з ботом, і замінюємо клавіатуру при відправленні першого повідомлення (Код функції і результати тестування див. у додатку Г, Рисунок Г. 9; Рисунок Г. 10.)

### 3.3.3. Обробка всіх команд. Застосування заготовлених функцій

Тепер ми вміємо створювати функції і робити так, щоб вони відкликалися на відповідні команди. Прийшов час «прикрутити» функціїї з NLP до нашого боту. Всі наступні функції матимуть дуже схожий алгоритм, а саме:

1. Відокремлюємо головний текст запросу.
2. Перевіряємо, щоб текст був не пустим.
3. Передаємо текст функції, яка поверне результат.
4. Виводимо результат користовачу.

Код із застосуванням функції аналізу настрою повідомлення і результати тестування див. у додатку Г, Рисунок Г. 11.; Рисунок Г. 12.

Далі робимо подібні функції. Звісно, в справжньому проекті такі функції треба виносити в окремий файл і оптимізувати, слідуючи принципам DRY, але в навчальних цілях я цього робити не буду. На гітхабі можна переглянути код всього проекту (посилання у висновку). Якщо вам не сподобається робота деяких моделей, які погано працюють з українською мовою, можете замінити їх або скоригувати вивід.

Тестуємо роботу інших функцій:

* /toxicity\_check (див. додаток Г, Рисунок Г. 13.)
* /answer\_by\_context (див. додаток Г, Рисунок Г. 14.)
* /word\_in\_context: (див. додаток Г, Рисунок Г. 15.).

Як бачимо, на українскій працює трохи гірше.

* /describe\_photo : (див. додаток Г, Рисунок Г. 16; Рисунок Г. 17.)

### 3.3.4 Удосконалення проекту

Як вже казав раніше, можна зробити рефакторинг коду. А далі – ваша фантазія. Наприклад, за допомогою FSM (кінцевий автомат стану), можна зробити так, щоби користувач спочатку вводив команду, а наступним повідомленням окремо писав текст. Можна зв’язати телеграм бота із базою даних, щоб збирати інформацію про запроси користувачів. І, звісно, можна додавати нові можливості і команди.

# ВИСНОВОК

В наші часи вже майже у кожної великої компанії є телеграм бот, який використовуюється в самих різних цілях. Чат-бот – це не тільки цікаво, але й корисно. Давайте узагальнемо, чого ми дізналися:

* Де застосовується аналіз природньої мови – сфери бізнесу, науки, медицини, сфери розваг та безпеки.
* В чому проблема аналізу природньої мови – спрощення, неоднозначність, знаки пунктуації, особливості окремої мови тощо.
* Методи створення та навчання моделей та алгоритм графематичного аналізу – глибоке навчання, навчання з учителем, навчання з підкрипленням тощо.
* Познайомилися з Hugging Face - платформа, яка спеціалізується на розробці та розповсюдженні готових моделей для аналізу природньої мови і не тільки.
* За допомогою моделей взятих у Hugging Face, створили функції для майбутньої інтеграції їх у проект.
* Розглянули плюси та недоліки чат-ботів.
* Створили ехо-бота і навчилися додавати команди.
* Інтегрували попередньо заготовлений функціонал до телеграм боту.
* Розглянули, як можна покращити даний проект.

Кому цікаво по експериментувати з проектом, тримайте посилання на

гітхаб – https://github.com/Youngertss/Analyser\_TGbot.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Друковані джерела:

1. Руденко В.Д. 2018. Підручник Інформатика 10 клас (профільний рівень)

2. Руденко В.Д. 2018. Підручник Інформатика 11 клас (профільний рівень)

Інтернет джерела:

1. https://towardsdatascience.com/

2. https://medium.com/

3. Dovidka.biz.ua https://dovidka.biz.ua/rechennya-z-omonimamy-pryklady

4. https://ai.meta.com/

5. https://paperswithcode.com/method/wordpiece

6. https://pypi.org/

7. https://evnuir.vnu.edu.ua/bitstream/123456789/7845/1/41.pdf

8. <https://ukr.ed-era.com/fonetika_grafka_orfoepya>

9. <https://sharkdevelop.com/boty-v-telegram/>

10. <https://docs.aiogram.dev/en/latest/>

11. <https://igorzuevich.com/fakty-o-telegramm-7-unikalnyx-xarakteristik/>

12. <https://core.telegram.org/api/bots>

13. <https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm>

14. <https://zapier.com/blog/best-ai-chatbot/>

# Додатки

Додаток А

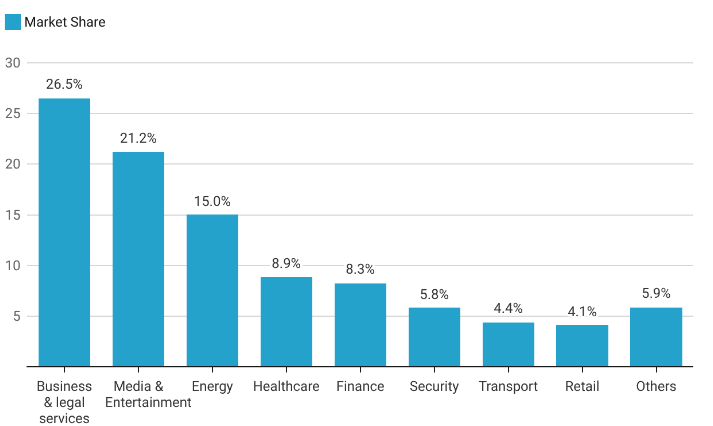


Рисунок А.1. Статистика використання аналізу природньої мови у різних сферах

Додаток Б

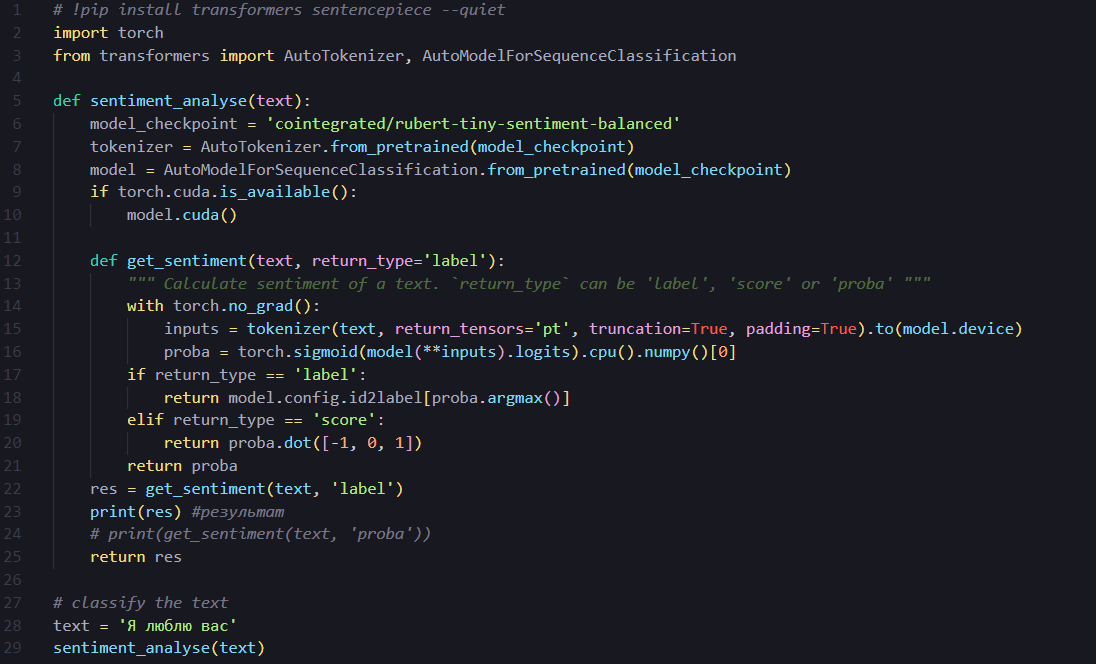


Рисунок Б. 1. Код визначення настрою повідомлення*.*Результат виконання: positive.

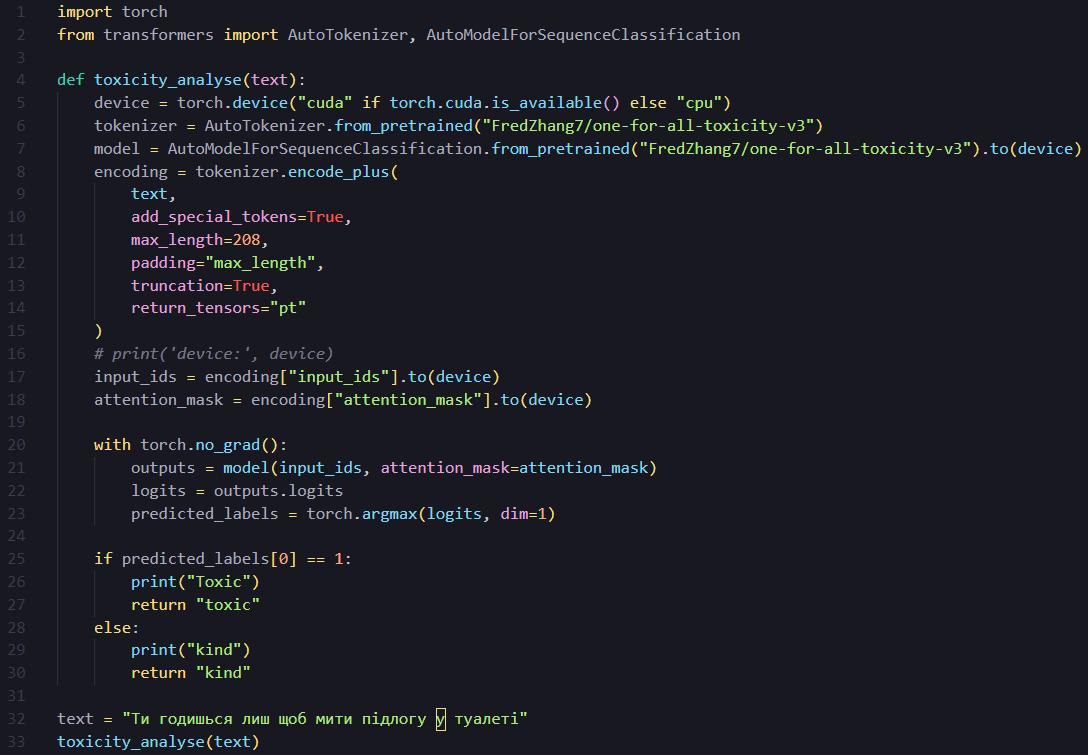


Рисунок Б. 2 Код класифікація на токсичність тексту

Результат виконання: Toxic.



Рисунок Б. 3. Вставлення слова у контекст



Рисунок Б. 4. Результат вставлення слова у контекст (топ 5 слів)

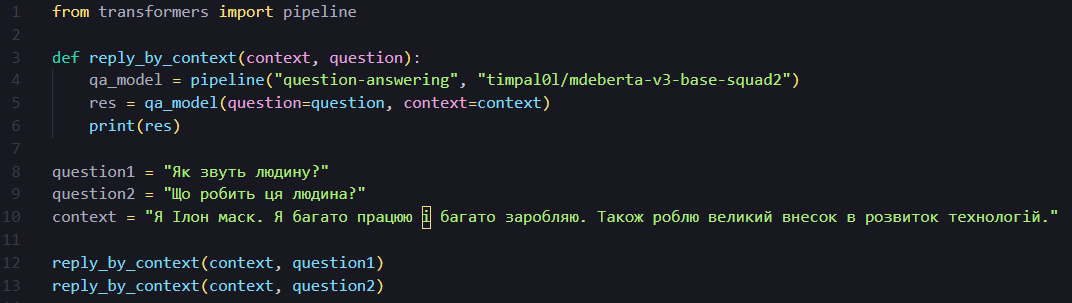


Рисунок Б. 5. Відповідь на питання по наданому контексту



Рисунок Б. 6. Результат виконання відповіді по контексту (текстова відповідь знаходиться за ключем answer)



Рисунок Б. 7. Код опис зображення

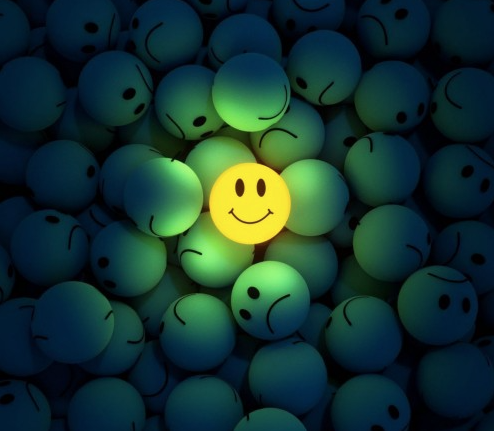


Рисунок Б. 8. Зображення для опису



Рисунок Б. 9. Результат опису зображення

Додаток В



Рисунок В. 1. Команда /help у BotFather

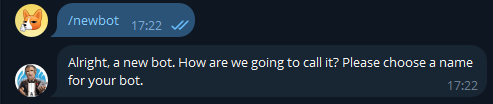


Рисунок В. 2. Команда /newbot у BotFather



Рисунок В. 3. Вводимо нікнейм нового бота



Рисунок В. 4. Вводимо юзернейм новго бота

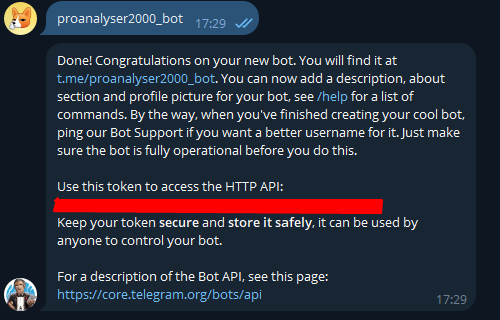


Рисунок В. 5. Текст успішного створення бота та отримання посилання на нього із TOKEN\_API

Додаток Г

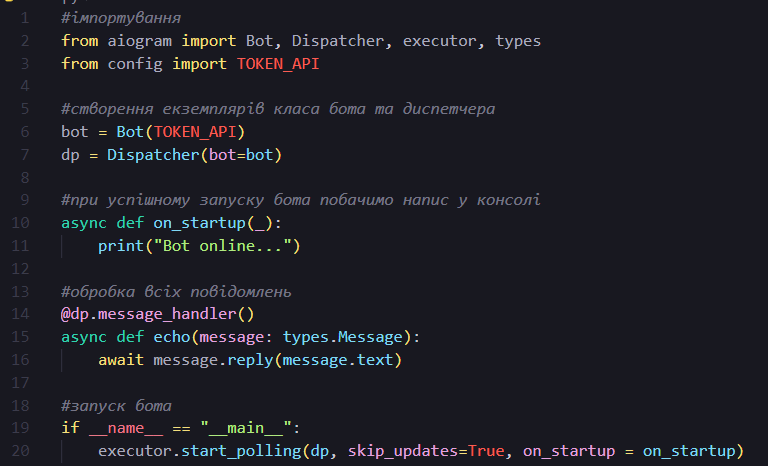


Рисунок Г. 1. Код базового ехо-бота



Рисунок Г. 2. Тестування ехо-бота



Рисунок Г. 3. Тестування ехо-бота без «відмітки» користувача

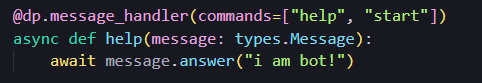


Рисунок Г. 4. Функція, що реагуєна команди /help і /start



Рисунок Г. 5. Тест реагування бота на команди

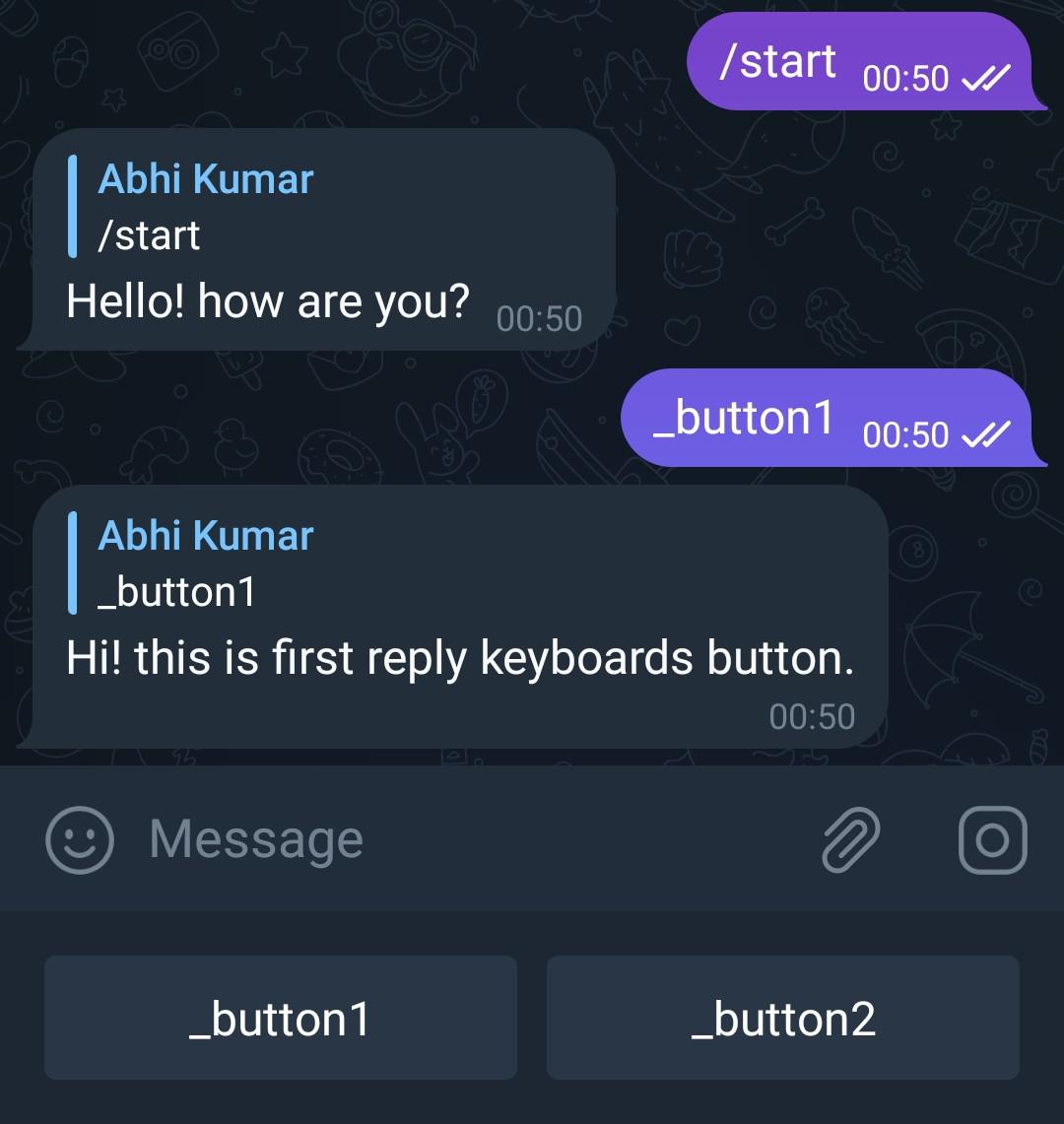


Рисунок Г. 6. Приклад ReplyKeyboardMarkup

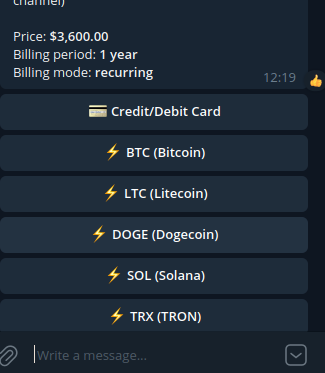


Рисунок Г. 7. Приклад InlineKeyboard

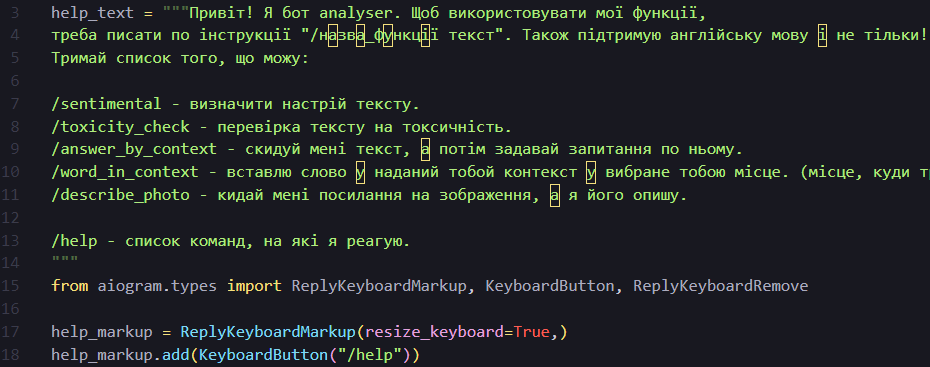


Рисунок Г. 8. Код з файлу config.py

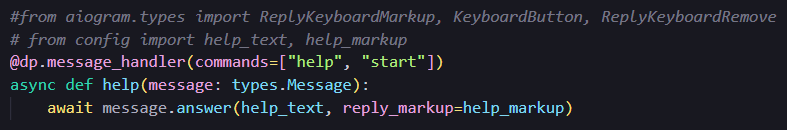


Рисунок Г. 9. Остаточна версія функції виводу можливостей бота

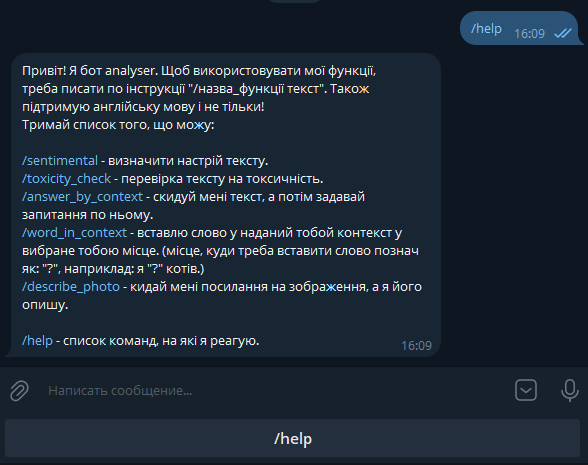


Рисунок Г. 10. Тест виводу можливостей бота

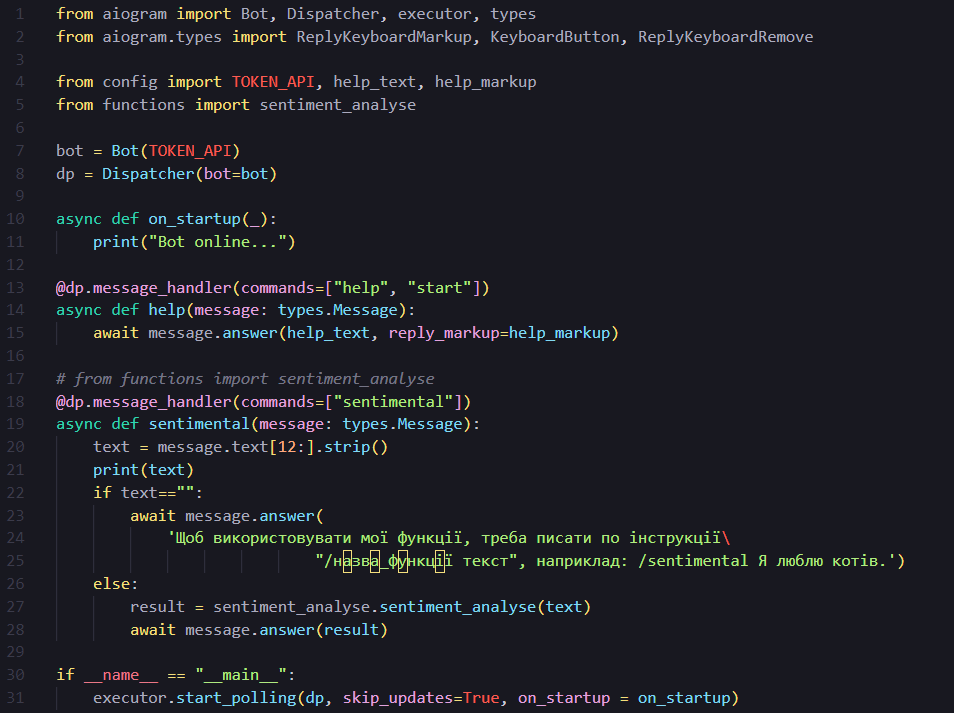


Рисунок Г. 11. Код функції аналізу настрою повідомлення (18-28 строки)



Рисунок Г. 12. Тест команди /sentimental

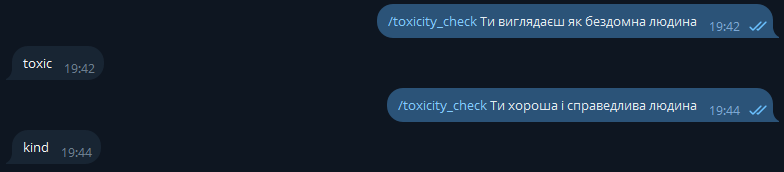


Рисунок Г. 13. Тест команди /toxicity\_check

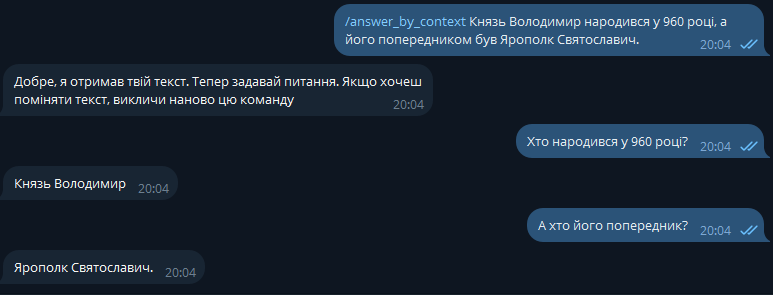


Рисунок Г. 14. Тест команди /answer\_by\_context:

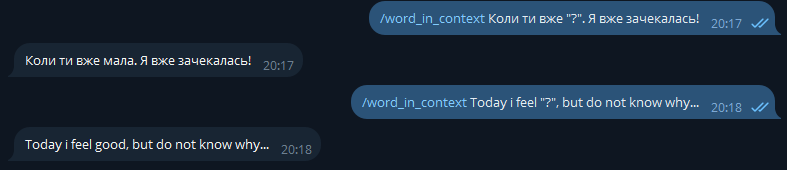


Рисунок Г. 15. Тест команди /word\_in\_context

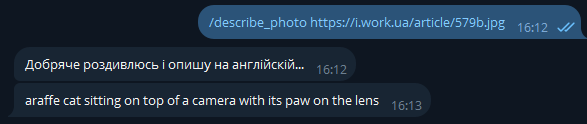


Рисунок Г. 16. Тест команди /describe\_photo



Рисунок Г. 17. Зображення для тесту опису