Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук

Кафедра штучного інтелекту

# РЕФЕРАТ

на тему «Бот Dvango»

з дисципліни «Системи інтелектуальної обробки природно-мовної інформації»

Виконали ст. гр. ІТШІ-18-1:
Соколенко Дмитро Олександрович
Апраксін Антон Романович

Прийняв:
проф. каф. ШІ Рябова Н. В.
з оцінкою “_____”
“___”_____20___р

Харків 2021

**ЗМІСТ**

# ВСТУП

Дана робота присвячена побудові чат-боту, що відповідає на запитання користувача, які побудовані природною мовою. Бот не підтримує діалог на побутові теми, тобто йому не можна написати “Hello” або “How are you?”. Але бот дуже гарно (як для бота) відповідає на запитання, що поставив користувач.

Ми живемо в епоху інформаційних технологій, які увійшли у життя кожного з нас. Кількість інформації зростає з кожним днем і знаходити відповіді на питання, що вас цікавлять стає складніше. З іншого ж боку, технології машинного навчання теж зростають так швидко, як ніяк раніше. І це призводить нас до концепту: використати сучасні технології для розв’язання сучасних проблем. Тобто ми можемо побудувати таку систему, що буде знаходити відповіді на питання, що нас цікавлять. І робити система це може інтерактивно, тобто у форматі чат-боту, якому ви пишете повідомлення, яке містить питання і на яке бот якось змістовно відповідає.

Дуже багато компаній вже використовують такі технології, бо вони допомагають новим користувачам або клієнтам у використанні системи. Вони відповідають на базові питання про функціонування системи, і тим самим допомагають людям розібратися у роботі системи.

# ОСНОВНА ЧАСТИНА

## 1 Аналіз предметної області

Сучасні технології та архітектури штучних нейронних мереж дозволяють проводити успішний аналіз текстів, що написані природною мовою. Як приклад можна привести GPT-3 (Generative Pre-trained Transformer 3), яка була розроблена лабораторією OpenAI. Модель змогла написати текст, який було складно відрізнити від тексту, який написала б людина. Звісно, подібні технології можуть нести небезпеку, бо можуть використовуватися проти людей.

Але також ці технології успішно використовуються для допомоги людству. Це настільки просунуті та потужні системи, що немає жодного сенсу обмежувати подібні моделі якоюсь однією предметною областю. Замість цього, можна побудувати таку систему, яка буде шукати відповіді у зазначених базах знань. Потужність таких систем дозволяє виходити на рівень такий рівень абстракції, що не враховує конкретної теми запитання, а шукає відповідь семантично.

Звісно, щоб мати можливість навчати та запускати моделі такої складності необхідні великі потужності. Але можна використовувати вже навчені моделі. Наприклад, є моделі, що вже навчені на датасеті SQuAD або навіть на всій Wikipedia. Як базу знань для такої навченої моделі можна використовувати якусь статтю, або взагалі написати самому невеликий текст, по якому буде проводитися пошук відповіді. Потім написати декілька таких текстів, які насправді називаються контекстами, та обирати необхідний, який ми будемо обирати за ключовими словами у питанні.

Тобто, постановка задачі – побудувати таку систему, яка могла б легко розширюватися новими темами питань.

## 2 Порівняння до альтернатив

В ході пошуку ідей, які бібліотеки використовувати, в якій середі йому доведеться працювати ми зробили вибір. В якості “інтелекту” нашого бота ми використали бібліотеку transformers [5]. Ця бібліотека містить ве-

личезний набір нейронних мереж – трансформерів. Вони всі відрізняють за архітектурою, кількістю параметрів та, звісно, якістю. У нас не було можливості натренувати для своїх даних, тому було вирішено використовувати модель BERT-large, яка натренована на датасеті SQuAD. Це оптимальний вибір для системи, яка має бути здатна відповідати на питання.

Як альтернативні варіанти “інтелекту” нашої системи ми розглядали двигун для створення чат-ботів ChatterBot та дуже старий алгоритм ELIZA. ELIZA є дуже старим та неефективним рішенням, тому майже одразу був відкинутий. ChatterBot виглядав більш привабливо. В основі його лежить модифікований наївний Байес. В цілому, така модель може доволі успішно відповідати на заздалегідь запрограмовані запитання. Але набагато цікавіше було попрацювати з чимось більш сучасним. Тому залишився тільки варіант з трансформером.

В якості платформи, де система має працювати, був обраний месенджер Telegram через його розповсюдженість та розвинутість в плані функцій. Також, доступні можливості з легкого створення ботів. Враховуючи ці фактори, ми обрали саме цей месенджер.

В якості сервера використовувався фреймворк Flask через свій малий розмір та простоту розробки.

# 3 Технології та середа розробки

## 3.1 Мова програмування

Для написання роботи була використана мова програмування Python. Python – інтерпретована мова програмування загального призначення високого рівня. Філософія дизайну Python наголошує на читабельності коду завдяки помітному використанню значних відступів. Його мовні конструкції, а також об’єктно-орієнтований підхід мають на меті допомогти програмістам писати чіткий логічний код для малих та великих проєктів.

Python зосереджується на читабельності коду. Мова універсальна, акуратна, проста у використанні та вивченні, читабельна та добре структурована.

Завдяки гнучкості Python легко провести дослідницький аналіз да-

них. Також він дозволяє використовувати найкращі з різних парадигм програмування. Він об’єктно-орієнтований, але також має функції з функціонального програмування.

З переваг цієї мови можна виділити:

1. Відкритий код.

   Інтерпретатор Python можна завантажити безкоштовно та переглянути його вихідний код. Це також означає, що мова розвивається згідно з потребами суспільства програмістів, та не може раптово змінити курс розвитку. Саме ця особливість в парі с низьким порогом входження роблять її однією з найгнучкіших мов програмування, що зараз існують.

2. Проста мова.

   Через свою філософію, Python можна швидко вивчити та відразу розпочати розробляти програми. Бо в основі лежить головна ідея – зробити мову для звичайних людей.

3. Бібліотеки.

   Через те, що для написання коду не потрібно проводити довгі години, читаючи документацію та літературу Python, став дуже поширеним. Через це, з’явилася потреба у написанні гарних абстракцій, щоб можна було легко та без зусиль починати писати програмне забезпечення.

   Також мова програмування поставляється з менеджером пакетів Pip, який дозволяє дуже швидко та без зайвих проблем керувати пакетами, які необхідні при розробці. Є можливість швидкої установки, видалення, запису стану у файл та відновлення стану з файлу.

Головні недоліки мови:

1. Дуже повільна.

   Python – це інтерпретована мова, тому вона працює повільніше, ніж деякі інші популярні мови програмування.

2. Споживає багато ресурсів.

   Python робить компроміс заради простоти, тому споживання пам’яті у нього велике. Це може бути проблемою для задач, що потребують багато одночасно активних об’єктів в пам’яті, але не для нашої.

3. Відсутність системи типів.

   Типи допомагають перевірити, чи програма правильно побудована. Через динамічну природу Python, вона стає дуже вразливою до помилок часу виконання, що може призводити до непередбачуваних збоїв, які треба буде відновлювати.

Виходячи з приведених переваг та недоліків, та того факту, що для Python є бібліотеки, які гарно виконують задачі, які нам необхідні, ми обрали саме цю мову програмування.

## 3.2 Середа розробки

Текстові редактори.

Для написання коду було використано текстові редактори Visual Studio Code та Vim.

Visual Studio Code – це редактор вихідних кодів, створений Microsoft для Windows, Linux та macOS. Особливості включають підтримку налагодження, підсвічування синтаксису, інтелектуальне заповнення коду, сніпети, рефакторинг коду та вбудована підтримка Git. Користувачі можуть змінювати тему, комбінації клавіш, налаштування та встановлювати розширення, що додають додаткову функціональність.

Vim – це термінальний текстовий редактор. Це розширена версія vi з додатковими функціями, включаючи підсвічування синтаксису, всеосяжну систему довідок, власні скрипти (vimscript), візуальний режим для вибору тексту, порівняння файлів (vimdiff) та інструменти з обмеженими можливостями, такі як rview та rvim.

Обидва редактори можна легко настроювати та розширювати, що робить їх привабливими інструментом для розробників, які вимагають вели-

кої кількості контролю та гнучкості над своїм середовищем редагування тексту.

Середа виконання.

В якості середи виконання використовувалася програма virtualenv. Це програма для створення і управління оточеннями Python. Вона дозволяє створити середовище зі своїми окремими модулями, налаштуваннями й програмами. Середа обмежується рамками одного каталогу та дуже зручна для роботи з різними версіями одних і тих же модулів, для створення проєктів, у яких “все з собою”, які не залежать від операційної системи.

## 3.3 Мова розмітки для написання звіту

Для написання звіту було обрано програмне забезпечення для розмітки LaTeX. Це система високоякісної верстки; вона включає функції, призначені для виготовлення технічної та наукової документації. Також LaTeX є фактичним стандартом для передачі та публікації наукових документів. Система доступна як безкоштовне програмне забезпечення.

Під час письма використовується звичайний текст на відміну від форматованого тексту, який можна знайти в текстових процесорах, які побудовані за принципом “Що ти бачиш, те й отримуєш”, таких як Microsoft Word, LibreOffice Writer та Apple Pages. Письменник використовує теги розмітки, щоб визначити загальну структуру документа (наприклад, статтю, книгу та лист), стилізувати текст у всьому документі (наприклад, напівжирний шрифт та курсив), а також додати цитати та перехресні посилання. Дистрибутив TeX, такий як TeX Live або MiKTeX, використовується для створення вихідного файлу (наприклад, PDF або DVI), придатного для друку або цифрового розповсюдження.

LaTeX широко використовується в наукових колах [1] [2] для передачі та публікації наукових документів у багатьох галузях, включаючи математику, статистику, інформатику, техніку, фізику, економіку, лінгвістику, кількісну психологію, філософію та політологію. Він також відіграє помітну роль у підготовці та виданні книг та статей, що містять складні багатомовні матеріали, такі як санскрит та грецька мова. LaTeX використовує

програму набору TeX для форматування своїх вихідних даних і сама написана мовою макросів TeX.

LaTeX намагається слідувати філософії дизайну відокремлення презентації від змісту, щоб автори могли зосередитися на змісті того, що вони пишуть, не приділяючи одночасно його візуальному вигляду. Готуючи документ LaTeX, автор визначає логічну структуру, використовуючи прості, звичні поняття, такі як глава, розділ, таблиця, малюнок тощо, і дозволяє системі LaTeX обробляти форматування та компонування цих структур. Як результат, це заохочує відокремлення макета від змісту, дозволяючи при цьому коригувати ручне введення набору, коли це потрібно. Ця концепція схожа на механізм, за допомогою якого багато текстових процесорів дозволяють глобально визначати стилі для цілого документа, або використання каскадних таблиць стилів при стилюванні HTML-документів.

# 4 Опис програмного продукту

Програма представляє собою Telegram чат-бота, який відповідає на запитання користувача. Теми запитань мають бути заздалегідь описані в базі даних чат-бота.

## 4.1 Комунікація з серверами Telegram

Telegram – це безкоштовне програмне забезпечення для обміну повідомленнями. Telegram надає можливість створювати ботів – просто акаунти Telegram, якими керує програмне забезпечення, а не люди, і вони часто мають функції штучного інтелекту.

Комунікація між програмним забезпеченням та сервером Telegram виконується через простий HTTPS-інтерфейс, який називається “Bot API”. Є два варіанти використання інтерфейсу:

1. опитування – ПО з деяким проміжком часу відправляє на сервер запит на нові повідомлення. Якщо боту хтось написав з часу попереднього запиту, то сервер відправляє це повідомлення;

2. вебхуки – це визначений користувачем зворотний виклик через HTTP [3]. ПО реєструє своє доменне ім’я у сервісі, та, на відміну

від методу опитування, вже сервер відправляє запити до нашого програмного забезпечення. Слід мати на увазі, що для використання цього методу треба володіти доменним ім’ям. З переваг цього методу над опитуванням можна виділити зменшене споживання ресурсів [4] та зменшення часу на відповідь на повідомлення.

Повідомлення з сервера Telegram приходять у форматі “JSON”, який ми парсимо за допомогою вбудованих засобів мови Python.

Ми обрали метод комунікації вебхуків через переваги, що були названі вище. Вебхук необхідно встановити перед стартом бота за допомогою нашого скрипта на мові Bash. Передбачається, що сервер має у глобальній мережі доменне ім’я. В нашому випадку ми використовували DDNS. На адресу нашого сервера Telegram буде висилати POST-запити на кожне нове повідомлення. ПО має правильно обробити запит та повернути відповідь у формі добре сформованого JSON. Також, при налагодженні з’єднання необхідно мати SSL-сертифікати. Їх можна взяти в авторизованих центрах сертифікації, або згенерувати самостійно, що ми і зробили. На цьому етапі з’єднання між нашим сервером та серверами Telegram буде налагоджено, та чат-бот буде готовий відповідати на запитання.

## 4.2 Організація бази даних

База даних організована у вигляді файлів. У директорії src/contexts знаходяться yaml-файли, які читаються при завантаженні програми. Вони мають специфічну структуру:

- Поле *name*

  *Задає ім’я цього контексту. Має бути унікальним.*

- *Поле file*

  *Задає ім’я файлу, який зберігає сам контекст.*

- *Поле synonyms*

  *Задає список синонімів або ключових слів, за якими буде здійснений пошук.*

Приклад:

Кожному файлу відповідає контекст, який вже містить текст для відповідей. Контексти знаходяться в окремих файлах, та не мають визначеної структури, тому можуть містити звичайні речення про обрану тему.

Приклад контексту можна бачити нижче:

## 4.3 Процес формування відповіді

Як бібліотеку для знаходження відповідей була обрана transformers. Це бібліотека, в якій зібрано багато архітектур різних трансформерів у зручному до використання виді. Також бібліотека надає можливість використовувати вже навчені моделі.

Нам довелося використовувати вже натреновану модель, бо в нас не було фізичної можливості навчити свою модель.

Ми обрали модель “bert-large-uncased-whole-word-masking-finetuned-squad”. Це модель вдалої моделі BERT-large (Bidirectional Encoder Representations from Transformers), який навчений на SQuAD (Stanford Question Answering Dataset). Для нашої задачі він підходить, хоча навчити було б краще та цікавіше.

Через свій розмір та складність подібну модель складно використовувати одну. Якщо просто дати їй величезний контекст, то на відповідь можна чекати понад 10 секунд, що занадто багато, як для чат-бота. Тому в парі з нейронною мережею ми використовували пошук за ключовими словами. В нашому випадку це алгоритм FlashText [6] та бібліотека [7] з однойменною назвою. Цей алгоритм працює на префіксному дереві та алгоритмі Ахо-Корасік [8], що дозволяє працювати йому дуже швидко. Тобто, пошуком ключових слів ми знаходимо тему запитання і вже відповідь шукаємо у контексті, що відповідає даній темі.

## 4.4 Концептуальна схема нейронної мережі

Трансформер складається з двох компонентів: енкодер та декодер. Енкодер перетворює вхідні дані на корисну інформацію, яка далі використовується для передачі у декодер. Декодер поступово формує вихід. Перший токер, що генерується – <start>. Далі, з кожним наступним токеном формується відповідь. Формування закінчується коли генерується токен

<end>.

В основі трансофрмеру лежить концепт Attention Layer. Це шар, що знаходить співвідношення слів у реченні.

# ВИСНОВКИ

Нами був розроблений бот для месенджеру Telegram, що відповідає на запитання. Такі рішення зараз є дуже популярними, бо можуть заміняти людей, відповідаючи на базові питання. Це приваблива сторона таких рішень для бізнесу. Для науковців такі системи мають академічний інтерес. Всі хочуть зробити таку систему, яка б пройшла тест Тьюринга.

В якості базового алгоритму була використана нейронна мережа з архітектурою трансформера BERT-large, натренована на датасеті SQuAD. В якості допоможного алгоритму використовувався алгоритм FlashText, який працює на префіксному дереві. Бот відповідає на питання, контекст яких є заданим у базі даних.

Дана робота дає лише базові уяви про те, як подібні системи працюють. Але навіть на такому прикладі видно, що необхідно використовувати декілька різних алгоритмів, шукаючи компроміс між якістю та швидкістю.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

[1] What are TeX and its friends? [Електронний ресурс] – Режим доступу до ресурсу: https://www.ctan.org/tex.

[2] Gaudeul A. Do Open Source Developers Respond to Competition? The LATEX Case Study / Alex Gaudeul. // Review of Network Economics. – 2007.

[3] Webhooks [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://developer.atlassian.com/server/jira/platform/webhooks/.

[4] Lindsay J. Web hooks to revolutionize the web [Електронний ресурс] / Jeff Lindsay. – 2007. – Режим доступу до ресурсу: https://web.archive.org/web/20180630220036/http://progrium.com/blog/2007/05/03/web-hooks-to-revolutionize-the-web/.

[5] State-of-the-art Natural Language Processing for Jax, PyTorch and TensorFlow [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/huggingface/transformers.

[6] Replace or Retrieve Keywords In Documents at Scale [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: https://arxiv.org/abs/1711.00046.

[7] FlashText module [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/vi3k6i5/flashtext.

[8] Aho–Corasick algorithm [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Aho

# ДОДАТОК 1