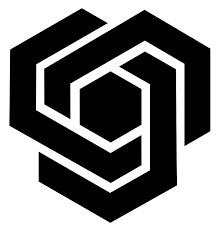
**Технически университет - София**

Факултет по Индустриални технологии

**Курсов Проект**

**Дисциплина: Проектиране и моделиране на данни**

**TRAVEL CHATBOT**

**Студент:**

Калина Катрапeлова

361222009

ИСИИ, III курс

**Преподавател:**

доц. д-р инж. Диляна Будакова

София, януари 2025

Съдържание

[Цел на проекта 3](#_Toc188363016)

[Компоненети: 3](#_Toc188363017)

[Поток на данните 4](#_Toc188363018)

[Процес на обучение 4](#_Toc188363019)

[Процес на генериране на отговори 4](#_Toc188363020)

[Поддържани функции 5](#_Toc188363021)

[Тхнически изисквания 5](#_Toc188363022)

[Примери 5](#_Toc188363023)

# Цел на проекта

Чатботът за пътуване е AI-базиран разговорен агент, създаден да предоставя информация за пътувания в три основни града: Париж, Лондон и Токио. Чатботът използва невронна мрежа, за да разбира потребителските въпроси и да предоставя подходящи отговори за забележителности, храна, време и обща информация за пътуването.

# Компоненети:

**Невронна мрежа (model.py)**

* **Реализация:** Невронна мрежа, базирана на PyTorch
* **Архитектура:**
  + Входен слой: Съобразява размера на входа с речника
  + Два скрити слоя с ReLU активация
  + Изходен слой: Брой класове (категории на намеренията)
  + Без финална активационна функция или softmax във функцията forward

**Подпомагащи функции за обработка на естествен език (nltk\_utils.py)**

* **Ключови функции:**
  + tokenize(): Разделя изреченията на думи чрез NLTK
  + stem(): Намалява думите до кореновата им форма с помощта на Porter Stemmer
  + bag\_of\_words(): Превръща токенизирани изречения в числови вектори

**Модул за обучение (train.py)**

* **Характеристики:**
  + Клас ChatDataset за обработка на данни
  + **Хиперпараметри:**
    - Размер на партида: 8
    - Размер на скритите слоеве: 8
    - Скорост на учене: 0.001
    - Епохи: 1000
  + **Съхранение:**
    - Тегла на модела (model\_weights.pth)
    - Конфигурация на модела (model\_data.json)

**Интерфейс за чат (chat.py)**

* **Имплементация:**
  + Клас SimpleChatBot:
    - Зарежда обучен модел и намерения
    - Генерира отговори в реално време
    - Включва праг на увереност (0.75) за отговори

**Конфигурация на намеренията (intents.json)**

* **Структурирани данни, съдържащи:**
  + Категории на намеренията
  + Обучителни шаблони
  + Шаблони за отговори
* **Обхванати категории:**
  + Информация, свързана с градове (Париж, Лондон, Токио)
  + Общи теми за пътуване
  + Основни разговорни елементи

# Поток на данните

1. Вход от потребителя → **Токенизация** → **Стеминг** → **Конвертиране в bag of words**
2. Обработен вход → Невронна мрежа → Прогнозиране на намерения
3. Намерение + Проверка на увереност → Избор на отговор → Изход

# Процес на обучение

**Подготовка на данни:**

* Зареждане на намеренията от JSON
* Обработка на шаблони в токени
* Създаване на тренировъчен набор от данни

**Обучение на модела:**

* Инициализиране на невронната мрежа
* Обучение с CrossEntropyLoss и Adam оптимизатор
* Запазване на състоянието и конфигурацията на модела

# Процес на генериране на отговори

1. **Обработка на входа:**
   * Токенизиране на входа
   * Създаване на bag of words
2. **Прогноза от невронната мрежа**
3. **Проверка на праг на увереност (>0.75)**
4. **Случаен избор на отговор от съвпадащото намерение**

# Поддържани функции

**Информация за конкретни градове:**

* Забележителности и атракции
* Местна кухня и хранене
* Време и най-добри периоди за посещение

**Обща информация за пътуване:**

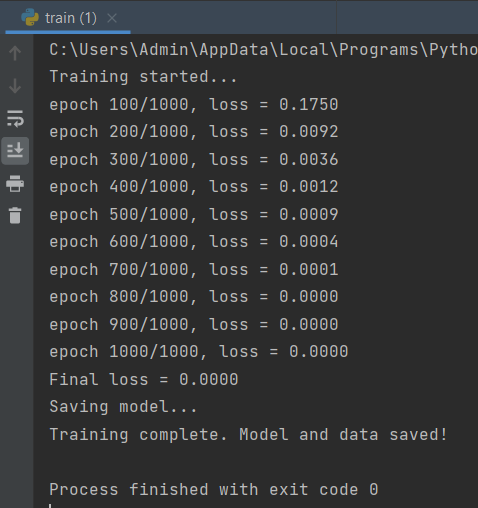
* Транспорт
* Разходи за пътуване
* Основни разговорни взаимодействия

# Тхнически изисквания

* Python 3.x
* PyTorch
* NLTK
* NumPy
* Поддръжка на JSON

# Примери

Трениране на модела, използвайки файл intents.json



Разговор с агента:

