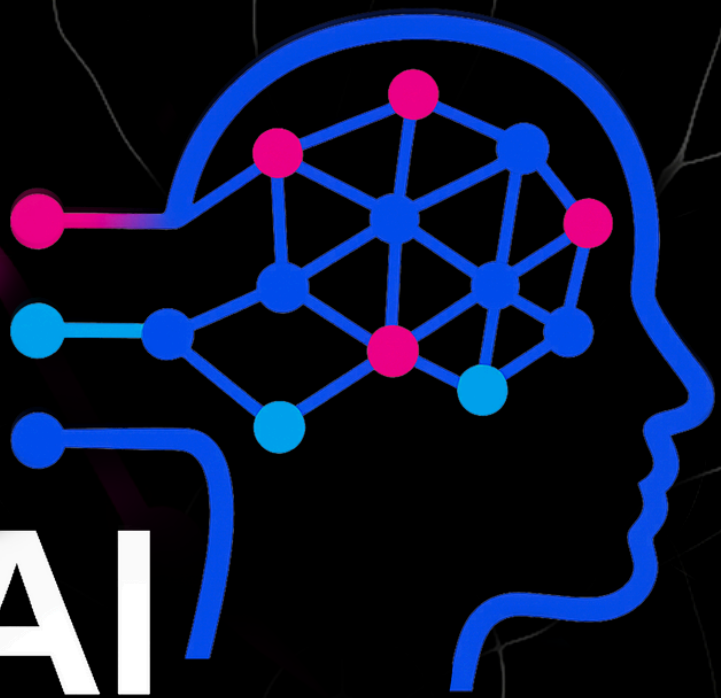

AI KRAK HACK

30/31 MAY 2025

Materiały przygotowawcze do wyzwania

Zaprojektuj Inteligentnego Asystenta Wydarzeń kulturalnych dla Krakowa

AI KRAK HACK

Przed rozpoczęciem pracy nad projektem warto zapoznać się z kilkoma obszarami tematycznymi i narzędziami, które mogą okazać się pomocne. Poniżej zebrano krótki przegląd zagadnień, technologii oraz bibliotek, których znajomość ułatwi realizację zadania:

- Przetwarzanie języka naturalnego (NLP) – metody reprezentacji tekstu w formie wektorów (embeddingi) oraz miary podobieństwa tekstów (np. miara cosinusowa) pozwalają porównywać opisy wydarzeń z preferencjami użytkownika. Warto zgłębić rozpoznawanie intencji wypowiedzi (intent recognition), aby lepiej rozumieć pytania i prośby użytkownika, oraz rozpoznawanie nazwanych encji (NER) w tekstach – dzięki temu można automatycznie wyłuskać kluczowe informacje, takie jak nazwy miejsc, daty czy nazwy wydarzeń z opisów lub zapytań.

- Systemy rekomendacyjne – różne podejścia do rekomendowania treści. Poznaj metody content-based (oparte na podobieństwie cech obiektów, np. porównywanie opisów wydarzeń z profilem zainteresowań użytkownika) oraz collaborative filtering (wykorzystujące podobieństwo zachowań lub ocen wielu użytkowników). Zrozumienie zalet i ograniczeń obu podejść pomoże zaprojektować skuteczny mechanizm sugerowania wydarzeń dopasowanych do odbiorcy.

- Pozyskiwanie danych (web scraping i API) – techniki automatycznego zbierania informacji z internetu. Często dane o wydarzeniach kulturalnych dostępne są na stronach WWW lub poprzez publiczne API. Warto sprawdzić, jak używać narzędzi do scrapowania stron (np. bibliotek BeautifulSoup czy Scrapy) oraz jak korzystać z API zewnętrznych serwisów, aby pozyskać aktualne dane o wydarzeniach w Krakowie. Umiejętność wydobycia i przetworzenia takich danych zapewni, że Wasz asystent będzie miał aktualną bazę wydarzeń do proponowania.

- Chatboty i interfejsy konwersacyjne – rozważcie, jak zbudować asystenta, z którym użytkownik może porozmawiać w języku naturalnym. Przydatna będzie wiedza o rozumieniu języka (NLP) do interpretacji pytań i poleceń użytkownika oraz generowaniu odpowiedzi. Można tu wykorzystać gotowe frameworki (np. Rasa do tworzenia chatbotów opartych o ustalone intencje i dialogi) albo sięgnąć po nowoczesne modele językowe (Transformers od Hugging Face) do zbudowania konwersacyjnego AI. Taki interfejs zwiększy interaktywność systemu i ułatwi komunikację z użytkownikiem.

- Wektorowe bazy danych – przechowywanie i wyszukiwanie danych na podstawie podobieństwa wektorów. Gdy używamy embeddingów do reprezentacji opisów i preferencji, przy dużej liczbie wydarzeń potrzebne są wydajne metody znajdowania najbardziej podobnych elementów. Zapoznajcie się z narzędziami takimi jak FAISS (biblioteka Facebook AI do wyszukiwania podobnych wektorów w pamięci lokalnej) czy Pinecone (usługa chmurowa typu Vector Database). Pozwolą one szybko przeszukiwać przestrzeń wektorów w poszukiwaniu wydarzeń najbardziej zbliżonych do profilu użytkownika.

- Retrieval-Augmented Generation (RAG) – podejście łączące wyszukiwanie informacji z generatywnymi modelami AI. Technika RAG pozwala na zadawanie pytań modelowi językowemu z dołączonym kontekstem w postaci znalezionych wcześniej informacji. W kontekście asystenta kulturalnego oznacza to możliwość wykorzystania bazy wiedzy o wydarzeniach (opisy, artykuły, recenzje) do generowania odpowiedzi przez model (np. dużego modelu językowego) w stylu konwersacyjnym. Dzięki RAG, Wasz asystent mógłby np. udzielać odpowiedzi na pytania o szczegóły wydarzenia lub uzasadniać, dlaczego dane wydarzenie jest polecane, korzystając z uprzednio wyszukanych danych.

AI KRAK HACK

- Profilowanie użytkownika – sposoby reprezentowania preferencji i zainteresowań użytkownika w systemie. Zastanówcie się, jakie informacje mogą tworzyć profil kulturalny użytkownika: ulubione gatunki muzyczne, preferowane typy wydarzeń (koncerty, wystawy, spektakle), lokalizacje, przedział wiekowy, budżet itp. Profil może być budowany explicite (na podstawie ankiety, wyborów użytkownika) lub implicite (analiza historii zachowań, wyszukiwań lub kliknięć). Umiejętność skonstruowania takiego profilu i aktualizowania go w miarę interakcji z systemem będzie kluczowa dla personalizacji rekomendacji.

- Przykładowe biblioteki Pythona – warto znać lub przejrzeć narzędzia, które mogą przyspieszyć prace nad projektem: sentence-transformers (łatwe generowanie embeddingów zdań i akapitów), Hugging Face Transformers (gotowe modele NLP do wielu zadań), spaCy (wszechstronne narzędzie do NLP, m.in. tokenizacja, NER), scikit-learn (klasyczne algorytmy ML – przydatne np. do podstawowych rekomendacji czy klasteryzacji), FAISS (wydajne wyszukiwanie podobnych wektorów), Pinecone (skalowalna baza wektorowa jako usługa), Rasa (framework do budowy chatbotów z rozpoznawaniem intencji), FastAPI (szybkie tworzenie interfejsów API dla aplikacji Python), pandas (obrabianie i analiza danych, np. wczytywanie CSV z wydarzeniami) oraz BeautifulSoup/Scrapy (scraping stron internetowych z informacjami o wydarzeniach).

Powyższe zagadnienia i narzędzia nie stanowią zamkniętej listy – zachęcamy do samodzielnego poszukiwania rozwiązań i technologii.

AI KRAK HACK

Wartościowe Źródła do Budowy Inteligentnego Asystenta Kulturalnego Krakowa

Poniżej prezentujemy kompleksową listę odnośników do zasobów, które pomogą w realizacji projektu inteligentnego asystenta kulturalnego. Materiały zostały podzielone według kluczowych obszarów tematycznych, które będą istotne podczas tworzenia rozwiązania.

Przetwarzanie języka naturalnego (NLP) i reprezentacja tekstu

• Sentence Transformers - https://sbert.net - Oficjalna dokumentacja biblioteki do generowania embeddingów
• SentenceTransformers Quickstart - https://sbert.net/docs/quickstart.html - Szybkie wprowadzenie do biblioteki
• Train & Fine-Tune Sentence Transformers - https://huggingface.co/blog/how-to-train-sentence-transformers - Przewodnik od Hugging Face
• Cosine Similarity Guide - https://www.datastax.com/guides/what-is-cosine-similarity - Szczegółowy opis miary cosinusowej
• Sentence Transformers z PyTorch (YouTube) - https://www.youtube.com/watch?v=nZ5j289WN8g
• Embeddingi, podobieństwo, klasteryzacja (YouTube) - https://www.youtube.com/watch?v=OlhNZg4gOvA

AI KRAK HACK

Rozpoznawanie intencji i encji w tekście

• Intent Recognition - https://www.lyzr.ai/glossaries/intent-recognition/ - Przegląd zastosowań
• Named Entity Recognition (Wikipedia) - https://en.wikipedia.org/wiki/Named-entity_recognition
• spaCy 101 - https://spacy.io/usage/spacy-101 - Przewodnik po NER i innych funkcjach

Systemy rekomendacyjne

• Content-Based Filtering (IBM) - https://www.ibm.com/think/topics/content-based-filtering
• Collaborative Filtering (Wikipedia) - https://en.wikipedia.org/wiki/Collaborative_filtering
• User Profiles in Recommender Systems - https://milvus.io/ai-quick-reference/how-do-recommender-systems-incorporate-user-profiles
• Explicit vs Implicit Profiling (PDF) - https://www.ijcai.org/Proceedings/03/Papers/196.pdf

Pozyskiwanie danych

• BeautifulSoup Web Scraper - https://realpython.com/beautiful-soup-web-scraper-python/
• Scrapy Tutorial - https://oxylabs.io/blog/scrapy-web-scraping-tutorial

Wektorowe bazy danych

- PG Vector - https://www.enterprisedb.com/blog/what-is-pgvector
- Veaviate - https://weaviate.io/
- FAISS Tutorial (Pinecone) - https://www.pinecone.io/learn/series/faiss/faiss-tutorial/
- Pinecone – baza wektorowa - https://www.pinecone.io

AI KRAK HACK

Retrieval-Augmented Generation (RAG)

- LangChain: Budowa aplikacji RAG krok po kroku

https://python.langchain.com/docs/tutorials/rag/
Przewodnik po tworzeniu aplikacji Q&A z wykorzystaniem RAG.

- LearnByBuilding: RAG od podstaw bez bibliotek

https://learnbybuilding.ai/tutorials/rag-from-scratch
Prosty tutorial pokazujący implementację RAG bez użycia zewnętrznych bibliotek.

- DataCamp: Jak działa RAG – przegląd kroków

https://www.datacamp.com/blog/what-is-retrieval-augmented-generation-rag

Integracja i zarządzanie modelami

• MLflow + Sentence Transformers - https://mlflow.org/docs/latest/llms/sentence-transformers/tutorials/quickstart/sentence-transformers-quickstart

Dodatkowe frameworki i biblioteki

• Rasa – chatboty i intencje - https://rasa.com
• FastAPI – szybkie API - https://fastapi.tiangolo.com
• pandas – analiza danych - https://pandas.pydata.org
• LangChain – aplikacje LLM i RAG - https://python.langchain.com