Wyszukiwarka sematyczna

W oparciu o wybraną tematykę z Wikipedii

Wprowadzenie

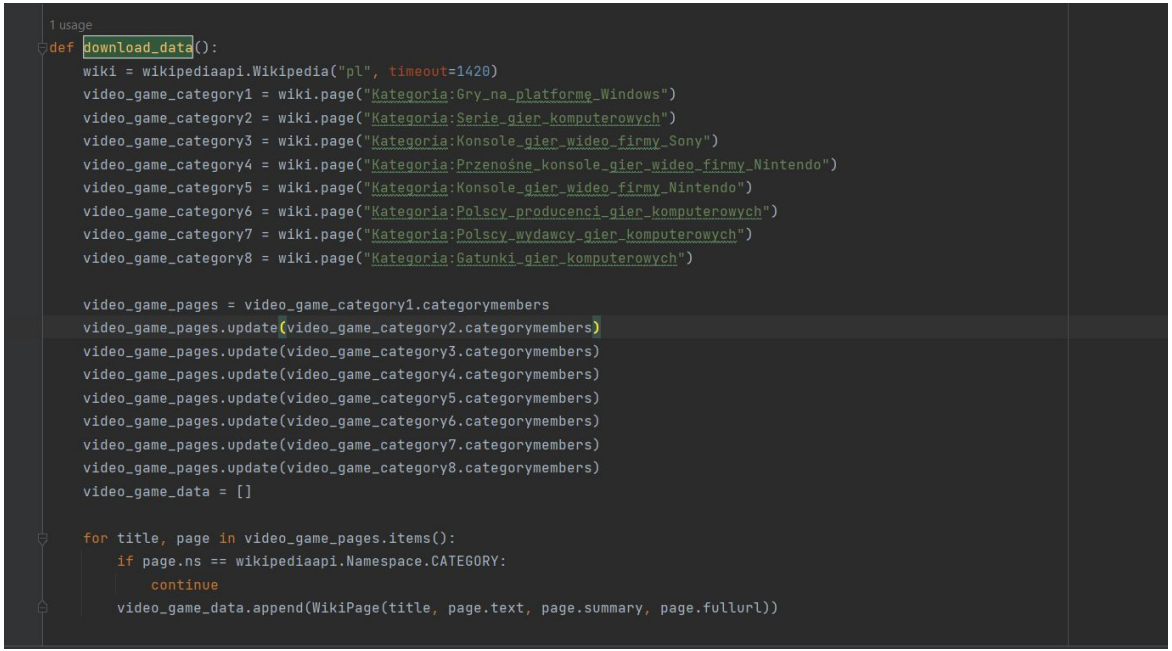
Przedstawiamy projekt wyszukiwarki semantycznej opartej o wybraną tematykę na Wikipedii. Pozwala ona na dopasowanie wyników zwrotu zapytania w oparciu o podobieństwo do zapytania

Co to jest wyszukiwanie sematyczne?

Wyszukiwanie semantyczne to technika, która pozwala na zrozumienie intencji i kontekstu zapytania, a następnie na zwrócenie najbardziej odpowiednich wyników. Zamiast polegać na dokładnym dopasowaniu słów, jak w przypadku tradycyjnych wyszukiwarek, wyszukiwanie semantyczne analizuje relację między słowami w zapytaniu.

Pobieranie danych z Wikipedii

Nasze dane pochodzą z różnych kategorii gier video z Wikipedii. Używamy biblioteki wikipediaapi do pobrania tych informacji.



```
1 usage
def download_data():
    wiki = wikipediaapi.Wikipedia("pl", timeout=1420)
    video_game_category1 = wiki.page("Kategoria:Gry_na_platformę_Windows")
    video_game_category2 = wiki.page("Kategoria:Serie_gier_komputerowych")
    video_game_category3 = wiki.page("Kategoria:Konsole_gier_wideo_firmy_Sony")
    video_game_category4 = wiki.page("Kategoria:Przenośne_konsole_gier_wideo_firmy_Nintendo")
    video_game_category5 = wiki.page("Kategoria:Konsole_gier_wideo_firmy_Nintendo")
    video_game_category6 = wiki.page("Kategoria:Polscy_producenci_gier_komputerowych")
    video_game_category7 = wiki.page("Kategoria:Polscy_wydawcy_gier_komputerowych")
    video_game_category8 = wiki.page("Kategoria:Gatunki_gier_komputerowych")

    video_game_pages = video_game_category1.categorymembers
    video_game_pages.update(video_game_category2.categorymembers)
    video_game_pages.update(video_game_category3.categorymembers)
    video_game_pages.update(video_game_category4.categorymembers)
    video_game_pages.update(video_game_category5.categorymembers)
    video_game_pages.update(video_game_category6.categorymembers)
    video_game_pages.update(video_game_category7.categorymembers)
    video_game_pages.update(video_game_category8.categorymembers)
    video_game_data = []

    for title, page in video_game_pages.items():
        if page.ns == wikipediaapi.Namespace.CATEGORY:
            continue
        video_game_data.append(WikiPage(title, page.text, page.summary, page.fullurl))
```

Cosine Similarity

Cosine Similarity to metoda, która pozwala nam na obliczenie podobieństwa między dwoma wektorami. W naszym przypadku, obliczamy podobieństwo cosinusowe między wektorem zapytania a wektorami dokumentów. W wyniku otrzymujemy wartości od 0 do 1, gdzie 1 oznacza pełne dopasowanie, a 0 brak dopasowania, w tym celu korzystamy z biblioteki cosine_similarity zawartej w sklearn

```
1 usage
def build_semantic_search_engine(query, vectorizer, X):
    query_vec = vectorizer.transform([query])
    similarities = cosine_similarity(query_vec, X)
    return similarities
```

Podobieństwo cosinusowe

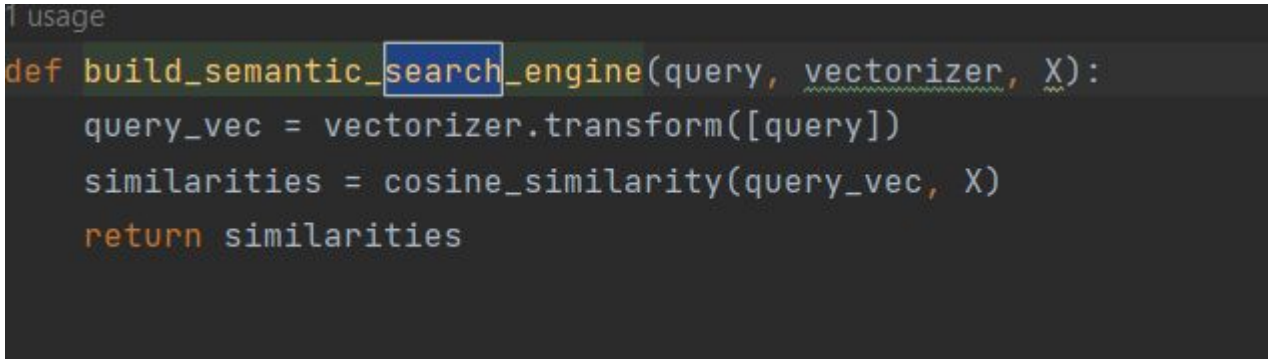
cosine_similarity to moduł pozwalający na porównanie podobieństwa zapytania i dokumentów za pomocą Cosine similarity, czyli podobieństwa kosinusowego, to miara używana do obliczenia podobieństwa między dwoma wektorami. Jest to metoda stosowana w wielu dziedzinach, w tym analizie tekstu, gdzie wektory reprezentują dokumenty lub frazy.

Podobieństwo kosinusowe mierzy kąt między dwoma wektorami. Jeśli wektory są identyczne, kąt między nimi wynosi 0 stopni, a podobieństwo kosinusowe wynosi 1. Jeśli wektory są całkowicie różne, kąt między nimi wynosi 90 stopni, a podobieństwo kosinusowe wynosi 0.

W kontekście naszego projektu, cosine_similarity z biblioteki sklearn jest używane do porównania wektora zapytania z wektorami dokumentów. Każdy dokument (w tym przypadku strona z informacjami o grze wideo) jest reprezentowany jako wektor w przestrzeni wielowymiarowej. Zapytanie jest przekształcane w tę samą przestrzeń wektorową. Następnie, podobieństwo kosinusowe jest używane do obliczenia kąta między wektorem zapytania a wektorami dokumentów. Dokumenty są następnie sortowane według ich podobieństwa do zapytania, a te o najwyższym podobieństwie kosinusowym są zwracane jako wyniki wyszukiwania.
Takie rozwiązanie pozwala na jak największe dopasowaniu wyniku zapytania z pożądanymi przez użytkownika odpowiedziami

Wyszukiwanie:

Wyszukiwanie zapytań związanych z tematyką gier wideo odbywa się poprzez przekazanie zapytania do silnika wyszukiwania, który zwraca wyniki na podstawie podobieństwa cosinusowego między wektorami zapytań a dokumentami



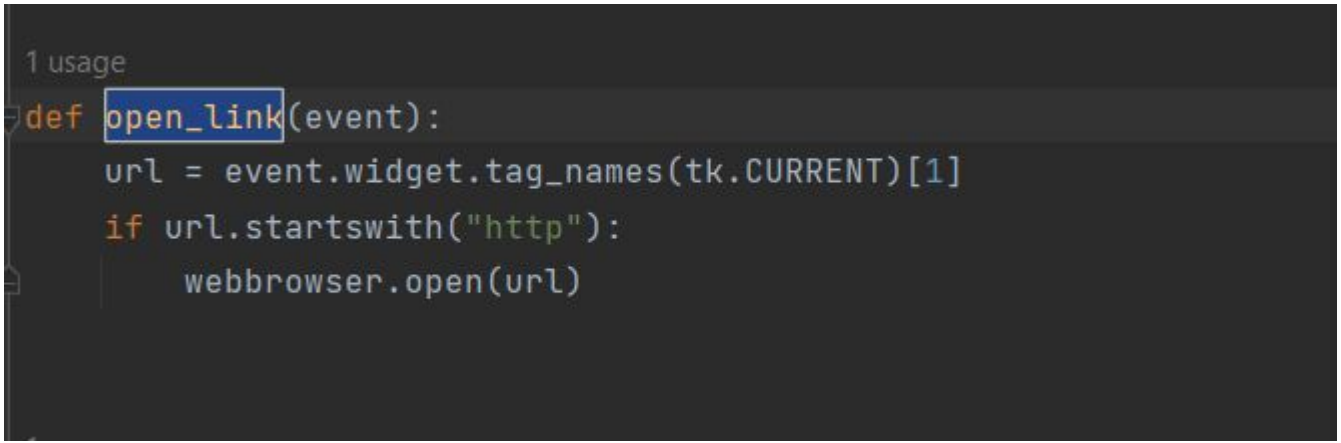
Interfejs użytkownika:

Stworzyliśmy prosty interfejs użytkownika z wykorzystaniem biblioteki tkinter, aby umożliwić użytkownikom wprowadzenie zapytania i wyświetlanie wyników wyszukiwania

```
root = tk.Tk()
root.title("Wyszukiwarka semantyczna - Gry wideo")

search_entry = tk.Entry(root, width=50)
search_entry.grid(row=0, column=0, padx=5, pady=5)

search_button = tk.Button(root, text="Wyszukaj", command=search_btn)
search_button.grid(row=0, column=1, padx=5, pady=5)

download_button = tk.Button(root, text="Pobierz dane", command=download)
download_button.grid(row=0, column=2, padx=5, pady=5)

progressbar = ttk.Progressbar(root, mode="indeterminate")
progressbar.grid(row=1, column=0, columnspan=3, padx=5, pady=5)
progressbar.stop()

link_list = tk.Text(root, state='disabled')
link_list.grid(row=2, column=0, columnspan=3, padx=5, pady=5, sticky=tk.NSEW)

root.mainloop()
```

Obsługa zdarzeń:

Podczas wyszukiwania, użytkownik może kliknąć na wynik, aby otworzyć stronę Wikipedii związaną z danym zapytaniem



Pobieranie danych:

Z racji na długi proces pobierania danych usprawniliśmy nasz projekt o możliwość pobrania danych lokalnie dzięki czemu użytkownik może również zaktualizować dane używane przez silnik wyszukiwania, klikając przycisk 'Pobierz dane’

```
download_button = tk.Button(root, text="Pobierz dane", command=download)
download_button.grid(row=0, column=2, padx=5, pady=5)
```

Demo:

teraz pokażemy kilka slajdów z funkcjonowania naszego programu:

po uruchomieniu programu pojawia nam się,okienko wygenerowane przez tinker zawierające:

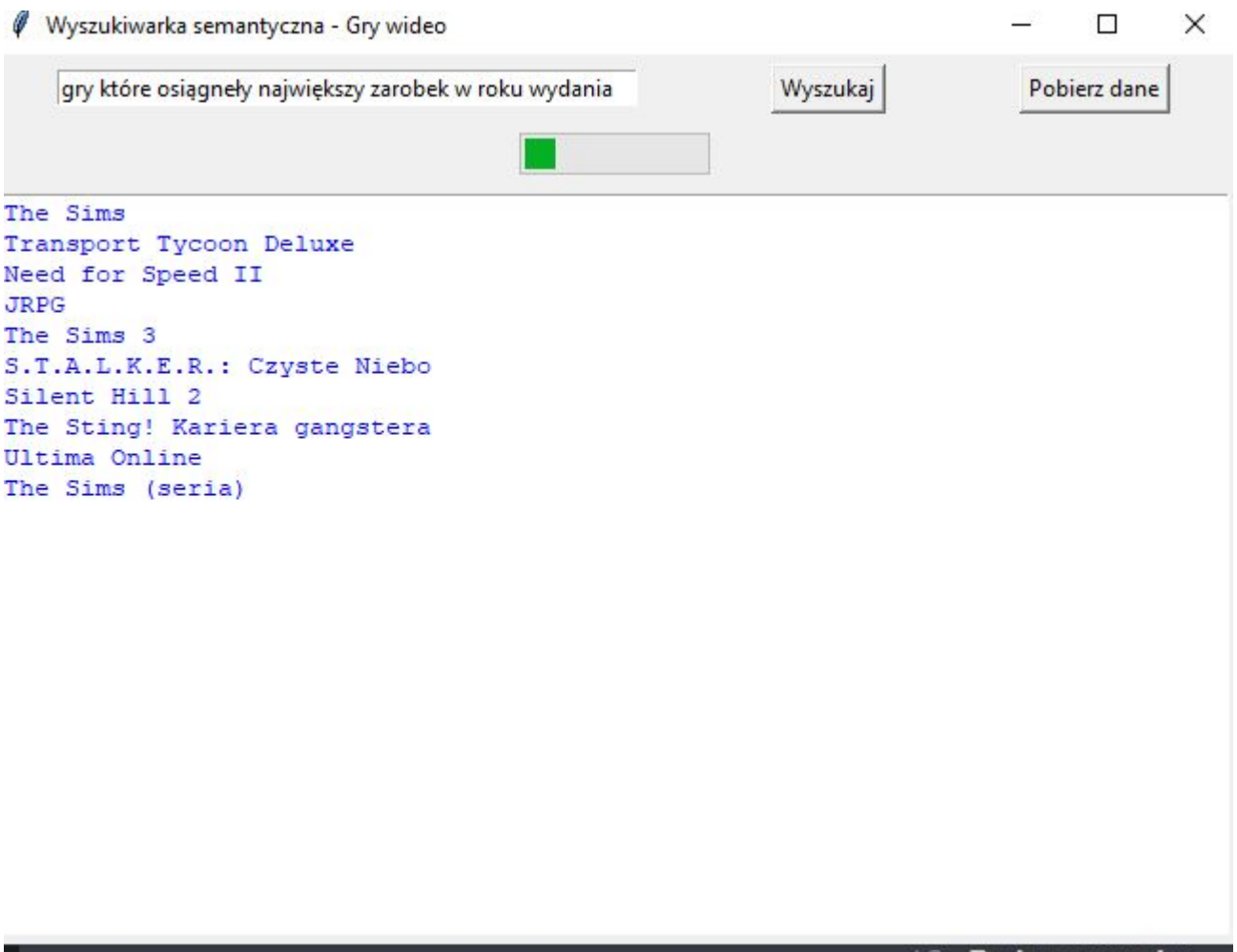
-pole do wpisania tekstu

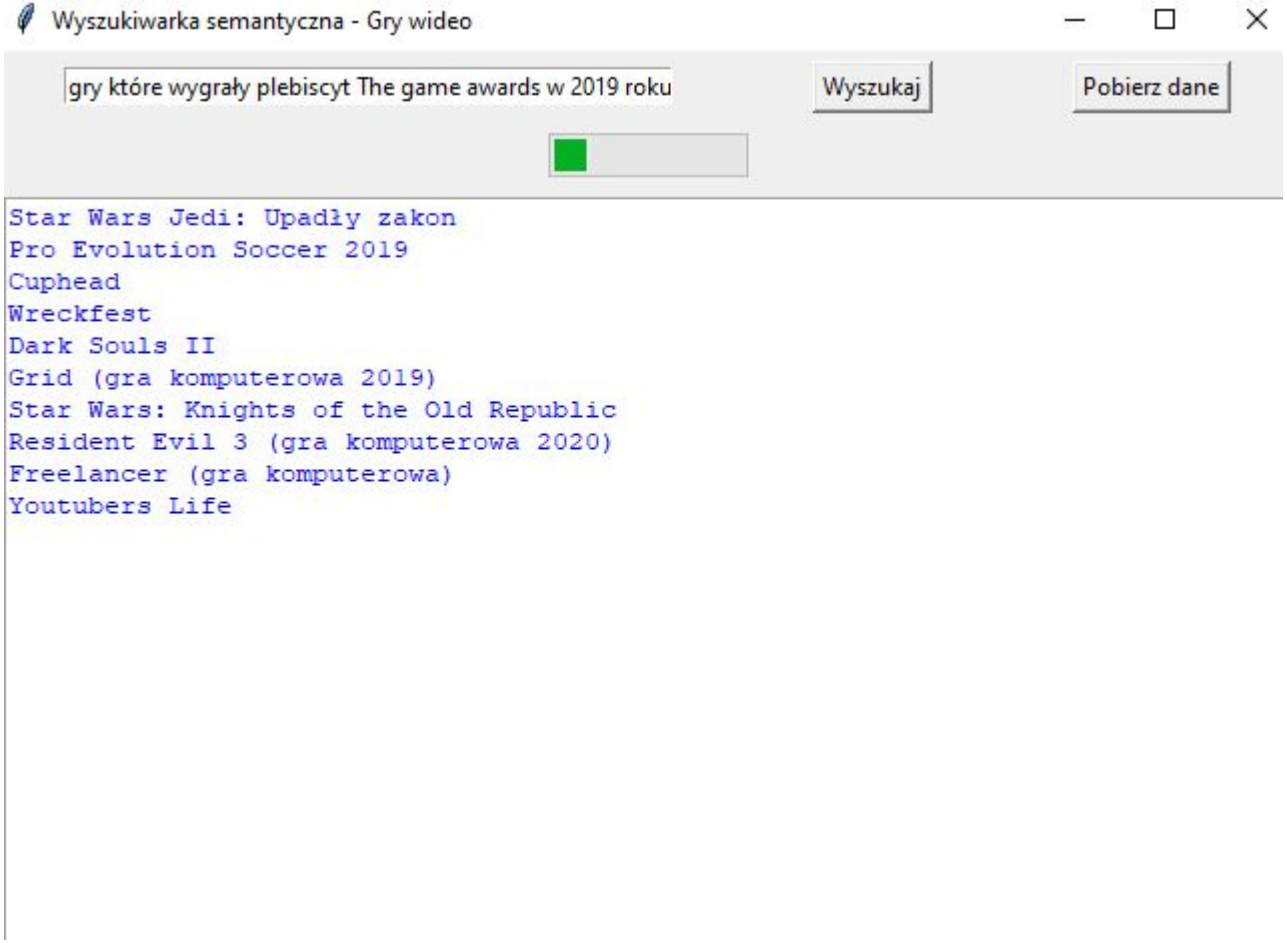
-przycisk wyszukaj

-przycisk pobierz dane który aktualizuje lokalne dane

-pasek postępu pobierania

-pole wyników


Wyszukiwarka semantyczna - Gry wideo
gry które osiągnęły największy zarobek w roku wydania
Wyszukaj
Pobierz dane
The Sims
Transport Tycoon Deluxe
Need for Speed II
JRPG
The Sims 3
S.T.A.L.K.E.R.: Czyste Niebo
Silent Hill 2
The Sting! Kariera gangstera
Ultima Online
The Sims (seria)


Wyszukiwarka semantyczna - Gry wideo
gry które wygrały plebiscyt The game awards w 2019 roku
Wyszukaj
Pobierz dane
Star Wars Jedi: Upadły zakon
Pro Evolution Soccer 2019
Cuphead
Wreckfest
Dark Souls II
Grid (gra komputerowa 2019)
Star Wars: Knights of the Old Republic
Resident Evil 3 (gra komputerowa 2020)
Freelancer (gra komputerowa)
Youtubers Life

Podsumowanie:

Nasz program pozwala na wyszukiwanie semantyczne wybranego tematu (w naszym przypadku gier wideo) na podstawie informacji z Wikipedii. Używamy wikipediaapi uzyskujemy dostęp do danych z wikipedii które są przetwarzane w format JSON,a następnie obliczamy podobieństwo cosinusowe między zapytaniem, a dokumentami, aby uzyskać wyniki wyszukiwania. Interfejs użytkownika umożliwia łatwą interakcję.

Dziękujemy za uwagę

Kamil Bajorek

Marcin Kalandyk