Narzędzie do tworzenia podsumowań tekstu

Wstęp

Problem, który staraliśmy się rozwiązać, polegał na automatycznym tworzeniu podsumowań długich tekstów. Podsumowania mają na celu przedstawienie głównych punktów omawianego tematu, zredukowanie tekstu do najważniejszych idei i ułatwienie czytelnikowi zrozumienia ogólnego przesłania tekstu.

Rozwinięcie

W celu rozwiązania problemu, zdecydowaliśmy się na implementację narzędzia do tworzenia podsumowań opartego na analizie częstotliwości słów, podobieństwie zdań i algorytmie PageRank.

Pierwszym krokiem było tokenizowanie tekstu na zdania i słowa oraz usunięcie tzw. stop words (słów funkcyjnych), które nie niosą znaczącej treści semantycznej, takich jak ‘the’, ‘is’, ‘and’, ‘z’, ‘i’ itp.

Następnie, wykorzystaliśmy wektoryzator TF-IDF do przekształcenia zdań w wektory. TF-IDF, co oznacza "term frequency-inverse document frequency", to statystyczna metoda do oceny ważności słowa w dokumencie, oparta na częstości występowania słowa i odwrotności częstości dokumentu.

Następnie obliczyliśmy podobieństwo zdań na podstawie tych wektorów, używając cosinusowej miary podobieństwa.

Miara podobieństwa cosinusowego, często nazywana podobieństwem cosinusowym, to miara podobieństwa między dwoma wektorami w przestrzeni wielowymiarowej. Podobieństwo cosinusowe jest miarą kąta między dwoma wektorami, a nie ich bezwzględnej wielkości. W kontekście analizy tekstu i NLP (Natural Language Processing), podobieństwo cosinusowe jest często używane do porównywania dokumentów lub zdań. Każdy dokument lub zdanie jest reprezentowane jako wektor, w którym wymiary odpowiadają unikalnym słowom, a wartości w tych wymiarach odpowiadają liczbie wystąpień tych słów (lub innemu rodzajowi ważenia, takiemu jak TF-IDF). Następnie podobieństwo cosinusowe jest używane do określenia, jak podobne są do siebie dwa dokumenty lub zdania.

Na podstawie macierzy podobieństwa stworzyliśmy graf zdań, a następnie zastosowaliśmy algorytm PageRank do rangowania zdań. Algorytm PageRank, który został stworzony przez założycieli Google, ocenia ważność węzłów w sieci na podstawie struktury linków.

PageRank został wybrany jako miara ważności zdań w grafie, ponieważ jest dobrze sprawdzonym algorytmem wykorzystywanym w wielu dziedzinach, takich jak analiza sieci społecznościowych czy ranking stron internetowych.

TfidfVectorizer został wybrany, ponieważ jest łatwy w użyciu i pozwala na uwzględnienie zarówno częstotliwości występowania słów (TF) w zdaniach, jak i ich unikalności w całym tekście (IDF). Jest to istotne, ponieważ często ważniejsze są zdania zawierające unikalne informacje.

Wybrano podobieństwo cosinusowe, ponieważ jest to popularna miara podobieństwa, która jest bardziej odporna na różnice w długości zdań niż inne metody. Dzięki temu, można ocenić podobieństwo zdań niezależnie od ich długości.

Kod źródłowy programu

```python
import networkx as nx
import nltk
from nltk.tokenize import sent_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

nltk.download('punkt')
nltk.download('stopwords')


def load_stopwords(file_path):
    with open(file_path, 'r', encoding='UTF-8') as file:
        stopwords = [line.strip() for line in file]
    return stopwords


def summarize_text(file_path, num_sentences):
    with open(file_path, 'r', encoding='UTF-8') as file:
        text = file.read()

    # stop_words = list(set(stopwords.words('polish')))
    stopwords_path = 'polish.stopwords.txt'
    stop_words = load_stopwords(stopwords_path)
    sentences = sent_tokenize(text)

    # Obliczenie podobieństwa zdań za pomocą wektorów TF-IDF i cosinusowej miary podobieństwa
    vectorizer = TfidfVectorizer(stop_words=stop_words)
    sentence_vectors = vectorizer.fit_transform(sentences)
    similarity_matrix = cosine_similarity(sentence_vectors)

    # Tworzenie grafu zdań z macierzą podobieństwa
    sentence_graph = nx.from_numpy_array(similarity_matrix)

    # Obliczenie wartości PageRank dla zdań
    sentence_scores = nx.pagerank(sentence_graph)

    # Sortowanie zdań według ich wartości PageRank
    sorted_sentences = sorted(sentence_scores, key=sentence_scores.get, reverse=True)

    # Wybór zdań o najwyższych wartościach PageRank jako podsumowania
    summary_sentences = [sentences[i] for i in sorted_sentences[:num_sentences]]

    summary = ' '.join(summary_sentences)
    return summary, sentence_scores


if __name__ == "__main__":
    file_path = "texts/bereszynski-polish.txt"
    num_sentences = 5
    summary, scores = summarize_text(file_path, num_sentences)
    print(summary)

```

Funkcjonowanie programu

Dla artykułu w języku angielskim:

Wynik dla num_sentences = 2

```
Two million cubic metres of rock is coming loose from the mountain above, and a rockslide could obliterate the village. Days of
heavy rain could bring two million cubic metres of loosened rock crashing down the mountainside onto the village, scientists wa
rned.
,
```

Wynik dla num_sentences = 7

```
Two million cubic metres of rock is coming loose from the mountain above, and a rockslide could obliterate the village. Days of
heavy rain could bring two million cubic metres of loosened rock crashing down the mountainside onto the village, scientists wa
rned. As the minutes ticked towards the deadline to leave, even Brienz's dairy cows were being taken to safety. Brienz's fewer
than 100 villagers were given just 48 hours to pack what they could and abandon their homes. Residents of a tiny Swiss village
have all been evacuated because of the risk of an imminent rockslide. The residents, some young, some old, families, farmers an
d professional couples, had two days to abandon their homes. Even the dairy cows were loaded up for departure after geologists
warned a rockfall was imminent.
```

Dla artykułu w języku polskim:

Wynik dla num_sentences = 3

```
Pewne jest, że Napoli nie wykupi Polaka po zakończeniu sezonu i ten wróci do Sampdorii. Bartosz Bereszyński jest wypożyczony do
Napoli do końca sezonu z opcją wykupu za 1,8 mln euro. Bartosz Bereszyński wróci po zakończeniu sezonu do Sampdorii, która jest
już pewna spadku do Serie B. Kontrakt Polaka jest ważny do końca czerwca 2025 roku.
```

Wynik dla num_sentences = 5

```
Pewne jest, że Napoli nie wykupi Polaka po zakończeniu sezonu i ten wróci do Sampdorii. Bartosz Bereszyński jest wypożyczony do
Napoli do końca sezonu z opcją wykupu za 1,8 mln euro. Bartosz Bereszyński wróci po zakończeniu sezonu do Sampdorii, która jest
już pewna spadku do Serie B. Kontrakt Polaka jest ważny do końca czerwca 2025 roku. Bartosz Bereszyński trafił do Napoli w styc
zniu tego roku na zasadzie wypożyczenia z Sampdorii. Bartosz Bereszyński zagrał do tej pory zaledwie jeden mecz w barwach Napol
i, konkretniej w 1/8 finału Pucharu Włoch, gdzie Cremonese wyeliminowało faworyta po rzutach karnych.
```

Podsumowanie

Nasze narzędzie do tworzenia podsumowań tekstu działa skutecznie, generując zwięzłe i sensowne podsumowania z długich tekstów. Algorytm PageRank, w połączeniu z analizą częstotliwości słów i podobieństwa zdań, pozwala na identyfikację najważniejszych zdań, które najlepiej reprezentują główne idee w tekście.

Bibliografia

Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.
"Text summarization with NLTK in Python." Stack Abuse. https://stackabuse.com/text-summarization-with-nltk-in-python/
"PageRank." Wikipedia. https://en.wikipedia.org/wiki/PageRank
"tf–idf." Wikipedia. https://en.wikipedia.org/wiki/Tf%E2%80%93idf
"Cosine similarity." Wikipedia. https://en.wikipedia.org/wiki/Cosine_similarity