Documentație Proiect Watson

Bucsa Mariana-Domnica

Ciocoiu Adela-Nicoleta

Cirstean Alexandra-Elena

Ianos Raluca-Ioana

Descrierea aplicatiei

Proiectul se concentrează pe crearea un sistem care poate răspunde la întrebări, similar cu Watson de la IBM. Scopul aplicației este să identifice paginile de pe Wikipedia care sunt cele mai probabile răspunsuri la întrebările de tip Jeopardy. Acest proiect implică în principal clasificarea și căutarea pe paginile Wikipedia furnizate pentru a găsi răspunsurile potrivite la întrebările date.

Structura proiectului

1. Biblioteci:
   - Folosește nltk pentru tokenizare, stemmare și eliminarea stop words - urilor
   - Biblioteca Whoosh este utilizată pentru indexare și căutare.
2. **Clasa Index**:
   - Această clasă este responsabilă pentru crearea și gestionarea indexului.
   - Dispune de metode pentru crearea indexului, răspunsul la întrebări pe baza unor indicii și întrebări furnizate și afișarea titlurilor care se potrivesc interogării.
   - Metoda `create()` creează indexul prin iterarea fișierelor din directorul de intrare, extragerea titlurilor și a conținutului și adăugarea acestora în index.

- Metoda `answer_question()` caută pagina cea mai potrivită ca si răspuns la întrebare pe baza indiciului furnizat.
- Metoda `get_top_titles()` recuperează top 10 titluri care se potrivesc interogării.

3. **Clasa TextProcessor**:
   - Procesează conținutul și răspunsurile pentru indexare și căutare.
4. Clasa ChatGpt:
   - Clasa ChatGpt gestionează interacțiunea cu serviciul OpenAI pentru a obține răspunsuri mai bune la întrebările utilizatorilor.
   - Metoda `answer_question(clue, question)`: Folosește indexul existent pentru a obține cele mai relevante titluri de pagini Wikipedia care ar putea conține răspunsul la întrebare. Apoi, face o cerere către serviciul OpenAI pentru a obține un răspuns mai precis.
   - Metoda `get_top_titles(clue, question, nr_pages)`: Folosește indexul existent pentru a obține cele mai relevante titluri de pagini Wikipedia care ar putea conține răspunsul la întrebare. Apoi, face o cerere către serviciul OpenAI pentru a obține un răspuns mai precis. Afișează titlurile relevante într-o alta ordine generată prin intermediul Api-ului.

6. Clasa MeasurePerformance:

- Această clasă este responsabilă pentru măsurarea performanței sistemului nostru de căutare și răspuns la întrebări.
- Metoda `print_results()` este utilizată pentru a afișa rezultatele performanței, inclusiv precizia la primul răspuns (P@1) și rata reciprocă medie (MRR) pentru ambele abordări - folosind indexul și ChatGpt.
- Metodele `__get_results_index()` și `__get_results_chat()` sunt utilizate pentru calcularea preciziei și MRR pentru index și, respectiv, ChatGpt.
- În cadrul acestor metode, fiecare întrebare din setul de date este procesată, iar apoi se calculează precizia și MRR-ul pe baza răspunsurilor obținute.

Aplicatia ne prezintă un meniu cu mai multe opțiuni:

Opțiunea 1

- Introducem 1
- Nu avem output, doar se va genera indexul

Opțiunea 2

- Introducem 2
- Introducem indiciul și întrebarea
- Este afișat titlul uneia dintre pagini ca răspuns la întrebare, folosind indexul creat anterior

Opțiunea 3

- Introducem 3
- Introducem indiciul și întrebarea
- Este afișat titlul uneia dintre pagini ca răspuns la întrebare, folosind API-ul CHATGPT

Opțiunea 4

- Introducem 4
- Introducem indiciul și întrebarea
- Sunt afișate top 10 titluri de pagini Wikipedia ca răspuns la întrebare, folosind indexul creat anterior

Opțiunea 5

- Introducem 5
- Introducem indiciul și întrebarea

- Sunt afișate top 10 titluri de pagini Wikipedia ca răspuns la întrebare, folosind API-ul CHATGPT

Opțiunea 6

- Introducem 6
- Se afișează precizia si MRR pentru întrebările din fișierul “questions.txt”

Opțiunea 7:

- Introducem 6
- Ieșim din aplicație

Pregătirea indexarii

Pentru indexare s-a folosit bibliotecile nltk si whoosh, mai specific Schema:

```
self.schema = Schema(
            title=TEXT(stored=True),
            content=TEXT(stored=True),
                )
```

Indexarea se realizează folosind un set de date. Se parcurge fiecare fișier din setul de date, se extrage atât titlul, cât și conținutul fiecărei pagini Wikipedia (luând în considerare că fiecare titlu are formatul [[Title]]), și, ulterior, aceste informații sunt supuse proceselor de tokenizare, eliminare a cuvintelor de oprire (specific pentru limba engleză) și stemming.

1. **Tokenizare:** Separă textul în unități mai mici, facilitând manipularea și analiza textului.
2. **Eliminarea cuvintelor de oprire:** Excluderea cuvintelor nesemnificative pentru o analiza mai ușoară a textului
3. **Stemming:** Reduce cuvintele la forma lor de bază

Probleme specifice conținutului Wikipedia

Extragerea titlului si a conținutului s-a făcut tinand cont de structura fișierelor.

Separarea titlului de conținutul acestuia s-a făcut folosit expresii regulare.

După procesul de tokenizare, s-a observat un număr mare de semne de punctuație si nu numai, ce nu ar trebui luate in considerare pentru crearea indexului, astfel partea de filtrare implica si eliminarea acestora.

Măsurarea Performantei

Performanța sistemului Jeopardy a fost măsurată folosind un set de 100 de intrebari.

Întrebările sunt extrase din fisierul questions.txt, care are urmatoarea structura:

```
question

clue

answer

newline
```

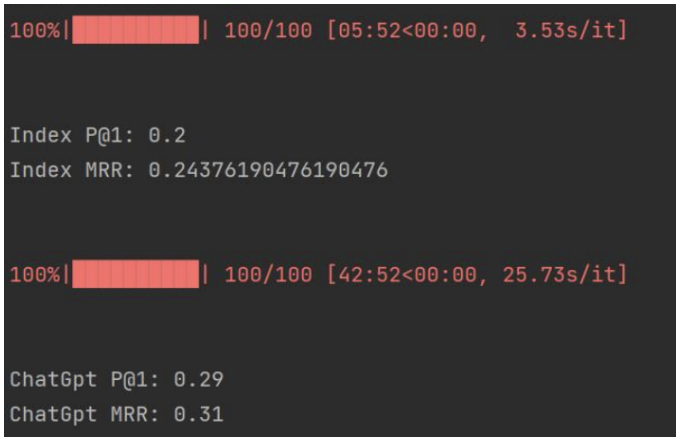
După ce întrebările si indiciile sunt trecute prin procesul de filtrare, acestea sunt trimise pe rând indexului si apoi metodelor specifice ChatGPT, așteptând un răspuns.

Răspunsul este analizat din 2 punte de vedere, folosind 2 metrici diferite P@1 si MRR.

Prima metrica implica măsurarea performantei sistemului, evaluând dacă răspunsul corect este clasat primul în rezultatele căutării.

Cea de a doua metrica evaluează modului în care sistemul clasifică răspunsurile corecte, chiar dacă răspunsul corect nu este primul în listă.

Am ales aceste două metode pentru a oferi o evaluare echilibrată a sistemului, având în vedere diversitatea modurilor în care răspunsurile corecte pot fi prezentate în rezultatele căutării.



Rezultatele indică performanța sistemului în două contexte distincte: indexarea și metodele ChatGPT.

- **Index P@1:** Pentru indexare, procentajul P@1 este de 0.2, sugerând că doar 20% dintre întrebările au răspunsul corect clasat primul în rezultatele indexului.
- **Index MRR:** Pentru indexare, MRR este 0.2437.. indicând că, în medie, răspunsurile corecte ocupă poziții relativ mai înalte în lista de rezultate.
- **ChatGPT P@1:** Pentru metodele ChatGPT, procentajul P@1 este de 0.29, suggerând că aproximativ 29% dintre întrebări au răspunsul corect clasat primul în rezultatele ChatGPT. Se observa o imbunatatie a metodei indexului.

- **ChatGPT MRR:** Pentru metodele ChatGPT, MRR este 0.31, indicând că, în medie, răspunsurile corecte ocupă poziții relativ mai înalte în lista de rezultate a ChatGPT.

Aceste rezultate indică faptul că metoda index + ChatGPT are o performanță mai bună decât simpla indexare. Integrarea tehnologiei de procesare a limbajului natural poate îmbunătăți eficiența sistemului în identificarea răspunsurilor corecte și clasarea acestora în mod adecvat în rezultatele căutării.

Analiza erorilor

La câte întrebări s-a răspuns corect/încorect?

- Sistemul a răspuns corect la 20 întrebări.
- Sistemul a răspuns incorect la 80 de întrebări.

De ce credem că întrebările corecte pot fi răspunse de către un sistem atât de simplu?

Sistemul funcționează in scenarii in care indiciile si întrebările se potrivesc exact cu conținutul din paginile Wikipedia.

Oferirea de indicii cât mai puțin ambigue cu o probabilitate mare de a fi reprezentate direct in pagina de Wikipedia relevanta este asociata cu obținerea unui răspuns corect.

Ce probleme am observat în cazul întrebărilor la care sistemul răspunde greșit?

Sistemul oferă raspusuri incorecte atunci când indiciile oferite sunt ambigue. Acesta se bazează doar pe tokanizarea termenilor, neținând cont de sinonime.

De asemenea, faptulul ca sistemul nu interpreteaza contextul întrebărilor ar mai putea fi o cauza a răspunsurilor incorecte.