IA368DD - Deep Learning aplicado a sistemas de buscas
Processo Seletivo (Alunos Especiais)

Data de entrega: 22/02/2023
Nome: Leonardo Bernardi de Avila
RA: l224016

Projeto:
Construindo um Sistema Simples de Recuperação de Informação usando BM25 e GPT-3 e avaliado na coleção CISI.

Planejamento:

Para utilizar o chatGPT (GPT3) como ferramenta auxiliar no projeto, pensei em 2 abordagens iniciais:

1) Testá-lo como ferramenta de consulta:

Pedir ao chatGPT para explicar o que é o algoritmo BM25, como se dá sua aplicação em Information Retrieval e indicar métodos que possam ser utilizados em sua avaliação.

2) Testá-lo como ferramenta para auxiliar na construção de código:

Gerar um código python que implemente uma classe do BM25 e aplique essa classe ao collection CISI.

Irei testar o chatGPT nas abordagens acima. Posteriormente, irei construir o IR System baseado no BM25 e avaliá-lo:

3) Implementação do Information Retrieval System no collection CISI.
4) Avaliação do Information Retrieval System.
5) Considerações Finais.
6) Ferramentas Utilizadas.
7) Material de apoio.

1) Testá-lo como ferramenta de consulta:

Foram testados alguns prompts como “what is BM25”, “how is IDF(qi) computed”, “metrics to evaluate information retrieval systems like BM25”. Por conter textos longos, a saída foi disponibilizada em sua totalidade no jupyter notebook.

Comentários dessa etapa:

As explicações dadas pelo chatGPT parecem estar condizentes em um primeiro momento, porém, são fornecidas tantas informações que não sabemos se a ferramenta está cometendo algum erro ou alucinação. Ao mesmo tempo, o usuário pode ter um grande trabalho ao tentar validá-las. Dessa forma, entendo que pode ser utilizado como uma ferramenta auxiliar, mas confiar cegamente em todos os seus outputs pode não ser adequado. Aqui, foi possível ter como saída a formulação do BM25, alguns fatos históricos e métricas de avaliação de Sistemas de Recuperação de Informação.

2) Testá-lo como ferramenta para auxiliar na construção de código:

Foi testado o prompt “how to code a BM25 class in python and apply it to CISI collection.” O ChatGPT foi capaz de estruturar: a implementação sugerida da classe em python, como seria o código em python para leitura do corpus do arquivo “cisi.all” do conjunto de dados e, para finalizar, sugeriu um código para utilizar a classe apresentada para ranquear os documentos do corpus dada uma query. Novamente, as saídas do ChatGPT foram apresentadas no jupyter notebook.

Comentários dessa etapa:

Não cheguei a validar 100% os códigos sugeridos pelo ChatGPT, mas sabemos que ele pode cometer erros então é essencial fazer essa validação em uma aplicação real. Apesar disso, foi possível observar que o ChatGPT foi capaz de dar alguns direcionamentos importantes na construção da classe como o método “initialize” (que seria o fitting da classe no corpus de interesse) e o método “score” (que seria o método de maior interesse onde, dada uma query, calcularíamos o score de cada documento). Na parte de aplicação no collection CISI, o ChatGPT trouxe elementos específico do dataset como a identificação de filtros necessários no conjunto de dados "line.startswith('.I')" e "line.startswith('.W')".

Devido ao tempo do projeto, optei por não validar todas as linhas de código do ChatGPT e sim partir de uma classe do BM25 que eu já havia utilizado anteriormente, mas foi possível observar que teríamos um bom direcionamento do código para concluir o projeto com a saída do ChatGPT.

3) Implementação do Information Retrieval System no collection CISI.

Optei por não utilizar a saída do ChatGPT para servir de base para construção da classe do BM25 e sim uma abordagem que já havia utilizado anteriormente e está disponível em [1]. A classe foi modificada para internamente remover stopwords e possuir opção de pré-processamento.

Dessa forma, a classe BM25 é inicializada com os parâmetros de controle k1 e b que servem para controlar no cálculo final de score os efeitos de "em quantos documentos o termo aparece" e o "tamanho do documento".

Além disso, a classe possui os métodos preprocessing, fit, search e _score:

- preprocessing: aplica o pré-processamento nos textos, deixando apenas caracteres alfanuméricos.
- fit: calcula os parâmetros necessários para cálculo do score do BM25 a partir de um corpus.
- search: faz na query o mesmo pré-processamento que foi aplicado no corpus e chama o método _score.
- _score: método que, para cada termo da query, calcula um score relativo a cada documento do corpus. Ao final, os scores são somados para compor o score de cada documento em relação à query como um todo.

Para aprofundar o entendimento do BM25 e métricas de avaliação, foi utilizado [2].

Para entendimento sobre o conjunto de dados e de como tratá-lo foram utilizadas informações e códigos disponíveis em: [3], [4], [5], [6].

Os códigos utilizados para implementação da classe do BM25, download e tratamento do dataset estão disponíveis em jupyter notebook.

O output do Sistema de Recuperação de Informação será um arquivo run.cisi.bm25.txt no formato do trec eval para que possamos avaliar métricas de desempenho com uma aplicação específica do trec eval, disponível no pygaggle.

Foi escolhido um k1 = 0.9 e b = 0.4 conforme utilizado algumas vezes em [2]. Esses valores podem ser otimizados para que possamos obter um desempenho final melhor, não existe configuração ideal para todos os datasets, os valores "ideais" de k1 e b irão variar de acordo com a estrutura dos documentos e queries de cada conjunto de dados.

Nessa etapa, apenas as 76 queries que possuem mapeamento de relevância de documentos no arquivo CISI.REL foram utilizadas para que, ao final, possamos aplicar uma métrica de desempenho, mas o ranqueamento pode ser realizado no corpus todo para todas as 112 queries.

Para que o sistema pudesse ranquear quase que todos os documentos do corpus, foi escolhido um n=1000 dos 1460 documentos possíveis, mas esse valor pode variar de acordo com a necessidade da aplicação. Esse valor parametriza os top-n documentos que o sistema de recuperação de informação irá retornar, dada uma query de interesse e um corpus com vários documentos.

O arquivo run.cisi.bm25.txt gerado foi escrito em um formato para que ele possa ser utilizado pela aplicação trec eval na avaliação de desempenho do sistema.

Ele possui as seguintes "colunas":

- QUERY_ID
- Q0 (não é utilizado)
- DOC_ID
- RANK do documento com relação à query
- SCORE calculado para o documento com relação à query
- STANDARD (BM25).

4) Avaliação:

Para avaliação do Sistema de Recuperação de Informação proposto foi utilizado o formato TREC_EVAL em conjunto com o pacote pygaggle, que disponibiliza a aplicação de avaliação.

A métrica de avaliação escolhida foi o precision e ele será cálculo para o top 1, 3 e 5. A métrica visa encontrar a proporção de documentos relevantes dentro dos top-1, 3, 5 documentos retornados pelo sistema para cada query.

A aplicação nos permite calcular a métrica para cada query ou para o sistema como um todo. Os resultados gerais para o sistema são apresentados na Tabela 1 a seguir, a visão por query pode ser encontrada no jupyter notebook.

Tabela 1 - Desempenho do Sistema Proposto

Métrica	Modo	Valor
P_1	all	0,3947
P_3	all	0,3728
P_5	all	0,3342

5) Comentários Finais:

1. ChatGPT:

- **PRÓS:** pode ser utilizado como ferramenta auxiliar de estudo, busca de informação e formulação de código.
- **CONS:** não temos meios rápidos de validar se a ferramenta está cometendo algum erro ou alucinação.

2. BM25:

- **PRÓS:** algoritmo de fácil entendimento e implementação e que serve como um bom baseline para sistemas de busca. Além disso, o "treinamento" é rápido e a latência de busca é baixa.
- **CONS:** algoritmo baseado em "exact match", sendo assim, palavras como "prefeito" e "prefeita" seriam entendidas pelo algoritmo como sendo completamente diferentes. Aqui, uma possível abordagem seria de se melhorar o processamento com tratamentos de stemming ou lemmatization.

6) Ferramentas Utilizadas:

- https://chat.openai.com/chat4

7) Material de apoio:

Implementação do BM25:

[1] http://ethen8181.github.io/machine-learning/search/bm25_intro.html

Material de estudo:

[2] https://arxiv.org/pdf/2010.06467.pdf

Material sobre o dataset CISI:

[3] http://ir.dcs.gla.ac.uk/resources/test_collections/cisi/

[4] https://www.kaggle.com/datasets/dmaso01dsta/cisi-a-dataset-for-information-retrieval

[5] https://www.kaggle.com/code/dmaso01dsta/cisi-eda/notebook

[6] https://www.kaggle.com/code/razamh/some-changes-in-cisi-eda

Avaliação do sistema de recuperação:

[7] http://www.rafaelglater.com/en/post/learn-how-to-use-trec_eval-to-evaluate-your-information-retrieval-system

[8] https://github.com/castorini/pygaggle