Trabalho Prático 1: Recuperação da Informação

Anna Paula Figueiredo Gonçalves[1]

[1]Departamento de Computação e Sistemas – Universidade Federal de Lavras
Caixa Postal 3.037 – 37.200-900 – Lavras – MG – Brazil

anna.goncalves2@estudante.ufla.br

1. Introdução

A recuperação de informação é um campo computacional que abrange a busca, identificação e acesso a dados [Salton 1971]. Esses dados podem se apresentar em variados formatos, incluindo texto, imagens, áudios e vídeos. O objetivo é que os sistemas de informação possam eficientemente buscar e acessar esses conteúdos em tempo hábil. Este domínio é de relevância notável em áreas como sistemas de busca na *web* bibliotecas digitais e bancos de dados.

Como um campo de pesquisa ativo, é objeto de investigação para o aprimoramento de técnicas e tecnologias visando otimizar o desempenho na recuperação de informação. Na era do *big data*, caracterizada pelo incessante acúmulo de dados em diversos formatos e de grande magnitude, é imperativo que a informação seja recuperada com máxima celeridade [Witten et al. 2016]. Assim, surge a necessidade de adaptar os sistemas de recuperação de informação, incorporando modelos de linguagem, refinando os métodos de armazenamento, bem como aprimorando técnicas de indexação e ferramentas de busca.

2. Descrição do Trabalho

O trabalho visa explorar a coleção de referência CFC (*Cystic Fibrosis*), disponibilizada pelo professor da disciplina, Denilson Alves Pereira[1]. O objetivo é desenvolver uma máquina de busca aplicando técnicas de vetorização e indexação para recuperar esses documentos. As técnicas de vetorização utilizadas são o TF-IDF (*Term Frequency — Inverse Data Frequency*) e *Embeddings*. Para a busca, adotou-se a biblioteca FAISS (*Facebook AI Similarity Search*), especializada em busca por similaridade e indexação eficiente de documentos [Facebook 2023].

A similaridade é mensurada pela distância de cosseno, para avaliar a proximidade das consultas através do *IndexFlatIP* do FAISS. Para avaliar o desempenho das buscas, foram calculadas métricas como precisão, revocação, precisão-R e tempo de execução das técnicas utilizadas. Além disso, foram computadas as métricas P@5 e P@10 médias para as consultas, com o MRR(Q) (*Mean Reciprocal Rank at rank Q*) com o *threshold* Sh=5, e o histograma da precisão-R para as 20 primeiras consultas.

2.1. Base de dados

A base de dados é constituída por dois conjuntos distintos: um composto por documentos e outro por consultas. O conjunto de documentos abarca seis arquivos independentes, cuja união totaliza 1.239 instâncias. Em contrapartida, o conjunto de consultas é composto por um total de 100 consultas, cada uma acompanhada de seus respectivos documentos relevantes e o texto da própria consulta.

[1]https://sites.google.com/ufla.br/denilsonpereira/

2.2. Pré-processamento

O pré-processamento de texto é um estágio crucial no processamento da linguagem natural, ocorrendo antes da modelagem textual para aplicação de algoritmos de aprendizado de máquina. Este procedimento de preparação do texto é fundamental para a análise e extração eficaz de informações. No âmbito deste trabalho, a etapa de pré-processamento abarcou as seguintes fases: *lower*, lematização, remoção de *stop words*, remoção de pontuação e remoção numérica.

2.3. Técnicas de Vetorização

A vetorização de texto envolve a transformação de dados de texto em representações numéricas, chamadas de vetores [Le and Mikolov 2014]. Essas representações são essenciais, já que muitos algoritmos de aprendizado de máquina requerem entradas numéricas. Os vetores resultantes podem ser processados por esses algoritmos, possibilitando análise e interpretação dos dados. Neste trabalho, foram utilizadas duas técnicas de vetorização: TF-IDF e *Embeddings*.

O TF-IDF é uma métrica estatística que avalia a relevância de cada palavra em relação a uma coleção de documentos, destacando termos raros no corpus [Ramos 2005]. Essa representação resulta em vetores, onde cada dimensão representa um termo e seu valor indica a importância do termo no documento. Por outro lado, os *Embeddings* são representações vetoriais de palavras ou frases obtidas de grandes conjuntos de dados textuais. Modelos de Processamento de Linguagem Natural, como o *Sentence-BERT*, geram *Embeddings* que capturam semântica e contexto [Smith and Johnson 2022].

Após o processamento das informações a serem recuperadas, os documentos devem ser indexados de maneira eficiente em uma estrutura apropriada, para a recuperação. Para esse propósito, utiliza-se a biblioteca FAISS, cuja apresentação detalhada é abordada na próxima subseção.

2.4. FAISS

O FAISS é uma biblioteca de código aberto voltado para pesquisas de similaridade. Escrito em C++ e *Python*, é extensamente empregada em aplicações de aprendizado de máquina, visão computacional e processamento de linguagem natural. O FAISS oferece uma gama de índices para busca de similaridade, criando índices baseados em árvores, tabelas *hash* e em aprendizagem de máquina. Um desses índices fundamentais é o *IndexFlatIP*. Esse índice demonstra eficácia notável em pesquisas que dependem da similaridade avaliada através do produto interno, ou seja, a similaridade do cosseno. Na próxima subseção, abordaremos as métricas de avaliação que permitem a quantificação da eficácia desse tipo de sistema.

2.5. Métricas de Avaliação

As métricas de avaliação são utilizadas para avaliar e comparar objetivamente os sistemas de recuperação. Elas oferecem uma medida da qualidade dos resultados de um sistema, permitindo aprimorar o desempenho do sistema e ajustar alguns parâmetros quando necessário. A seguir é apresentado um resumo dessas métricas:

- precisão: representa a proporção de documentos relevantes recuperados em relação ao total de documentos recuperados.

- revocação: indica a proporção de documentos relevantes recuperados em relação ao total de documentos relevantes no conjunto de dados. É uma medida da capacidade do sistema de recuperação em localizar todos os itens relevantes.
- precisão-R: avalia a qualidade dos resultados, sendo especialmente relevante quando se está interessado na relevância dos documentos recuperados. É a proporção dos documentos relevantes (ou itens relevantes) sobre o total de documentos recuperados.
- P@K: mensura a precisão dos primeiros K documentos recuperados, sendo a proporção de documentos relevantes entre os K primeiros documentos recuperados.
- MRR(Q): é uma métrica que avalia a eficácia de sistemas de recuperação de informação, especialmente em cenários de recuperação de documentos ou itens onde a relevância é crucial. A MRR(Q) é uma variação do *Mean Reciprocal Rank* (MRR) padrão, no qual o "Q" indica a posição máxima de interesse para calcular a média do recíproco da posição. Neste trabalho, considera-se um Sh=5, implicando que apenas os primeiros 5 documentos relevantes são considerados para o cálculo do MRR.

A visualização dos dados e dos resultados é fundamental para compreender as saídas dos algoritmos e, consequentemente, auxiliar na escolha adequada de um mecanismo de busca. A seguir, são apresentados os gráficos gerados para avaliar os resultados, com a interpretação de cada tipo de gráfico:

- Precisão x Revocação para 11 níveis de revocação: mostra a relação entre precisão e revocação em 11 pontos de revocação diferentes (10% a 100% das consultas), proporcionando um entendimento do equilíbrio entre precisão e revocação.
- Histograma da precisão-R para as 20 primeiras consultas: exibe a distribuição da precisão em diferentes valores de R (limite de *rank*) para as primeiras consultas, fornecendo uma visão da variabilidade da precisão nos primeiros resultados e facilitando comparações entre as máquinas de busca.

3. Resultados

Analisou-se duas máquinas de busca usando indexação com TF-IDF e *Embedding*. A simulação foi conduzida com a biblioteca FAISS, utilizando o método de indexação *IndexFlatIP*, que se baseia na similaridade do cosseno como mecanismo principal.

3.1. Escolha do K

Parâmetro k no *IndexFlatIP*, é importante para determinar o número de documentos a serem recuperados. Para essa definição, foi analisado as características da amostra de consultas, valor mínimo(2), valor máximo(304), incluindo média (48.19), moda (17), mediana (32) desvio padrão (49), primeiro quartil(17), terceiro quartil(55.5) e intervalo interquartil (38.5).A Figura 1 mostra o *boxplot* para entender a dispersão desses dados. Como adicional, foi incrementado os valores de Q1, Q3 e o IQR.

Após a análise, concluí-se que definir um número k, como sendo a média de documentos retornados pelo conjunto de consultas, não é ideal para este tipo de base de dados, pois a base apresenta alguns *outliers*, resultando na dispersão dos dados. No entanto, ao escolhermos o Q1 como limite, estamos focando os resultados apenas nos 25%

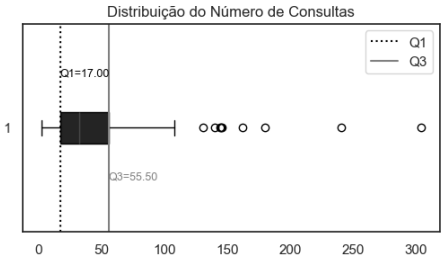

Figure 1. *Boxplot* da Quantidade de Documentos Relevantes.

dos dados, o que é pouco. Se optarmos pelo valor de corte Q3, estamos satisfazendo as consultas para 75% da base, entretanto, deixaríamos de incluir 25% das consultas, além de poder, trazer ruído para as demais consultas. O IQR é um número que fica entre Q1 e Q3, assim é possível equilibrar o *trade-off* para o sistema entre não retornar muitos documentos irrelevantes quando não seria necessário um grande retorno. Portanto, o mais indicado é retornar no mínimo 38 documentos (IQR), pois, esse número representa melhor a quantidade de documentos relevantes para cada registro no conjunto de dados completo. Isso também satisfaz o mínimo de consultas para calcular o p@5 e p@10 estipulado na descrição do trabalho prático.

3.2. Tempo

O tempo é uma variável importante em uma máquina de busca, por isso foi avaliado o tempo de transformação dos dados para cada técnica de vetorização, bem como o tempo de busca. Para a técnica de *Embedding*, para a vetorização temos um tempo de execução de 63.55s e para a busca de 0.004s. Já para a técnica de TF- ID, para vetorização temos o tempo de 0.16s e para a busca 0.12s. Apenas na etapa de pré-processamento é que o TF-IDF se sobresai em relação à técnica de *Embedding*, o que para base de dados maiores pode ser ainda mais complexo. Todavia, em quesito busca, a técnica de *Embedding* chega a ser 30 vezes mais rápido.

3.3. Métricas de Avaliação

Após a busca, foi analisado o resultado das distâncias, apresentadas na 2. É interessante observar que a média das distâncias para o TF-IDF é bem menor, enquanto para os *Embeddings* as distâncias possuem uma amplitude maior. As estatísticas sobre os resultados das consultas, podem ser visualizadas na Tabela 1.

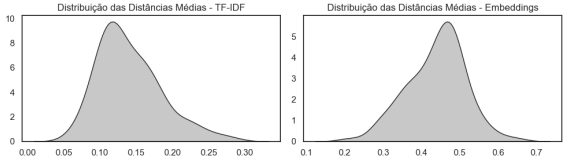

Figure 2. Distribuição da Média das Distâncias

Quando se trabalha com métricas baseadas em distâncias, é importante definir um *treshod* de aceitação, isto é, qual é a fronteira de aceitação do resultado? Será baseado nessas distâncias, ou somente no número k, conforme na Subseção 3.1. Para este trabalho,

Table 1. Estatísticas das Distâncias.

Métricas	***Embeddings***	**TF-IDF**
mínimo	0.1833	0.0330
máximo	0.7553	0.6535
média	0.4384	0.1408
desvio padrão	0.0401	0.0573

foi definido somente a escolha do k como estudo para aceitar ou não um documento aprovado em sua consulta.

A seguir na Tabela 2 é apresentado o restante das métricas, fundamentais para distinção do desempenho e prisão entre às duas máquinas de busca.

Table 2. Métricas utilizando o Embedding.

Métricas	***Embeddings***	**TF-IDF**
precisão média	0.2461	0.2592
revocação média	0.2670	0.2927
precisão-R	0.3359	0.3745
p@5	0.47	0.52
p@10	0.39	0.46
MRR(5)	0.88	0.93

Os resultados apresentados para precisão e revocação em contexto geral, são similares, o que não diferenciaria tão bem as técnicas de busca. Entretanto, quando se começa a restringir o espaço de busca e trazer as métricas relacionadas ao *ranking*, a diferença entre os dois sistemas começa a aparecer, onde a precisão do TF-IDF é o superior. A figura 3 apresenta a comparação dos dois sistemas baseando-se na métrica precisão-R. Neste quesito, ambos os sistemas possuem comportamento muito similar, a medida que vai aumentando o espaço de busca dos documentos retornados por consulta, a precisão vai diminuindo, certamente por trazer mais resultados fora do esperado.

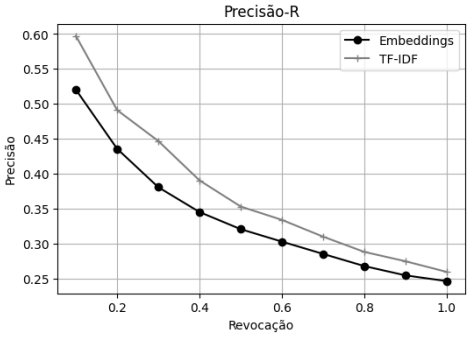

Figure 3. Comparativo da Precisão-R

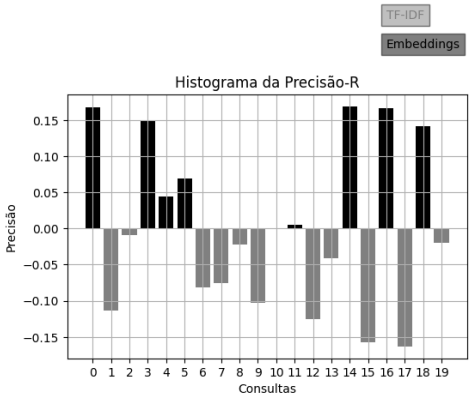

Figure 4. Histograma da Precisão-R.

Além disso, é possível avaliar o sistema de acordo com um número de consultas

em específico, no trabalho foi indicado a avaliação isolada das 20 primeiras consultas. Para este tipo de avaliação, usa-se o Histograma da Precisão-R, onde é possível avaliar o desempenho dos sistemas para identificar para quais consultas o sistema trará melhores resultados. Essa análise está indicada na Figura 4.

Na representação do histograma, temos que sistema TF-IDF tem melhor desempenho a partir da consulta 5, enquanto para as primeiras consultas, a técnica de *Embeddings* é melhor representada. Ademais, para algumas consultas como, por exemplo, as consultas 2, 10 e 11, podem ser representadas muito bem pelos dois sistemas.

4. Conclusão

Em Sistemas de Recuperação da Informação é possível utilizar várias técnicas de pré-processamento e indexação de documentos. Este trabalho analisou duas técnicas conhecidas, o TF-IDF e a de *Embeddings*, utilizando o Modelo SBERT. Os resultados se mostraram bastante promissores em relação ao conjunto total de consultas, onde precisão e revocação de ambos os sistemas similares. Entretanto, ao avaliar a precisão-R houve diferença entre os dois sistemas, e ao calcular o MRR(Q) observou-se para este conjunto de dados uma melhor aplicação da técnica TF-IDF com MRR(Q) de 0.93 enquanto o de *Embeddings* foi de 0.88. Entretanto, para avaliar o *trade-off* entre duas máquinas de busca, é necessário entender o contexto da aplicação, para entender qual métrica deve ser mais precisa para avaliar e quais requisitos o sistema deve preencher para satisfazer a definição de um bom sistema de busca.Por exemplo, em quesito tempo, o TF-IDF é mais rápido quanto ao pré-processamento, mas quando o assunto é a busca fica em até 30 vezes atrás do *Embeddings*. No caso da avaliação, das 20 primeiras consultas, o sistema de *Embeddings* não é tão eficiente, no geral, mas atua muito bem para as 5 primeiras consultas do problema, entretanto, para as consultas $2, 10 e 11$, o desempenho dos sistemas chegam a ser equivalentes. Apesar disso, ainda há espaço para explorar os resultados e testes, principalmente ao avaliar e analisar os resultados da similaridade do cosseno. Explorar esse ponto em específico, e trazer uma limitação não só dos *top* K documentos, mas também com um *treshold* mínimo para aceitação de um documento, pode alterar o desempenho de ambos os sistemas.

References

Facebook (2023). Faiss: A library for efficient similarity search. `https://github.com/facebookresearch/faiss`.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Ramos, J. (2005). Using tf-idf to determine word relevance in document queries. *Institute of Electrical and Electronics Engineers Transactions on Knowledge and Data Engineering*, Volume 17(Número 9):1264–1273.

Salton, G. (1971). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, New York, NY, USA.

Smith, J. and Johnson, M. (2022). Exploring embeddings: A preference for sentence-bert. *Journal of Natural Language Processing*, Volume 25(Número 3):450–465.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Amsterdam, Netherlands.