CAPÍTULO 3
DESENVOLVIMENTO E EXPERIMENTO: BM25 + BERTOPIC

3.1. Introdução e problema de pesquisa

Partimos do problema de incompatibilidade de vocabulário ("*vocabulary mismatch*") em que os usuários usam termos de consulta diferentes daqueles usados em documentos relevantes. Esse é um dos desafios centrais sobre recuperação de informação (*Information Retrieval*) (Nogueira *et* al, 2019, p.1).

Nosso objetivo é melhorar a recuperação de informação (RI) de um sistema de busca para uma situação específica. Partimos da hipótese de que, se um usuário está interessado em um documento específico de um grupo, também interessa a ele que retorne outros documentos do mesmo grupo (Moura, 2009, p.17 *apud* Chakrabarti, 2003; Kobayashi e Aono, 2004).

Nossa proposta foi utilizar uma técnica de expansão dos documentos com termos que são representativos do conteúdo dos documentos, compondo um novo campo no sistema de busca com os termos extraídos. Criamos esse campo através da modelagem de tópicos dos nossos documentos melhorado por modelos de embeddings pré-treinados (BERT).

Para isso, criamos um ambiente experimental que utiliza o Elasticsearch para recuperação de informação. O Elasticsearch tem como base o algoritmo BM25 (um algoritmo clássico de RI baseado em TF-IDF) (Beiske, 2013). Nesse sentido, obtemos um *framework* que é composto de (BERT + modelo de tópicos) + BM25.

3.2. Metodologia

A partir desse ambiente experimental, podemos comparar os resultados da RI entre:

(a) Nosso baseline, composto dos documentos originais;

(b) Os documentos enriquecidos com termos extraídos de um modelo de tópicos, compondo um campo no sistema de busca com os novos termos;

Optamos por uma variação do modelo de tópicos que incorpora também modelos de embeddings pré-treinados baseados no BERT, de forma a obter tópicos com melhor conteúdo semântico. Para isso, é utilizada a biblioteca de Python chamada BERTopic (Grootendorst, 2022) e o modelo pré-treinado "*distiluse-base-multilingual-cased-v1*".

Será utilizada a métrica de entropia para comparar os resultados de (a) e (b).

3.2.1. Função de Ranking - BM25

Precisamos encontrar a relevância dos termos de queries e documentos para constituir um sistema de busca (RI). "Contagem de palavras" não é uma métrica suficiente para encontrar a relevância dos termos. Geralmente, os termos que mais aparecem são irrelevantes, como por exemplo conjunções (a, o, da, do etc) (Jimenez *et* al, 2018, p.2888). Uma maneira de contornar isso e encontrar os termos mais relevantes é através de algoritmos como TF-IDF (*term frequency–inverse document frequency*). Assim, conseguimos filtrar os termos que aparecem muito em todos os documentos, porque entendemos que eles são irrelevantes; como consequência, damos maior peso para os termos que aparecem bastante em alguns documentos e podemos ver a distribuição deles ao longo do corpus de documentos.

$$\text{TF-IDF} = \ tf(w,d)\ x\ idf(w,D)$$

Enquanto TF-IDF como um modelo de vetorial favorece a frequência de termos e penaliza a frequência de documentos, desconsidera também o tamanho dos documentos e a saturação da frequência dos termos (Seitz, 2022). Outro modelo, chamado BM25, é uma proposta para resolver esta limitação usando um modelo probabilístico. Assim, enquanto a frequência de termo é controlada por uma função de saturação para impedir o seu crescimento linear, o parâmetro usado para calcular o tamanho médio dos documentos penaliza os documentos longos com alta frequência de termos buscados (Jimenez *et* al, 2018, p.2888).

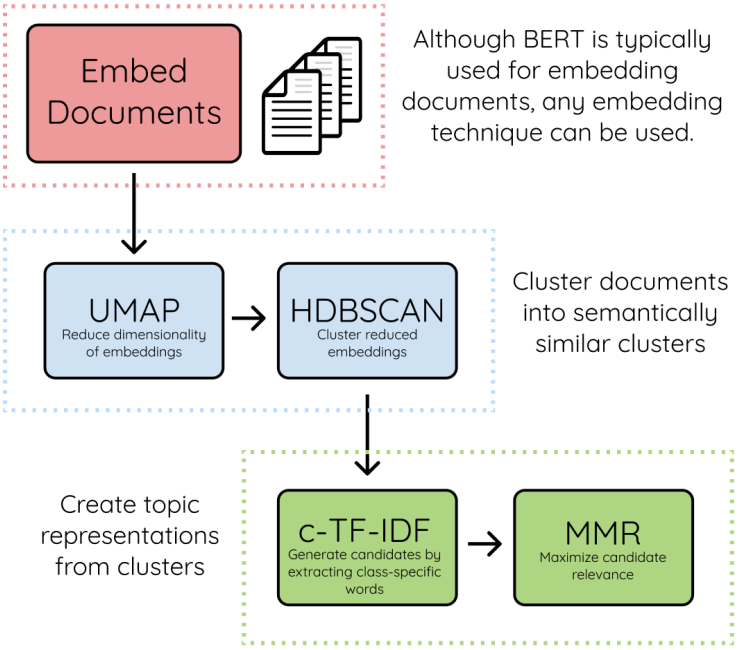
$$\text{BM25 :} \qquad \sum_{w \epsilon Q} idf(w)\ x\ \frac{(k1+1)\ x\ tf(w,d)}{k1\ x\ K(d)+tf(w,d)}\ x\ \frac{(k3+1)\ x\ tf(w,q)}{k3+tf(w,q)}$$

BM25 é o algoritmo mais utilizado em sistemas de busca pelo seu *tradeoff* de acurácia e performance computacional, incorporado em sistemas baseados no Lucene (como o Elasticsearch e Solr).

3.2.2. Geração de Tópicos

Nosso baseline (a) utiliza os documentos originais sem enriquecimento semântico. Para enriquecê-los, geramos tópicos (conjunto de palavras hierarquizadas) que irão compor um novo campo no sistema de busca com os termos extraídos dos tópicos.

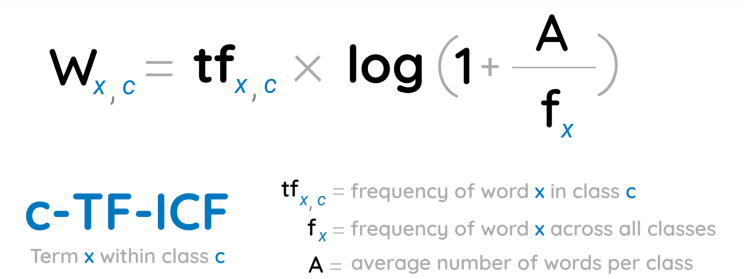
Para isso, utilizamos o *framework* da biblioteca BERTopic (Grootendorst, 2022) para criar nosso modelo de tópicos. Ele passa por algumas etapas para produzir nossos tópicos.



Em um primeiro momento, aproveitamos um modelo de *embeddings* pré-treinado para projetar os nossos documentos nesse espaço vetorial e extrair os seus vetores utilizando a biblioteca *sentence-bert*. O modelo pré-treinado é utilizado para melhorar a qualidade de nossos vetores do que seria o caso se usássemos apenas o nosso conjunto de documentos para constituir nosso espaço vetorial (Nogueira *et al*, 2019).

Em um segundo momento, aplica-se o algoritmo UMAP para redução de dimensionalidade do nosso espaço vetorial. Em seguida, aplica-se o algoritmo HDBSCAN para clusterização dos nossos documentos (Grootendorst, 2022).

Por fim, aplicamos a fórmula CTF-ICF (uma variação da fórmula TF-IDF) para os clusters desenvolvidos na fase anterior. Assim, cada cluster é convertido em um único documento, em vez de um conjunto de documentos. De cada cluster, extraímos a frequência da palavra x no cluster c, onde c se refere ao cluster que criamos anteriormente. Isso resulta em nossa representação TF baseada em cluster. Como no TF-IDF clássico, multiplicamos TF por IDF para obter a pontuação de importância por palavra em cada classe (Grootendorst, 2022).



O TF-IDF é usado para comparar a importância das palavras entre todos os documentos do nosso corpus. Em vez disso, tratamos apenas dos documentos em cada cluster e depois aplicamos o TF-IDF respectivamente para cada cluster como se fosse um documento (Grootendorst, 2022). O resultado seria a medida de importância para palavras dentro de um cluster. Quanto mais palavras importantes estiverem dentro de um cluster, mais ele será representativo daquele tópico. Em outras palavras, se extrairmos as palavras mais importantes por cluster, obteremos descrições de tópicos. Este modelo é chamado de TF-IDF baseado em cluster (CTF-ICF) (Grootendorst, 2022).

Uma vez que temos o nosso modelo de tópicos, podemos inferir qual é o tópico mais importante para cada documento. Assim, expandimos nossos documentos, nos aproveitamos dos termos mais importantes do seu tópico relativo com o documento, para compor um novo campo no sistema de busca que possa melhorar nossa recuperação de informação.

3.3. Critérios de Avaliação

3.3.1. Datasets

Em 2013, o Instituto de Ciências Matemáticas e da Computação da Universidade de São Paulo (ICMC-USP) tornou disponíveis conjuntos de documentos para avaliação de experimentos computacionais (Rossi, Marcacini, Rezende, 2013). Escolhemos o dataset de computação extraído *Open Directory Project* (*Dmoz-Computers-500 Collection*) (Netscape *apud* Rossi, Marcacini, Rezende, 2013). Sua distribuição possui as seguintes características:

Dmoz-Computers-500 collection characteristics.

Domain	Web Pages
# Documents	9500
# Terms	5011
$\overline{\#}$ Terms	10.83
Matrix Sparsity	99.78%
# Classes	19
Class S.D.	0.00%
Majority Class	5.26%
S-Index	

Fonte: (Rossi, Marcacini, Rezende, 2013)

Cada documento possui sua classificação. Foram escolhidos 500 documentos estratificados em 19 categorias. São essas classificações que serão usadas para computar a qualidade do nosso information retrieval.

Class Labels	Abs. # Docs.	Rel. # Docs.	S-Index
Artificial_Intelligence	500	5.26%	0.544
CAD_and_CAM	500	5.26%	0.570
Companies	500	5.26%	0.562
Computer_Science	500	5.26%	0.714
Consultants	500	5.26%	0.582
Data_Communications	500	5.26%	0.632
Data_Formats	500	5.26%	0.664
Education	500	5.26%	0.806
Graphics	500	5.26%	0.832
Hardware	500	5.26%	0.578
Internet	500	5.26%	0.626
Mobile_Computing	500	5.26%	0.598
Multimedia	500	5.26%	0.512
Open_Source	500	5.26%	0.596
Programming	500	5.26%	0.524
Robotics	500	5.26%	0.796
Security	500	5.26%	0.506
Software	500	5.26%	0.320
Systems	500	5.26%	0.508

Fonte: (Rossi, Marcacini, Rezende, 2013)

3.3.2. Métricas

Em aplicações de recuperação de informação é necessário ter uma medida de confiança para testar os resultados alcançados. A entropia (incerteza) fornece uma medida valiosa na verificação de sistemas de informação, que é calculado através de conjunto dos K documentos mais bem classificados para uma consulta (Grivola *et* al., 2005). É esperado que bons resultados de recuperação proporcionem um conjunto de documentos mais homogêneos. Portanto, a entropia do conjunto de documentos deve ser menor quando o desempenho obtido para uma determinada consulta for bom.

Se a entropia do conjunto for alta, a estrutura linguística dos documentos é altamente variável. Os documentos com uma grande variabilidade linguística geram maior risco de que alguns documentos recuperados sejam irrelevantes. A probabilidade de recuperar documentos irrelevantes aumenta com K (Grivola *et* al., 2005). Como um único documento longo e não relevante pode causar um aumento significativo na entropia, é aconselhável manter K pequeno (Grivola *et* al., 2005). Considerando estes fatores, usamos a fórmula seguinte de entropia:

$$H = -\sum_{c}^{K} nf(c)\, x\, \frac{log(nf(c))}{log(K)}$$

Em nossa fórmula do cálculo da entropia *H*, *nf(c)* representa a frequência normalizada de cada classe recuperada e *K* é a quantidade de classes encontrada na busca.

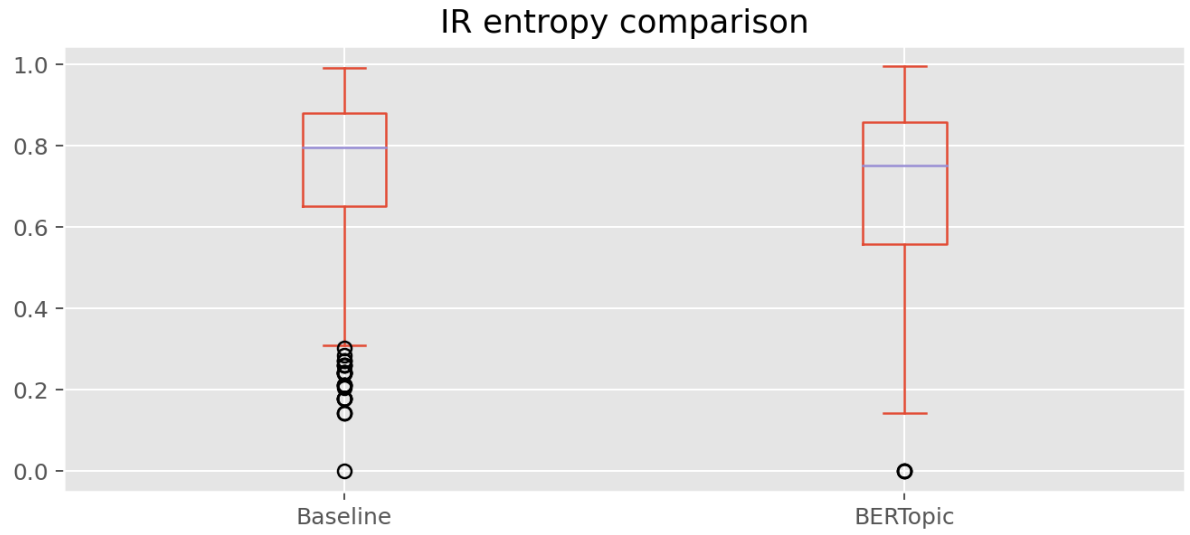
3.4. Experimentos Preliminares

Conduzimos nosso experimento em um Jupyter Notebook usando a linguagem Python. Começamos configurando um servidor de Elasticsearch, criando um ambiente de BM25 para realizar nossos testes empíricos. Alimentamos o sistema com os nossos dados textuais originários do dataset *Dmoz-Computers-500*.

Primeiro, fizemos nosso experimento com nosso modelo baseline, ou seja, nossos documentos originais sem enriquecimento semântico. Testamos apenas uma query ('*neural networks for linux systems*') no nosso sistema de busca e calculamos a entropia a partir da distribuição das classes no resultado da recuperação de informação. Em seguida, encontramos bi-gramas mais importantes do nosso conjunto de documentos para usarmos cada um deles como uma nova query, reservando os resultados obtidos e o cálculo de entropia para cada uma delas.

Na segunda parte do nosso estudo, transformamos nossos dados em modelo de tópicos. Começamos eliminando *stopwords*, depois vetorizamos os nossos documentos. Em seguida, reduzimos a dimensionalidade do nosso espaço vetorial com o algoritmo UMAP e clusterizamos os vetores dos nossos documentos com o algoritmo HDBSCAN. Por fim, criamos um modelo de tópico CTF-ICF. Uma vez que tivemos nosso modelo de tópicos, enriquecemos os documentos com as palavras mais relevantes do tópico mais relevante de cada documento, criando um novo campo no sistema de busca com os termos dos tópicos. Repetimos o processo aplicado para o baseline, mas agora com o novo campo, ou seja, criamos bi-gramas para usar como queries e reservamos os resultados obtidos com nosso modelo enriquecido para calcular e entropia em cada uma das queries.

Assim que tivemos os scores do nosso baseline e nossa hipótese pudemos compará-los.



Como podemos observar em nosso gráfico de *boxplot*, a média da entropia dos documentos enriquecidos foi mais baixa do que nosso baseline (0,68 e 0,74 respectivamente), lembrando que quanto menor a entropia melhor o nosso resultado. Além disso, o terceiro quartil obteve score mais baixo nos documentos enriquecidos. Podemos concluir então que o modelo de hipótese, (BERT + topic model) + BM25, melhorou os resultados da busca. No futuro, conduziremos novos experimentos buscando outros parâmetros que poderão gerar resultados ainda melhores.

REFERÊNCIAS BIBLIOGRÁFICAS

Beiske, K. (2013) "Similarity in elasticsearch," *Elastic Blog*. Elastic, 26 November. Available at: https://www.elastic.co/pt/blog/found-similarity-in-elasticsearch (Accessed: July 3, 2022).

Grivolla, J., Jourlin, P. and De Mori, R. (no date) *Automatic classification of queries by expected retrieval performance*, *Grivolla.net*. Available at: http://www.grivolla.net/articles/sigir2005-qp.pdf (Accessed: July 3, 2022).

Grootendorst, M. P. (no date) *The algorithm*, *Github.io*. Available at: https://maartengr.github.io/BERTopic/algorithm/algorithm.html (Accessed: July 3, 2022).

Jimenez, S. *et* al. 'BM25-CTF: Improving TF and IDF Factors in BM25 by Using Collection Term Frequencies'. 1 Jan. 2018 : 2887 – 2899. Available at: https://www.researchgate.net/profile/Sergio-Jimenez-7/publication/325231406_BM25-CTF_Improving_TF_and_IDF_factors_in_BM25_by_using_collection_term_frequencies/links/5b0

d8349aca2725783f140e5/BM25-CTF-Improving-TF-and-IDF-factors-in-BM25-by-using-collection-term-frequencies.pdf (Accessed: July 3, 2022).

Kamal, A. (2021) *Building your favourite TV series search engine - Information Retrieval Using BM25 Ranking*, *Medium*. Available at: https://abishek21.medium.com/building-your-favourite-tv-series-search-engine-information-retrieval-using-bm25-ranking-8e8c54bcdb38 (Accessed: July 3, 2022).

Manning, Christopher D., et al. (2008) *Introduction to information retrieval*. Cambridge University Press. Available at: https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf (Accessed: July 3, 2022).

Moura, M. F. (2009). 'Contribuições para a construção de taxonomias de tópicos em domínios restritos utilizando aprendizado estatístico'. Tese de Doutorado. Instituto de Ciências Matemáticas e da Computação da Universidade de São Paulo (ICMC-USP), São Carlos. Available at: https://teses.usp.br/teses/disponiveis/55/55134/tde-05042010-162834/publico/MFM_Tese_5318963.pdf (Accessed: July 3, 2022).

Nogueira, R. *et al.* (2019) "Document expansion by query prediction," *arXiv [cs.IR]*. Available at: http://arxiv.org/abs/1904.08375 (Accessed: July 3, 2022).

Open directory project.org: ODP web directory built with the DMOZ RDF database (no date) *Odp.org*. Available at: http://www.odp.org/homepage.php (Accessed: July 3, 2022).

Seitz, R. (no date) *Understanding TF-IDF and BM-25*, *KMW Technology*. Available at: https://kmwllc.com/index.php/2020/03/20/understanding-tf-idf-and-bm-25/ (Accessed: July 3, 2022).

Yates, A., Nogueira, R., & Lin, J. (2021). Pretrained transformers for text ranking: BERT and beyond. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Available at: https://arxiv.org/pdf/2010.06467v1.pdf (Accessed: July 3, 2022).