Análise da Robustez de Mecanismos de Busca quanto a Ruídos em Consultas

Leonardo Augusto da Silva Pacheco e Marcus Vinícius Borela de Castro

Dezembro 2021

Resumo

O trabalho presente analisa o impacto que ruídos inseridos em consultas podem gerar na efetividade de mecanismos de buscas.Foram gerados diversos tipos de ruídos sobre 54 consultas. Para cada mecanismo empregado, foram comparados os resultados para consulta com ruído em relação à consulta original, a partir da métrica proposta DG:nDCG@10 que traduz perdas e ganhos de efetividade. Foram encontrados variados impactos dependendo da query e do ruído gerado nos diversos mecanismos de pesquisa avaliados.

1 Introdução

Desde a introdução das redes neurais profundas e, em particular, dos novos modelos de transformers, diversos mecanismos novos de pesquisa vêm sendo propostos, entre eles, mecanismos de Rerank (MonoT5, MonoBert e DuoBert), mecanismos de busca densa (DPR [1]) e técnicas de enriquecimento de conteúdo (doc2query). Esses mecanismos têm promovido resultados positivos em *benchmarks[[1]](#footnote-0)* de Recuperação de Informação[2]. Nesse contexto, outros pontos devem ser considerados no processamento de buscas, como o tempo de resposta e o uso de recursos computacionais.

O objetivo do presente trabalho é analisar o impacto que ruídos inseridos em consultas (*queries*) podem gerar na efetividade de mecanismos de buscas. Um ruído é uma alteração no texto de uma *query*, como exclusão ou troca de palavras. Uma mudança pode impactar o resultado da pesquisa, por exemplo, deixando de trazer documentos relevantes nas primeiras posições. Buscou-se avaliar o impacto em termos de perda ou ganho em relação à efetividade da consulta original..

2 Conjunto de Dados

Utilizamos a base[[2]](#footnote-1) *TREC 2020 DL MsMarco Passage* com suas 8.841.823 passagens e 54 queries que possuem uma média de 210,9 julgamentos de relevância.

3 Metodologia

Como era objetivo replicar as análises para textos em português e por não haver GPU com fácil acesso, gerou-se uma base menor apenas com as 11.224 passagens que possuem julgamento (*TREC 2020 MsMarco Passage with judment*) em uma instância local *Elastic Search*. Essa base menor foi utilizada para as buscas densas (DPR) tanto em português quanto em inglês. Para se gerar a base em português, foram traduzidos os 11.224 documentos usando-se *library* python *googletrans*.

Os dados[[3]](#footnote-2) usados e gerados durante o projeto estão publicados no repositório do projeto como dataframes Pandas/Python. Também estão disponíveis diagramas entidade x relacionamento (MER) dos conceitos representados[[4]](#footnote-3).

Sobre as 54 queries originais aplicamos oito tipos de ruídos[[5]](#footnote-4):

* Permutação de palavras em determinadas posições da query: primeira e segunda, primeira e última e penúltima e última.
* Remoção de palavras: da segunda, da última e retirando palavras com probabilidade de 20% (pelo menos a palavra central).
* Back translation: tradução para outra língua (português/inglês) seguida de nova tradução de volta para o idioma original, gerando mudanças de palavras.
* Inserção de um erro tipográfico[[6]](#footnote-5), como trocando “what” por “waht”.

Executaram-se buscas tanto com as queries com os ruídos[[7]](#footnote-6), como com as queries originais, usando as funções de ranqueamento[[8]](#footnote-7): BM25 [3], BM25 + Rerank (MonoT5) e busca densa (DPR). Usando os julgamentos de relevância (com as queries originais) das dez primeiras passagens retornadas, calculou-se as métricas DCG@10 (*Discounted cumulative gain at 10 documents*) e nDCG@10 (*Normalized Discounted cumulative gain at 10 documents)*.

Para confrontar a efetividade da query ruidosa com a query original, foi proposta a métrica DG:nDCG@10 (Discount ou Gain on normalized Discounted cumulative gain at 10 documents):

, onde eps: 10e-6. Valores menores que zero representam desconto na efetividade, e maiores que zero representam ganhos na efetividade. No contexto desse trabalho chamaremos DG:nDCG@10 por DG.

Foram retirados os cálculos realizados na base em português referente a quatro modelos devido à pouca eficácia encontrada. Critério usado: mais de 23% das pesquisas com nDCG@10 zerado e média menor do que a metade do valor alcançado pela função BM25 em mesma base. Assim, foram descartados os modelos de Rerank[[9]](#footnote-8): ptt5-small-portuguese-vocab, ptt5-base-portuguese-vocab e ptt5-base-t5-vocab. E na busca densa, o modelo voidful/dpr-question\_encoder-bert-base-multilingual.

Para diminuir impacto de outliers no cálculo da métrica DG, optou-se por excluir 29 registros referentes a queries com ruído que obtiveram nDCG@10 superior a zero enquanto que a query original alcançou zero. E por excluir, na sequência, outros 25 registros que tiveram z-score absoluto superior a três (três vezes o desvio padrão).

4 Resultados

Os cálculos[[10]](#footnote-9) e os detalhes da análise[[11]](#footnote-10) estão publicados no github do projeto.

Os resultados das pesquisas de Rerank em português com 3 modelos diferentes obtiveram resultados próximos e por isso foram unificados pela média na análise que se segue. Cabe ressaltar que o modelo unicamp-dl/mt5-base-multi-msmarco alcançou menor DG (-0.068) em relação aos outros unicamp-dl/mt5-base-en-pt-msmarco (-0.100) e unicamp-dl/ptt5-base-pt-msmarco-100k (-0.091). Mas este último, se mostrou mais eficiente por consumir 63% menos memória (810 mb contra 2.17 gb) e ser 30% mais rápido.

O Gráfico 1 mostra os valores médios de DG para os diversos contextos de busca e tipos de ruído. Percebe-se maior robustez em relação a permutações e erros tipográficos, e maior sensibilidade a *back-translation* e remoções de palavras. A busca densa (DPR) apresentou menores descontos, seguida pelo Rerank e o BM25. A busca BM25 é plenamente robusta quanto às permutações, porém, em geral, é relativamente mais sensível às deleções. O Rerank mostrou-se mais sensível do que o BM25 para deleções e mais robusto em outros tipos de ruídos. Empataram com o maior ganho (0.55) o ruído de permutar a primeira e a segunda palavra na busca densa em inglês e o ruído de *back translation* na busca por BM25 na base em português. E os piores foram com o BM25 na deleção.

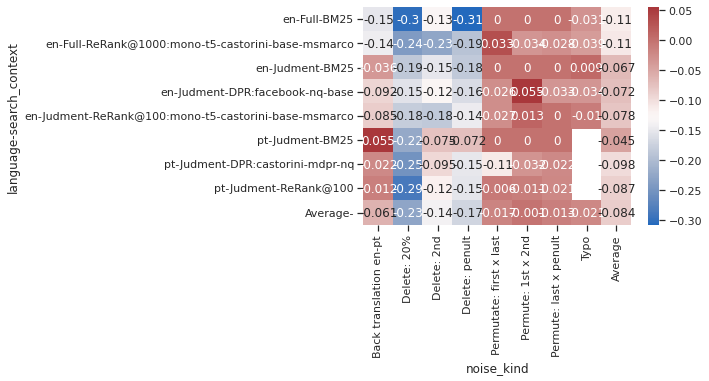


Gráfico 1 - Médias da métrica DG:nDCG@10 para vários mecanismos de busca e tipos de ruído

O Gráfico 2 demonstra o desvio padrão encontrado na métrica DG nas pesquisas. Percebe-se que os ruídos de permutação obtiveram uma variação menor. E que a busca com Rerank na base em português também teve valores em média menores em relação aos outros ruídos.

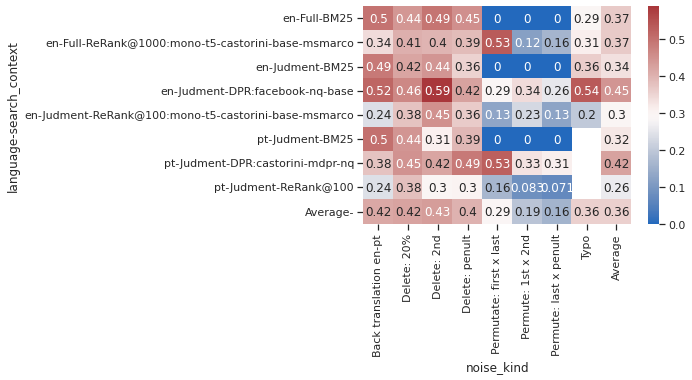


Gráfico 2 - Desvio padrão da métrica DG:nDCG@10 para vários mecanismos de busca e tipos de ruído

Os Gráficos 3 demonstra a distribuição dos valores da métrica DG por tipo de ruído. Percebe-se uma maior variação nas deleções, em que perdas maiores são identificadas. Em particular na deleção com 20% de probabilidade, em que mais palavras são retiradas.

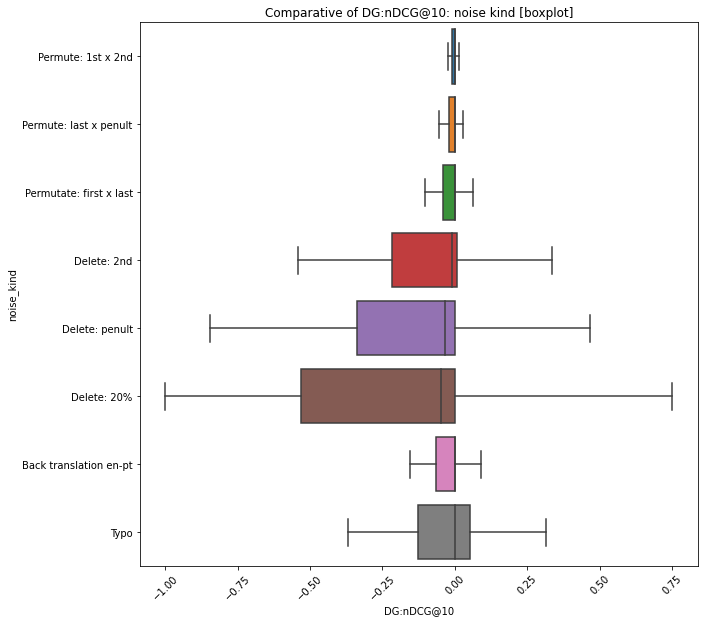


Gráfico 3 - Distribuições de DG:nDCG@10 por tipo de ruído

O Gráfico 4 traz as distribuições por contexto de pesquisa. Observam-se menores variações interquartis no caso da busca densa e de Rerank. Mas este sofre nas deleções, porém menos do que o BM25 que é extremamente sensível a deleções, mas totalmente robusto quanto a permutações.

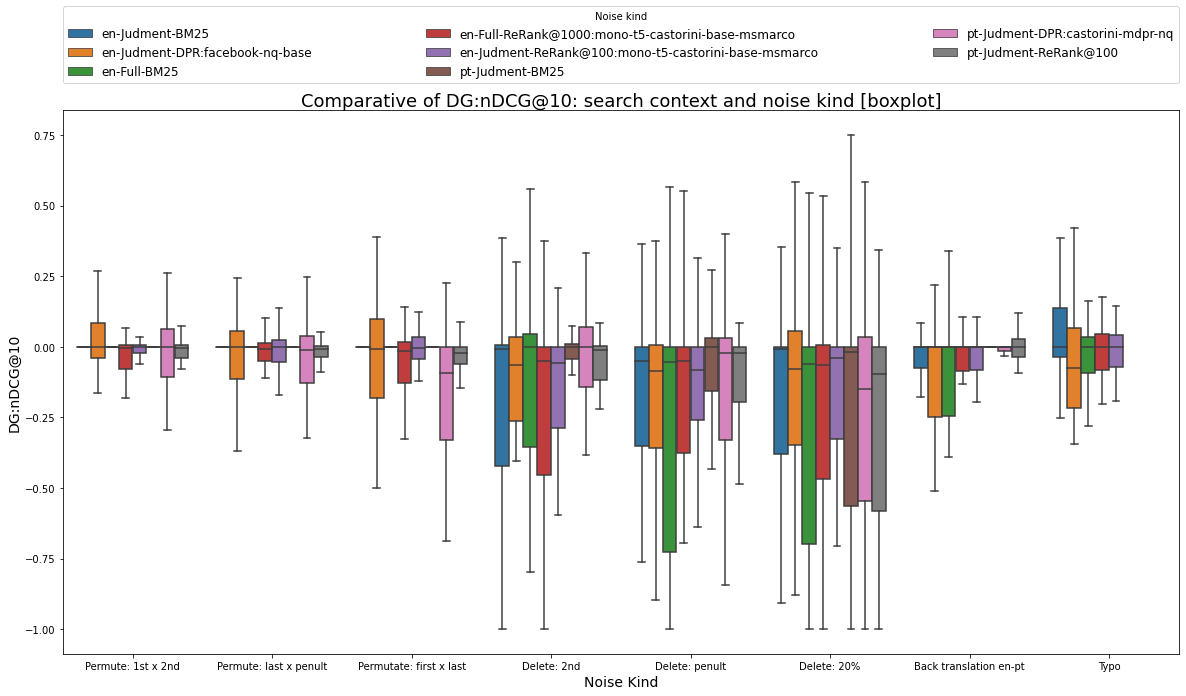


Gráfico 4 - Distribuições de DG:nDCG@10 para vários mecanismos de busca e tipos de ruído

O impacto de um determinado ruído em cada query varia. Por exemplo, no caso da remoção de uma palavra em uma determinada posição na query, se esta palavra for uma stopword, é possível que a resposta para a query ruidosa seja melhor do que a resposta para a query original. Por outro lado, se a palavra removida for um substantivo com significado chave para a intenção por trás da query, descaracteriza-se completamente a consulta. O gráfico 5 ilustra um caso desses, em que há queries onde há desconto reduzido ou mesmo ganhos, até queries com desconto máximo (-1).

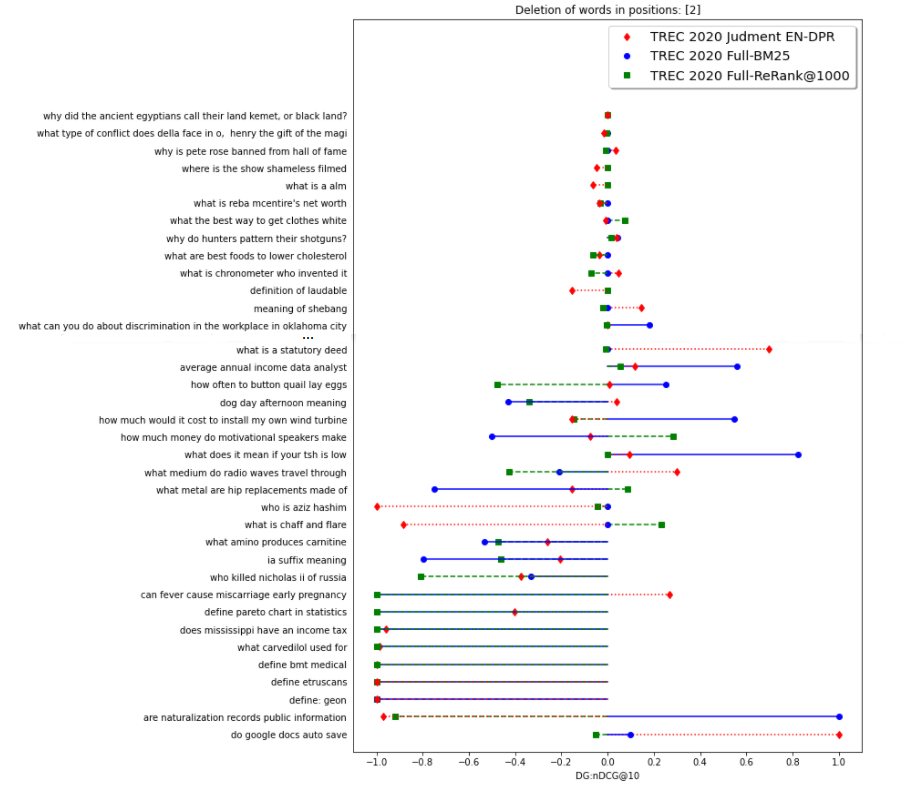


Gráfico 4 - Valores obtidos de DG:nDCG@10, quey por query (parcial)

Como tendência geral, observou-se que textos maiores de consulta têm menores variações de DG, sendo mais robustos aos ruídos experimentados, enquanto textos mais curtos apresentam resultados mais variados. O Gráfico 5 demonstra essa tendência. Pois há uma menor chance de se perder palavras mais relevantes.

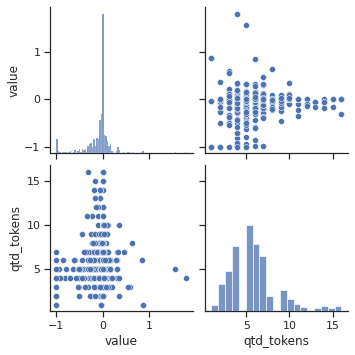


Gráfico 5 -Correlação = 0.166 entre quantidade de tokens e DG para contexto en-judment-DPR:facebook-nq-base

5 Conclusão

Em relação a permutações de palavras, o BM25, por sua forma de cálculo, é plenamente robusto. Para outros tipos de ruídos, porém as buscas densas e buscas com rerank realizadas apresentaram-se mais robustas na média e com menores desvios, em relação a buscas empregando rankeamento BM25. O treinamento de modelos baseados em transformers, por capturar significado das palavras e relação com o contexto, parecem ser mais robustos em relação a diversos tipos de ruído. Por exemplo, no caso de back-translation, uma palavra pode ser transformada em um sinônimo, o que normalmente não impacta de forma significativa o desempenho. Também pode contribuir para a robustez dos mecanismos de busca o próprio ruído que é encontrado nos documentos onde a pesquisa é executada. Além disso, nos procedimentos de treinamento dos modelos, são empregados ruídos como forma de regularização dos modelos, o que contribui ainda mais para a robustez.

Comparando a robustez obtida entre consultas diferentes, observou-se que textos maiores de consulta têm menores variações de DG, sendo mais robustas aos ruídos experimentados, enquanto textos mais curtos apresentam resultados mais variados. A possível explicação para isso é que em textos maiores há uma menor chance de se perder palavras mais relevantes.

6 Trabalhos Futuros

Como trabalhos futuros, vislumbram-se algumas possibilidades. Quanto aos modelos experimentados, pode-se buscar o uso dos quatro modelos descartados, com uma revisão de sua configuração e até mesmo um *fine-tunning* dos mesmos. Ou mesmo experimentar outros modelos. Pode-se também avaliar o impacto de mecanismos de regularização no treino de modelos na robustez quanto a ruídos.

Quanto aos tipos de ruídos, sugere-se aumentar o número de ruídos, por variação dos sugeridos ou por junção dos mesmos. Ou mesmo com uso de novos tipos.

Quanto às bases de documentos, havia uma restrição de uso de GPU o que levou ao uso de uma base menor na maioria dos contextos de pesquisa. Sugere-se repetir os experimentos com os modelos na base completa TREC 2020 MsMarco Passage.

Quanto à métrica considerada na performance, sugere-se avaliar o comportamento da métrica proposta DG sobre outras métricas. Por exemplo: DG:MRR (Mean Reciprocal Rank) e DG:MAP (Mean Average Precision).

Quanto à análise, análises mais detalhadas por tipo de função de ranqueamento ou por modelo usado ou por base de texto pesquisada ou por linguagem.Também cabe uma avaliação mais detalhada nos casos de ruídos que levaram a ganhos inesperados. Talvez floresçam novas técnicas de enriquecimento de índices para consultas com a aplicação desses ruídos.

Por fim, espera-se que cada vez mais maiores ganhos sejam alcançados em detrimento de menores descontos na desafiadora e, hoje, onipresente, área de recuperação da informação.

References

[1] [Dense Passage Retrieval for Open-Domain Question Answering. Vladimir Karpukhin and Barlas Oğuz and Sewon Min and Patrick Lewis and Ledell Wu and Sergey Edunov and Danqi Chen and Wen-tau Yih.2020.](https://arxiv.org/pdf/2004.04906)

[2] Pretrained Transformers for Text Ranking: BERT and Beyond. Jimmy Lin, Rodrigo Nogueira, and Andrew Yates (anotações)

[3] [The Probabilistic Relevance Framework: BM25 and Beyond.Stephen Robertson & Hugo Zaragoza. 2009](https://www.researchgate.net/publication/220613776_The_Probabilistic_Relevance_Framework_BM25_and_Beyond)

1. MS MARCO Passage Ranking Leaderboard em <https://microsoft.github.io/msmarco/> [↑](#footnote-ref-0)
2. <https://microsoft.github.io/msmarco/TREC-Deep-Learning-2020> [↑](#footnote-ref-1)
3. <https://github.com/leonardo3108/robustez-query/tree/main/data> [↑](#footnote-ref-2)
4. <https://github.com/leonardo3108/robustez-query/tree/main/model> [↑](#footnote-ref-3)
5. <https://github.com/leonardo3108/robustez-query/blob/main/data/tab_noise_kind.csv> [↑](#footnote-ref-4)
6. Apenas esse ruído não foi gerado para pesquisas em português (restrição de tempo no projeto). [↑](#footnote-ref-5)
7. <https://github.com/leonardo3108/robustez-query/blob/main/data/tab_noisy_query.csv> [↑](#footnote-ref-6)
8. <https://github.com/leonardo3108/robustez-query/blob/main/data/tab_ranking_function.csv> [↑](#footnote-ref-7)
9. <https://huggingface.co/unicamp-dl> [↑](#footnote-ref-8)
10. <https://github.com/leonardo3108/robustez-query/blob/main/data/tab_calculated_metric.csv> [↑](#footnote-ref-9)
11. [https://github.com/leonardo3108/robustez-query/blob/main/code/analysis/Analysis\_Discount\_Gain.ipynb](https://github.com/leonardo3108/robustez-query/blob/main/code/analysis/Analysis_Discount_Gain_pt.ipynb) [↑](#footnote-ref-10)