Recuperação de Informação de Bibliografias

Vinicius Matumoto Diogo
Universidade Ferederal de São Paulo,
Campos São José dos Campos
São José dos Campos, Brasil
vinicius.matumoto@unifesp.br

Matheus Ramos Esteves
Universidade Ferederal de São Paulo,
Campos São José dos Campos
São José dos Campos, Brasil
matheus.esteves31@unifesp.br

***Abstract*— O projeto tem como objetivo utilizar o modelo da Google chamado BERT (Bidirectional Encoder Representations from Transformers), para resolver um problema de recuperação de informação em dados bibliográficos, buscando descobrir os limites desse modelo para esse tipo de trabalho, e assim analisar o quão bom e/ou preciso esse modelo pode ser quando se trata da recuperação de informação. Por meio de testes, aplicando nuances nas entradas oferecidas ao modelo, como remoção de palavras das referências, remoção de caracteres e substituição de caracteres foi analisado que o modelo é robusto a poucas alterações.**

Keywords—BERT, Inteligência Artificial, Recuperação de Informação

I. Introdução

Com o início do desenvolvimento de inteligências artificiais, surgiram diversos modelos e projetos enormes, e um deles foi o BERT, desenvolvido pela Google em 2018 e que quebrou diversos recordes em tarefas de desempenho em tarefas de processamento de linguagens naturais. Em função desse alto desempenho e da necessidade de desenvolver um projeto a pedido do Dr. Thiago Macedo, surgiu a ideia de desenvolvermos um projeto que possa testar os limites desse modelo que demonstra ser tão robusto para a recuperação de informação em referências bibliográficas, onde dado uma referência deveríamos retornar à referência que mais condiz com a desejada.

II. Conceitos Importantes

Para esse projeto vai ser necessário que o leitor entenda algumas coisas e terminologias. De começo temos algo básico como entender o que são referências bibliográficas e entender que elas são conjuntos de informações que identificam as fontes de informações utilizadas em um trabalho acadêmico ou científico.

Muito importante também entender o que é o Processamento de Linguagem Natural, conhecida também como PLN, tomando como base o livro de 'Processamento de linguagem natural: conceitos, técnicas e aplicações em português' publicado em 2023, o leitor irá precisar entender que em PLN buscamos gerar soluções para problemas computacionais que tem como necessidade de tratar computacionalmente uma língua seja ela em texto ou fala.

Recuperação de Informação (RI) podemos usar o artigo Recuperação de informação e processamento da linguagem natural, postado no XXIII Congresso da Sociedade Brasileira de Computação em 2003, para vermos uma aplicação ou até mesmo o artigo Recuperação de Informação de Olinda Nogueira Paes Cardoso em 2000, Onde RI é o processo de obter informações relevantes de grandes repositórios de dados.

Web Scraping, elemento que vamos usar para podermos montar nossa base de dado, podemos utilizar como base o Web scraping using Python postado em 2022, que engloba grande parte do que vamos utilizar, sendo requests, pegar9 elementos da web usando o Selenium, entre outros.

III. Objetivo

Neste projeto existem dois objetivos a serem alcançados, sendo eles:

1. Entregar uma IA que busque solucionar da melhor forma o problema dado pelo professor Dr. Thiago Macedo. Ou seja, uma IA que tenha a capacidade de receber referências bibliográficas e fazer a separação dela em formato de autor, título e ano de publicação, para que seja um facilitador do trabalho do professor;

2. Fazer experimentos e testar os limites que o modelo BERT pode alcançar na recuperação de informação, para apresentar no Workshop de Inteligência Artificial. Utilizando de métricas já consagradas, para fazer uma análise de seu desempenho em vários cenários.

IV. Metodologia Experimental

A. Banco de Dados

Para a criação do modelo de Recuperação de Informação precisamos criar uma base de dados robusta, para tanto utilizamos de meios como Web Scraping e requisições HTTP.

Para o Web Scraping foi utilizado a biblioteca Selenium, que permite a criação de scripts em Python que permitem abrir sites na internet e conseguir de maneira mais simples as informações de alguns sites. E com esses scripts nós pegamos os links dos artigos presentes no site do Arxiv na área de Matemática, para podermos mapear quantos artigos seriam capturados e utilizados.

Existem artigos desde 1992, e inicialmente decidimos pegar apenas 3 temas, Dynamical Systems, Algebraic Geometry e Optimization and Control. Com isso foram gerados 3 arquivos de texto que possuem os links de cada artigo existente em cada um dos 3 temas, e esses links foram possíveis utilizando o Selenium.

O próximo passo após pegar o link de cada artigo seria o de utilizarmos as requisições HTTP onde estaríamos utilizando a biblioteca requests do Python em conjunto com uma pequena adaptação nos links obtidos de cada um dos artigos, nós obtemos com essas chamadas o BibTex dos artigos, é quem fornece ao Latex, por exemplo, como deve ser feita e citação e referenciação do artigo em questão.

Utilizamos a extensão de arquivo .pkl para armazenar as informações dos BibTex pôr ser mais leve e fácil de trabalhar, alterar e salvar. Com todos as citações em mãos, utilizamos a biblioteca pybtex que nos permite montar as referências e separar em autor, título e ano.

Com todas essas informações nós montamos nosso banco de dados em um Excel, que contém mais de 52 mil referências, com referência completa, autor, título e ano.

B. Tratamento de Dados

Após o banco de dados ter sido criado, fez-se necessário fazer o tratamento dos dados contidos, com a finalidade de contribuir para uma melhor tokenização e criação de *embeddings*. Para isso foi criado um *DataFrame* do banco de dados, e nele todas as referências tiveram seus caracteres especiais removidos, com exceção do '-'. Além da remoção dos caracteres especiais, também foi removido o sufixo do *id* que cada referência possuía no site. O resultado foi como exemplificado a seguir:

Antes: Wolfram Decker and Sorin Popescu. On surfaces in p^4 and 3-folds in p^5. 1994. arXiv:alg-geom/9402006.

Pós tratamento: Wolfram Decker and Sorin Popescu On surfaces in p4 and 3-folds in p5 1994

Dessa maneira o DF possuía pouco mais 52 mil linhas, e as referências com uma média de 14 palavras.

C. Modelo Bert

O BERT (Bidirectional Encoder Representations from Transformers) é um modelo de aprendizado profundo desenvolvido pela Google em 2018, que revolucionou o campo do Processamento de Linguagem Natural (PLN) ao utilizar a arquitetura de Transformers para processar e entender textos de maneira bidirecional. O grande diferencial do BERT é sua habilidade de capturar o contexto de palavras em ambas as direções (esquerda e direita) simultaneamente, permitindo uma compreensão mais profunda das nuances semânticas e contextuais em frases e parágrafos.

Para este projeto, utilizamos o BERT devido à sua capacidade de capturar nuances semânticas e contextuais, o que é essencial para a tarefa de Recuperação de Informação em um banco de dados bibliográfico.

O modelo BERT pré-treinado utilizado foi o **bert-base-uncased**, uma versão padrão que possui 12 camadas de transformadores, cada uma com 768 dimensões e 12 cabeças de atenção. Este modelo foi treinado em grandes corpora de texto, incluindo a Wikipedia e o BookCorpus, o que lhe confere uma vasta compreensão de diferentes domínios linguísticos. Ao não diferenciar maiúsculas de minúsculas (uncased), o modelo se torna mais robusto em cenários onde as variações de capitalização não influenciam o significado, garantindo maior consistência nos resultados.

D. Recuperação de Informação

A Recuperação de Informação (RI) é o processo de buscar, localizar e extrair informações relevantes em um grande conjunto de dados. Neste projeto, o objetivo da RI é encontrar referências bibliográficas que correspondam de forma precisa a uma referência fornecida, utilizando o modelo BERT para aprimorar a acurácia e a relevância dos resultados.

Todas as referências no banco de dados foram tokenizadas e convertidas em embeddings usando o modelo BERT. Esses embeddings foram então armazenados e vinculados à base de dados, de forma que cada referência esteja associada ao seu respectivo embedding.

E. Testes realizados

Três baterias de testes foram conduzidas com o objetivo de analisar como o modelo responde a diferentes tipos de perturbações aplicadas às referências utilizadas como entradas para realizar Recuperação de Informação. A partir desses testes, foram aplicadas métricas de desempenho para avaliar os resultados obtidos.

Os testes aplicados incluem:

1. **Remoção de Palavras**: Neste teste, palavras foram removidas gradualmente de forma aleatória da referência de entrada. A quantidade de palavras removidas foi aumentando progressivamente, começando com a remoção de uma palavra até atingir sete palavras.
2. **Remoção de Caracteres**: Similar ao teste anterior, mas com a remoção de caracteres aleatórios ao invés de palavras. O número de caracteres removidos também aumentou de forma incremental, de um até sete.
3. **Troca de Caracteres**: Neste teste, caracteres da referência foram substituídos por outros caracteres aleatórios escolhidos dentre as 26 letras do alfabeto. Assim como nos outros testes, as trocas começaram com um único caractere, aumentando até sete trocas.

Cada um desses testes visava avaliar a robustez do modelo frente a perturbações crescentes nas referências de entrada, permitindo uma análise detalhada de seu desempenho sob diferentes condições.

F. Métricas de avaliação

Para avaliar o desempenho da Recuperação de Informação (RI) em cada uma das três baterias de testes, foram utilizadas duas métricas principais. Primeiramente, em cada um dos sete testes de cada bateria, foi criado um gráfico que atribuía valores de 1 a 4, onde o valor 1 correspondia às RI que retornaram a referência correta na primeira posição entre os resultados; o valor 2 para aquelas que retornaram na segunda posição; 3 para as que retornaram na terceira posição; e 4 para as que não apareceram entre as três primeiras posições. Esses rankings foram estabelecidos com base na **similaridade cosseno**, uma métrica que mede a similaridade entre dois vetores, dada pela fórmula:

$$Similaridade\ Cosseno = \frac{A \cdot B}{\|A\| \|B\|}$$

onde A e B são os vetores dos embeddings das referências.

Além disso, foi criado um gráfico para cada bateria de testes, no qual o eixo x representava os diferentes testes realizados (por exemplo, remoção de uma palavra, remoção de duas palavras, etc.) e o eixo y apresentava o valor do **Mean Average Precision (MAP)** para cada teste. O MAP é uma métrica amplamente utilizada para avaliar a precisão em sistemas de RI, calculada como:

$$\text{MAP} = \frac{1}{N} \sum_{q=1}^{N} Average\ \ precision\ \left(\frac{1}{q}\right)$$

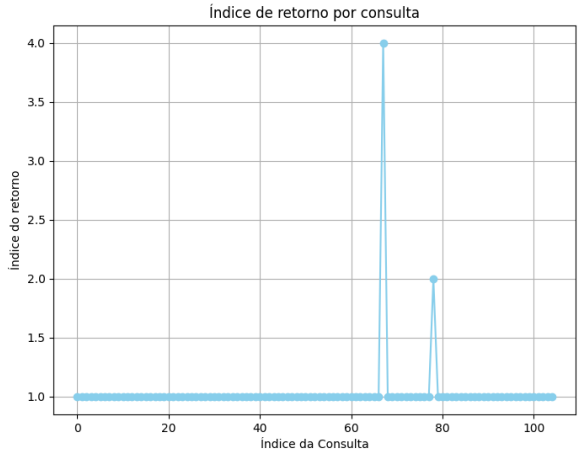
onde N é o número total de consultas e *Average precision* é a média das precisões obtidas nas diferentes posições relevantes para a consulta q.

V. Resultados Obtidos

Os resultados dos testes realizados foram consolidados em um total de 24 gráficos, distribuídos entre as três baterias de testes. Esses gráficos oferecem uma visualização detalhada do desempenho da Recuperação de Informação (RI) sob diferentes condições de perturbação das referências. Cada conjunto de gráficos apresenta, individualmente, o comportamento do modelo em termos de posicionamento das referências retornadas (valores de 1 a 4) e do valor do Mean Average Precision (MAP) ao longo das modificações incrementais aplicadas às entradas de teste. A análise desses gráficos permite observar padrões de desempenho, identificar a robustez do modelo frente a variações nas entradas, e compreender como cada tipo de perturbação impactou a eficácia da RI. Esses resultados fornecem uma base sólida para discussões sobre a eficácia do modelo BERT aplicado à tarefa de recuperação em contextos bibliográficos.

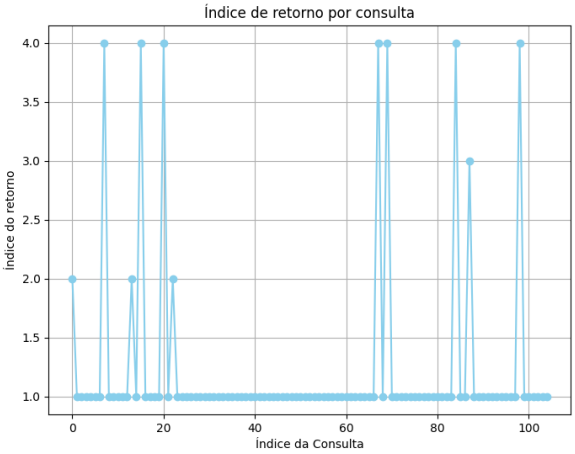
Na primeira bateria de testes, na qual eram realizadas a remoção de palavras da referência foram obtidos os seguintes resultados:

1. Remoção de uma palavra da referência:



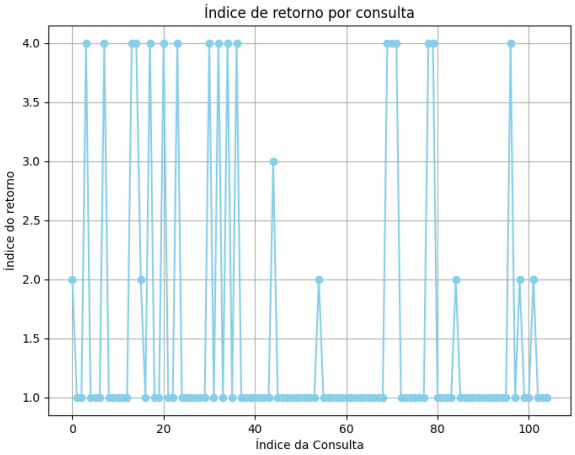
Com a remoção de uma palavra da referência, apenas uma consulta das 105 não obteve seu retorno correto dentre as três primeiras mais similares;

2. Remoção de duas palavras da referência:



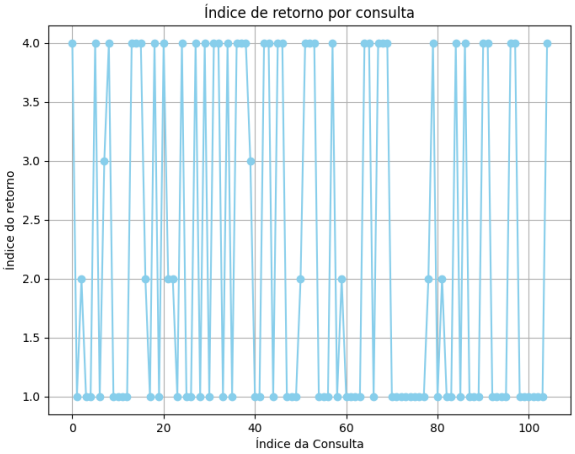
Com a remoção de duas palavras da referência, 98 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

3. Remoção de três palavras da referência:



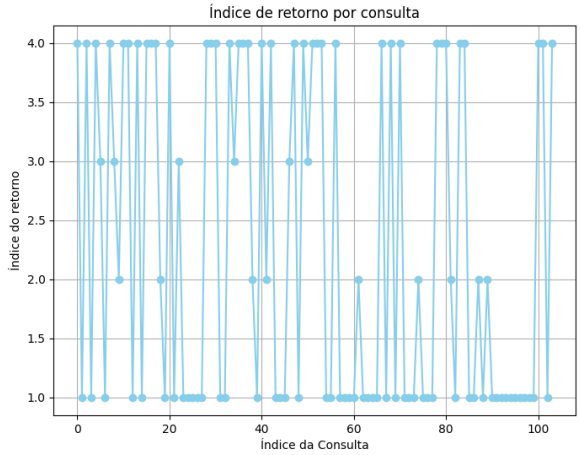
Com a remoção de três palavras da referência, 88 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

4. Remoção de quatro palavras da referência:



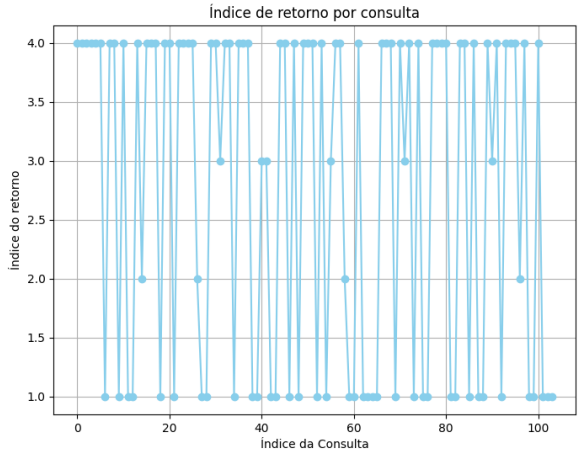
Com a remoção de quatro palavras da referência, 67 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

5. Remoção de cinco palavras da referência:



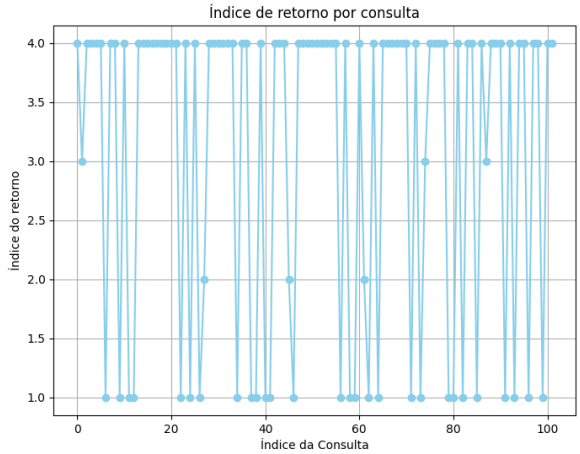
Com a remoção de cinco palavras da referência, 67 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

6. Remoção de seis palavras da referência:



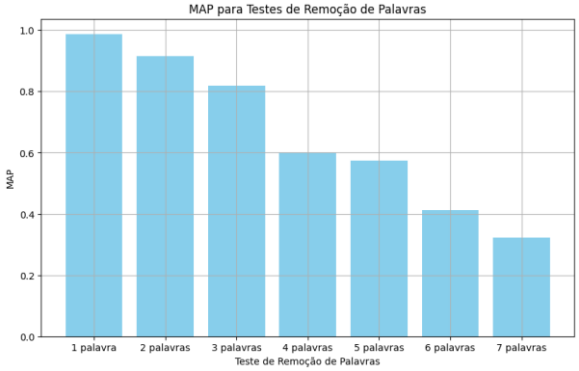
Com a remoção de seis palavras da referência, 48 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

7. Remoção de sete palavras da referência:



Com a remoção de sete palavras da referência, 34 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

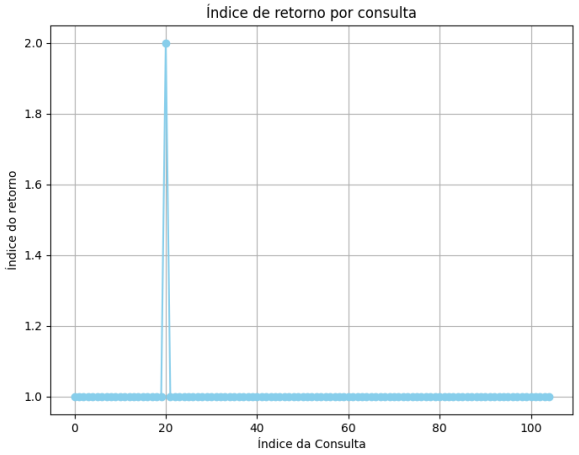
8. MAP para os testes de remoções de palavras:



Com o MAP de cada teste foi possível notar que a cada palavra a mais removida da referência o trabalho de encontrar sua correspondência correta se torna maior. Entretanto é notável também uma queda mais acentuada nos resultados obtidos com a redução de três palavras para a redução de quatro e que a partir da remoção de quatro palavras, o modelo BERT começa a retornar a referência correta cada vez mais de forma mais demorada, indicando que outras referências possuíam maior similaridade com a consulta dada.

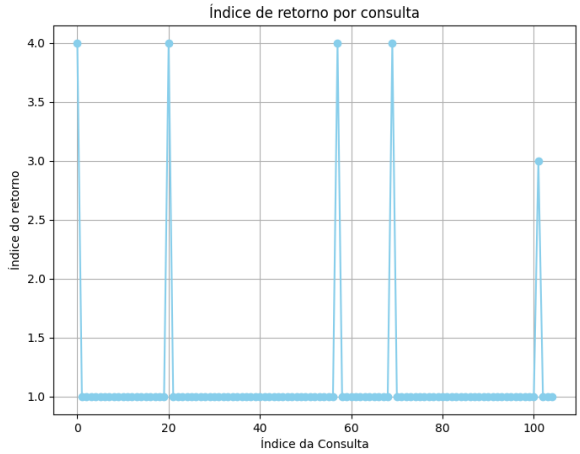
Na segunda bateria com a remoção de caracteres, foram obtidos os seguintes resultados:

1. Remoção de um caractere:



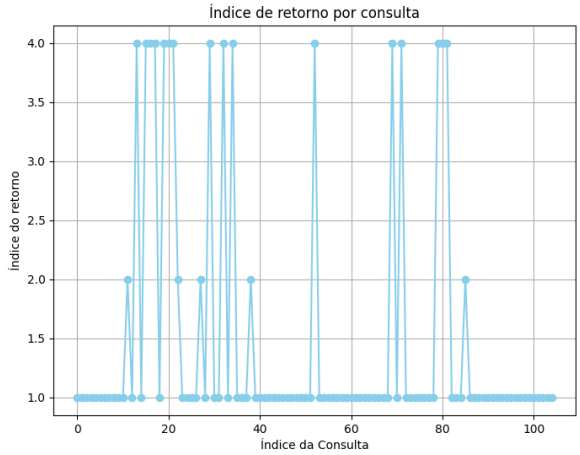
Com a remoção de um caractere da referência, todas as consultas das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

2. Remoção de dois caracteres:



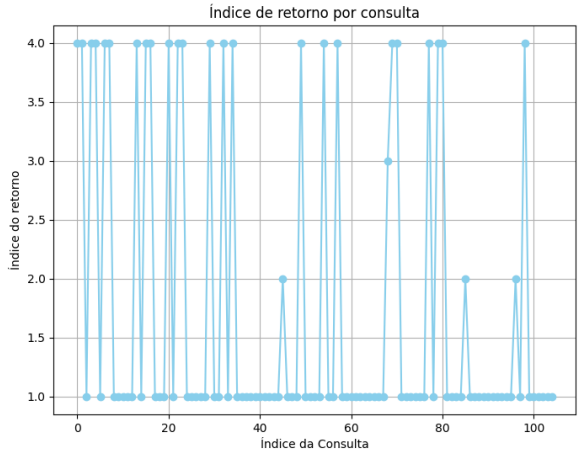
Com a remoção de dois caracteres da referência, 101 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

3. Remoção de três caracteres:



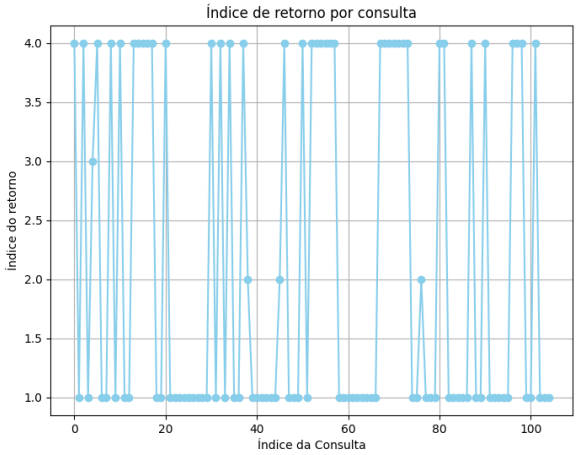
Com a remoção de três caracteres da referência, 89 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

4. Remoção de quatro caracteres:



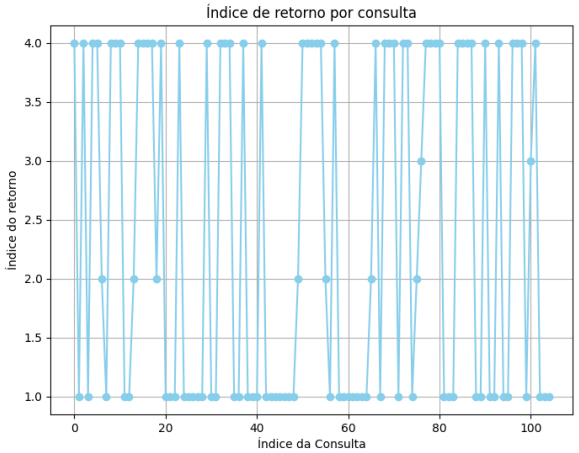
Com a remoção de quatro caracteres da referência, 81 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

5. Remoção de cinco caracteres:



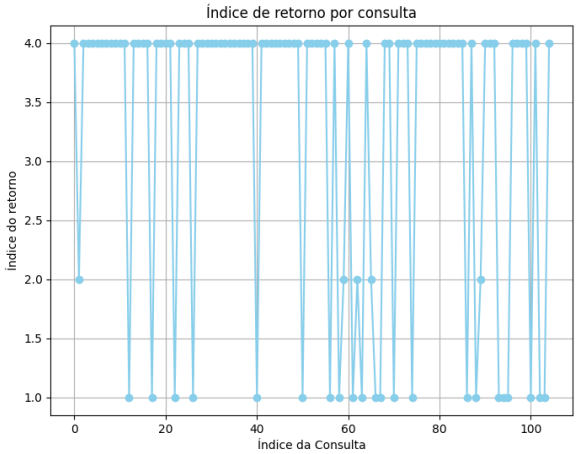
Com a remoção de cinco caracteres da referência, 67 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

6. Remoção de seis caracteres:



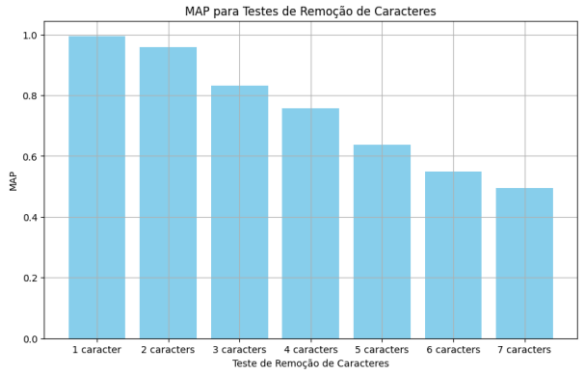
Com a remoção de seis caracteres da referência, 60 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

7. Remoção de sete caracteres:



Com a remoção de sete caracteres da referência, 27 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

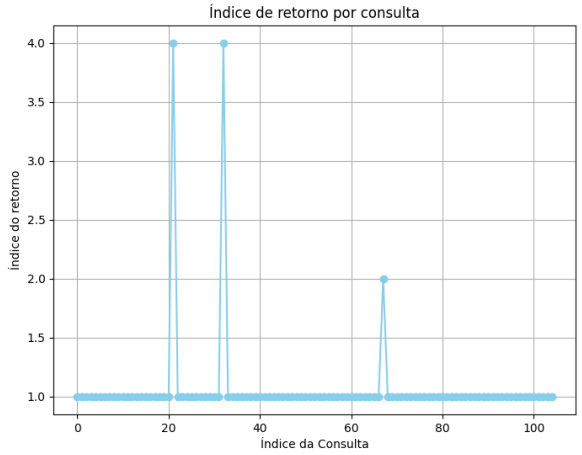
8. MAP para os testes de remoção de caracteres:



Com o MAP de cada teste foi possível notar que a cada caractere a mais removido da referência o trabalho de encontrar sua correspondência correta se torna maior. Não obstante, é notável uma queda mais acentuada nos resultados obtidos com a redução de dois caracteres para a redução de três e que até a remoção de cinco caracteres o modelo obteve um MAP maior ou igual a 0.6, um resultado melhor que a remoção de palavras.

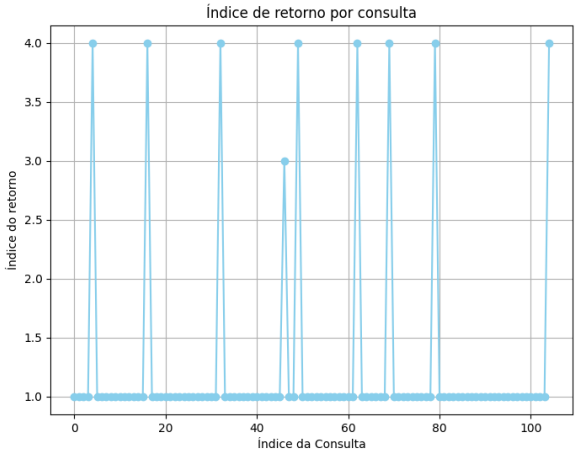
Por fim, na terceira bateria, realizando a troca de um caractere aleatório por outro, foram obtidos os seguintes resultados:

1. Troca de um caractere:



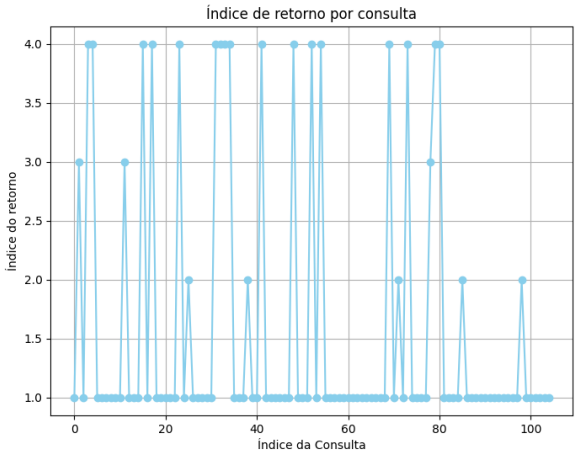
Com a troca de um caractere da referência, 103 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

2. Troca de dois caracteres:



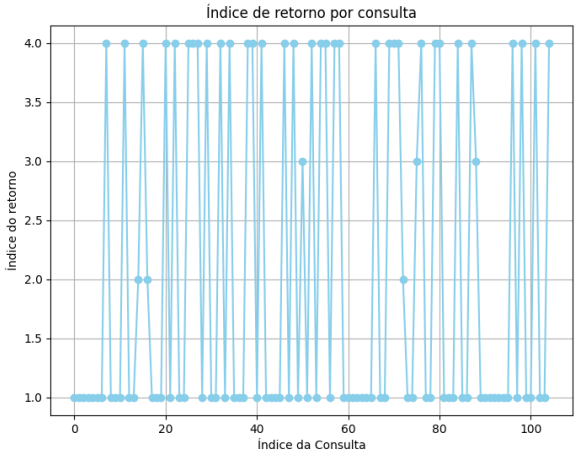
Com a troca de dois caracteres da referência, 97 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

3. Troca de três caracteres:



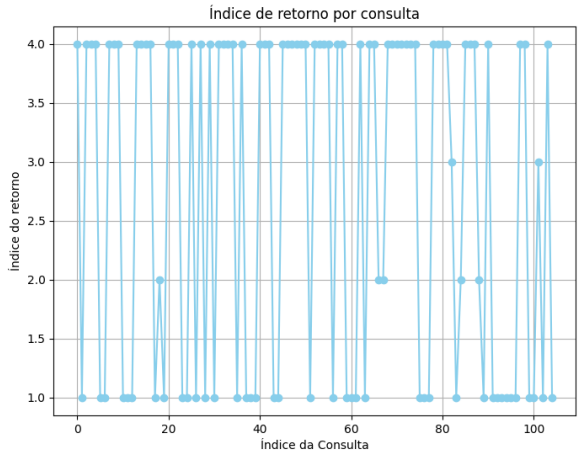
Com a troca de três caracteres da referência, 88 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

4. Troca de quatro caracteres:



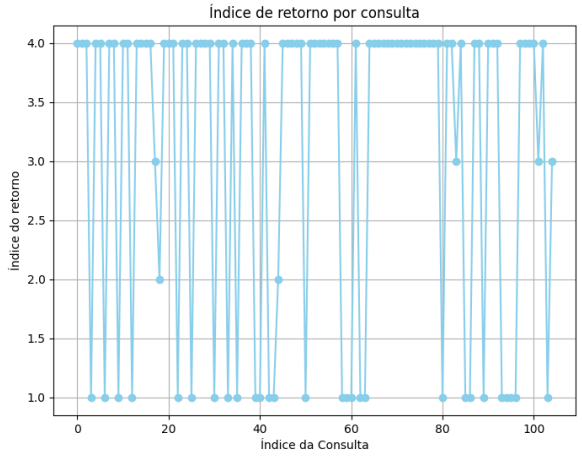
Com a troca de quatro caracteres da referência, 71 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

5. Troca de cinco caracteres:



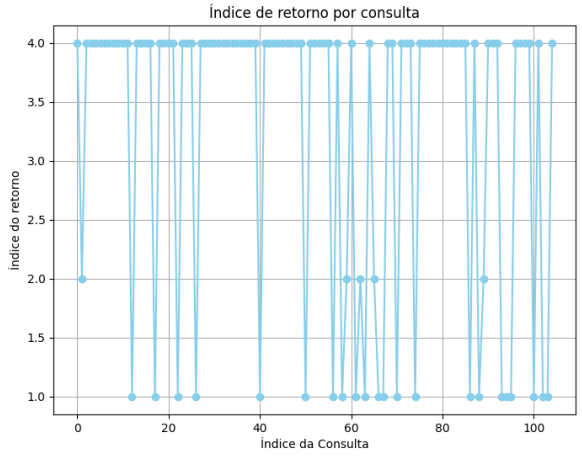
Com a troca de cinco caracteres da referência, 47 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

6. Troca de seis caracteres:



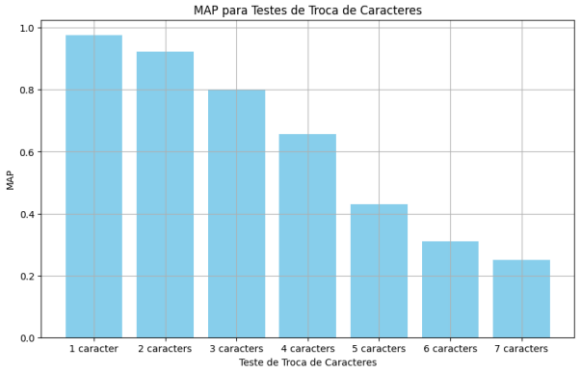
Com a troca de seis caracteres da referência, 34 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

7. Troca de sete caracteres:



Com a troca de sete caracteres da referência, 27 das 105 obtiveram seu retorno correto dentre as três primeiras mais similares;

8. MAP para os testes de troca de caracteres:



Como esperado a cada caractere a mais que era substituído, o valor de MAP diminui, com um decréscimo mais discrepante na troca de cinco caracteres.

Com base em todos os resultados apresentados, pode-se observar que, em todos os casos, a alteração de uma palavra ou caractere na referência afeta pouco o desempenho do modelo, que consegue recuperar a referência correta com facilidade. Mesmo com a alteração de duas palavras ou caracteres, o modelo mantém um desempenho robusto, com valores de MAP em torno de 0,9 ou mais, indicando uma alta precisão na recuperação das informações.

A partir da terceira alteração, os valores de MAP começam a cair, estabilizando-se em média na faixa de 0,8. Essa queda é mais acentuada nos casos de remoção de caracteres, sugerindo que esse tipo de perturbação tem um impacto maior na capacidade do modelo de identificar a referência correta. Em contraste, com quatro alterações, o teste de remoção de palavras apresentou a queda mais pronunciada, enquanto cinco alterações levaram à maior queda no desempenho para o teste de troca de caracteres.

Entre os testes, o que obteve os melhores resultados até o nível de sete alterações foi o de remoção de caracteres. Até a quarta modificação, o teste de troca de caracteres demonstrou melhor desempenho na recuperação de informações em comparação com a remoção de palavras; entretanto, após a quinta modificação, esse padrão se inverteu, com a remoção de palavras superando a troca de caracteres.

Estes resultados geram algumas observações interessantes, levando em conta que as consultas utilizadas possuíam em média 14 palavras, em relação a robustez do modelo BERT, que consegue um bom desempenho ao analisar pequenas perturbações, especialmente quando uma ou duas palavras ou caracteres são alterados. A queda do valor MAP com maior grau para remoção de caracteres a partir da terceira modificação pode indicar que a estrutura semântica da palavra começa a ser significativamente comprometida, afetando a capacidade do modelo de encontrar a correspondência correta. A substituição de um caractere por outro não possui tanto impacto com poucas

trocas quanto a remoção de palavras, todavia, ao começar a acumularem a quantidade de trocas, o modelo é mais afetado do que a remoção de palavras.

VI. Conclusão

Os resultados obtidos demonstram que o modelo BERT utilizado neste projeto exibe uma considerável robustez em face de pequenas perturbações nas referências de entrada, particularmente quando as alterações envolvem a modificação de uma ou duas palavras ou caracteres. No entanto, à medida que o nível de alterações aumenta, observa-se uma queda na precisão, especialmente em casos de remoção de caracteres. A análise das diferentes formas de perturbação revelou que a troca de caracteres inicialmente afeta menos o desempenho do modelo, mas se torna mais prejudicial com um número maior de alterações. Esses achados ressaltam a importância de considerar o tipo e a severidade das perturbações ao aplicar modelos de Recuperação de Informação em contextos práticos, e sugerem que, embora o BERT seja eficiente em contextos estáveis, ajustes adicionais podem ser necessários para garantir a precisão em cenários mais dinâmicos e variáveis, e para o futuro pode ser interessante utilizar o conjunto de teste alterado, para treinar o modelo BERT, e assim gerar um modelo novo com possivelmente maiores habilidades ao lidar com nuances na entrada.

VII. Referências Bibliográficas

ALVES, Maria Bernadete Martins; ARRUDA, Susana Margareth. Como fazer referências bibliográficas. Biblioteca Universitária. UFSC, Florianópolis, sd, 2008.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. Sistemas inteligentes-Fundamentos e aplicações, v. 1, n. 1, p. 32, 2003.

PADOVA, Ted. Adobe acrobat 9 PDF bible. John Wiley & Sons, 2008.

GONZALEZ, Marco; LIMA, Vera Lúcia Strube. Recuperação de informação e processamento da linguagem natural. In: XXIII Congresso da Sociedade Brasileira de Computação. sn, 2003. P. 347-395.

CASELI, Helena de Medeiros; NUNES, Maria das Graças Volpe; PAGANO, Adriana Silvina. O que é PLN?. Processamento de linguagem natural: conceitos, técnicas e aplicações em português, 2023.

FALCÃO, João Vitor Regis et al. Redes neurais deep learning com tensorflow. RE3C-Revista Eletrônica Científica de Ciência da Computação, v. 14, n. 1, 2019.

CARDOSO, Olinda Nogueira Paes. Recuperação de Informação. INFOCOMP Journal of Computer Science, v. 2, n. 1, p. 33-38, 2000.

KUMAR, Sanit et al. Web scraping using Python. International Journal of Advances in Engineering and Management, v. 4, n. 9, p. 235-237, 2022. https://sbert.net/index.html

GOLLAPALLI, S. D., Li, Z., & Mitra, P. (2011). CiteSeerX: A Scholarly Big Dataset for Literature Search and Beyond. ACM Transactions on Information Systems (TOIS).

WHITE, W. E., & Hernandez, P. (2016). Automated Reference Checking in Scholarly Manuscripts. Journal of Scholarly Publishing.