Roteiro para apresentação

Slide 1 - CAPA

O título do meu trabalho é “Inovação na Pesquisa de Jurisprudência Selecionada do TCU: Expansão de Documentos usando Modelos de Linguagem”.

Apesar do título parecer um pouco complicado, a ideia é simples: O TCU tem uma pesquisa de jurisprudência e a gente pretende melhorar a qualidade dela usando uma técnica que chama expansão de documentos, ou seja, temos o documento original e vamos inserir textos criados artificialmente dentro do documento. E a criação desses textos será feita usando modelos de linguagem, que é um nome genérico para modelos tipo o ChatGPT.

Slide 2 – ÍNDICE

Eu vou dividir a apresentação em 4 partes. Inicialmente apresento o problema que quero abordar, depois o método que usei para tratar o problema, os resultados e, por fim, concluir o trabalho.

Slides 3, 4 e 5 – PESQUISA INTEGRADA

Pra gente ter uma noção do problema a ser resolvido, precisamos primeiro entender um pouco sobre a pesquisa do TCU.

O Tribunal tem uma ferramenta de pesquisa online, a PESQUISA INTEGRADA. É uma interface única que permite pesquisar em diversos temas. Por exemplo, permite pesquisar dentro de Acórdãos, dentro de Processos, dentro do BTCU e por aí vai. Cada um desses temas é uma base de dados.

O funcionamento é simples. O usuário fornece um termo de pesquisa nessa caixa aqui e aí a Pesquisa Integrada consulta todas as bases de dados e dá o resultado da pesquisa em cada uma delas. O usuário entra num documento e vê detalhes do documento.

Slide 6 – ACÓRDÃO 878/2022-PL

O problema é que a ferramenta nem sempre retorna os documentos que o usuário espera.

Em uma recente fiscalização feita pelo próprio TCU em 2022, a antiga SEFTI identificou que tem alguns problemas no mecanismo de busca do órgão. Indicam, por exemplo, que há casos inclusive de usuários internos que preferem pesquisar pelo Google do que usando a ferramenta do portal.

É esse o problema que quero tratar.

Slides 7 e 8 – PROCESSO DE INDEXAÇÃO

Pra gente entender as possibilidades de intervenção na Pesquisa Integrada, é necessário primeiro analisarmos como funciona uma pesquisa textual.

Qualquer pesquisa pode ser dividida em dois processos, o de indexação e o de consulta.

A indexação é o processo em que você coleta os documentos, processa eles e guarda para que sejam consultados depois. Em linhas gerais, funciona assim:

1. Como entrada, você pode ter qualquer tipo de documento. Mesmo que a pesquisa seja textual, a origem do texto pode estar em um arquivo do Word ou em uma imagem ou em um vídeo.
2. A partir desses documentos, você extrai o texto deles e guarda uma versão do documento que pode ser retornada pro usuário. Isso pode ser algum texto processado ou o documento original. Ou as duas coisas.
3. O texto extraído é processado e depois guardado num banco de dados que chamamos de índice. Eu não sei a origem do nome, mas suspeito que seja alguma coisa referente a um índice remissivo.

No caso da Pesquisa Integrada, cada caixa desse esquema pode fazer algo diferente dependendo da base de dados. O que é comum a todas as bases da integrada é que é uma pesquisa léxica, ou seja, baseada no texto exato do documento. Usamos um índice invertido implementado no Solr ou Elastic. O resto depende da base. Tem base de dados que o documento de entrada é um word, outras é um pdf que pode passar por um OCR e por aí vai. O tratamento do texto depende da base. Algumas vezes é necessário limpar o texto (remover algumas partes, HTML etc).

Uma vez tratado, a gente tem um texto que será indexado. Para a inserção do índice invertido é feito um processo de tokenização, lematização e remoção de stopwords. Então na prática, o texto que é guardado é uma versão simplificada do texto extraído. Isso diminui a complexidade do problema.

Slides 9 e 10 – PROCESSO DE CONSULTA

Uma vez que o documento foi indexado, a pesquisa é feita no processo de consulta. A ideia é simples, basicamente o usuário acessa uma interface de pesquisa, que por sua vez acessa um índice e um modelo de ranqueamento. Dependendo da implementação, os resultados (ou alguma versão disso) podem ser guardados num log para avaliação.

No caso das pesquisas da Integrada, que são léxicas, a query passa por processamento similar ao dos documentos antes de ser pesquisada. Nesse exemplo, suponha que temos esse termo de pesquisa (técnica preço) e esse documento, que só tem a palavra preço, mas não tem a palavra técnica (e sim técnicos). Apesar disso, depois da lematização, ambos os termos são iguais.

Slides 11 e 12 – PROBLEMAS ABORDAGENS LÉXICAS

Existem alguns problemas com as pesquisas léxicas.

Um dos problemas é o descasamento de vocabulário. Nesse caso, como as consultas dependem do termo exato (depois de processado) pesquisado, se o usuário pesquisar por um termo que seja diferente, mas semanticamente próximo, o sistema não vai encontrar.

Para ilustrar isso, eu trouxe um exemplo em que a pesquisa é feita pelo termo sancionados e o documento foi encontrado. Em seguida, pesquisei pelo termo multados e, embora semanticamente próximos, o documento não foi encontrado.

Uma outra questão muito importante é que os melhores métodos de ranqueamento consideram no cálculo da relevância a frequência das palavras no texto. Assim, mesmo que uma palavra seja importante, se ela aparecer pouco no documento, ela contribuirá pouco para o score. Isso pode impactar na ordenação desejada de documentos. Esse caso é mais difícil de ver, por isso eu não trouxe um exemplo.

Slides 13, 14 e 15 – POSSÍVEIS SOLUÇÕES

Há 3 formas para melhorar o resultado de uma pesquisa léxica.

1. A primeira é a expansão de queries, que expande os termos de pesquisa inseridos pelo usuário para considerar novos termos. Isso pode ser feito usando um dicionário de sinônimos por exemplo. O método tem algumas vantagens e desvantagens:
   1. É muito fácil de testar, pois você precisa apenas alterar o processo de consulta
   2. Como estamos ainda no mundo léxico, você também não precisa alterar o motor de busca
   3. Tradicionalmente, gerava resultados melhores que a técnica de expansão de documentos (próximo slide)
   4. Como desvantagem, a gente insere um novo elemento na consulta, que pode ou não gerar impactos dependendo do caso
2. O segundo método é a expansão de documentos, que funciona adicionando um novo texto a cada documento da base. Enquanto a expansão de queries altera o processo de consulta, a expansão de documentos altera o processo de indexação. E, claro, também tem vantagens e desvantagens:
   1. Ele insere um novo elemento apenas na indexação. Ou seja, eventuais problemas que ocorram ficam restrito a esse momento e podem ser mais facilmente tratados
   2. Também não exige mudança do motor de busca
   3. Modelos mais recentes tem resultados melhores que a expansão de queries
   4. Como desvantagem, ele é mais difícil de testar, pois exige a reindexação de toda a base de dados
3. Por fim, a última alternativa é você sair do mundo léxico e ir pro mundo semântico. Potencialmente, é o método que tem os melhores resultados. Entretanto, também há desvantagens:
   1. Ele exige muita alteração em uma pesquisa léxica que já está funcionando, pois é necessário trocar o motor de busca
   2. Depende de GPU no processo de consulta, o que em alguns casos pode ser um problema
   3. Depende de um bom modelo de *embeddings*

Slide 16 – EXPANSÃO DOC NA JS – PORQUE EXPANSÃO DE DOC?

Considerando o que foi discutido e o problema identificado na pesquisa integrada, a ideia do trabalho é testar a técnica de expansão de documentos em apenas uma das bases de dados, a de JS.

A escolha da expansão de documentos é porque é a que menos altera a busca atual, e o faz apenas no momento da indexação.

Slide 17 – EXPANSÃO DOC NA JS – PORQUE JS?

E a ideia de estudar a jurisprudência selecionada e não todas as bases é por vários motivos.

1. Primeiro que é inviável, num contexto de trabalho de especialização, tratar diversas bases de dados ao mesmo tempo, então era preciso selecionar uma
2. Como estamos num contexto de trabalho relacionado a Controle Externo, era preciso restringir as bases disponíveis para esse contexto, e a base de Jurisprudência Selecionada atende a esse critério
3. Terceiro que a base possui um qrels, ou seja, um conjunto de queries padronizadas e de documentos esperados que podem ser usados para teste. Para verificar se as coisas estão melhorando ou piorando é necessário medir, e um qrels é a forma de se fazer isso.
4. Por fim, ela é uma base que não é muito grande nem muito pequena. E os documentos também não são gigantes. Ou seja, por não ser muito grande, a gente consegue fazer testes relativamente rápidos. E por não ser muito pequena, a gente consegue ver o efeito real do método, sem ficar mascarado – se estivéssemos usando uma base com cerca de 100 documentos como a de questões de ordem, a gente provavelmente teria resultados nas métricas com saltos bruscos.

Slide 18 – DOCUMENTOS DA JS

Esse slide ilustra o que é um documento da base de jurisprudência selecionada.

Basicamente cada documento é estruturado ao redor de um enunciado, que é sumário de uma decisão considerada importante por um órgão especializado do TCU, ou seja, ele retrata o entendimento contido em uma deliberação. O documento também possui um excerto da decisão de onde esse enunciado foi gerado e alguns outros campos.

Slide 19 – QRELS

Eu tinha comentado que pra gente testar, precisamos comparar com algo, o que é feito com um qrels.

No caso dessa base, o qrels foi elaborado por dois colegas, Borela e Leo, a quem eu agradeço por disponibilizarem esses dados.

O qrels que foi gerado contém 150 queries, divididas em 3 grupos. O primeiro grupo é de termos de consulta que foram muito utilizadas na pesquisa. E o segundo e terceiro grupo são de queries sintéticas. No caso, o pessoal identificou os enunciados mais acessados e geraram usando LLM queries para eles. O grupo 2 é de queries num formato de palavra-chave e o grupo 3 é formado por queries longas, em formato de perguntas.

Na prática, do ponto de vista atual o grupo 1 é o mais importante, pois representa efetivamente como os usuários pesquisam hoje.

Slide 20 – MÉTRICAS

Uma vez que temos o QRELS, o que vamos fazer é executar as queries e comparar o resultado obtido com o resultado esperado. Para isso, os resultados serão informados usando quatro métricas:

1. Precisão
2. Recall
3. MRR
4. nDCG

As duas primeiras são métricas que indicam o percentual de documentos relevantes retornados, a diferença é que a precisão o faz em relação ao total de documentos retornados e o recall em relação ao total de documentos relevantes.

As duas últimas são métricas que indicam o posicionamento dos documentos relevantes na lista. O MRR indica apenas a posição do primeiro elemento e o nDCG sintetiza a ordem de todos eles.

Todas as métricas aqui vão de 0 a 1, quanto mais próximo de 1, melhor.

Slide 21 – EXPERIMENTO – BASELINE

Dito isso, passamos agora para a modelagem dos experimentos.

A primeira coisa que precisamos fazer é definir *baselines*, cenários base para compararmos. Como a ideia é melhorar a pesquisa de JS, é natural que um cenário base seja exatamente a pesquisa atual.

No entanto, é comum em trabalhos na área de recuperação da informação sempre comprar os resultados com o BM25. Isso é feito porque o BM25 é um método que, apesar de antigo, tem bons resultados. Nesse trabalho, estou indexando a base considerando apenas os campos Enunciado e Excerto. Isso não implica que os outros campos da base não são importantes, foi apenas uma decisão de projeto porque é onde está o texto dessa base. Os outros campos são normalmente usados como filtros, o que não é o caso aqui.

Nesse ponto, vale a pena explicar um pouco também como é a pesquisa atual de jurisprudência. Em síntese, ela é uma pesquisa léxica em que é necessário ter todos os termos consultados presentes no documento, ou seja, usa o operador booleano AND. Além disso, ela dá pesos distintos para os diversos campos do documento.

A pesquisa difere do BM25 padrão pelo fato do BM25, por padrão, não dar pesos distintos para campos diferentes. Você agrega todo o texto num bolo só e tudo tem o mesmo peso. Além disso, você não precisa ter todos as palavras presentes no documento – é como se ele usasse o operador booleano OR entre os termos.

Slide 22, 23, 24 – EXPERIMENTO COM EXPANSÃO

Indo para os experimentos com expansão de documento, vamos testar algumas abordagens. Todas as abordagens propostas expandem o documento com um texto gerado por um modelo de linguagem. O texto é criado a partir do enunciado, pois é ele o sumário da decisão. E, no fim, o resultado do documento original (Enunciado e Excerto) é indexado num BM25 junto com o novo texto gerado.

O primeiro experimento com expansão foi é com o método docT5query. Basicamente ele usa o modelo de linguagem T5 e foi treinado para, dado um documento, gerar possíveis queries que podem ser usadas para pesquisar. O modelo de linguagem tenta adivinhar possíveis queries.

O segundo experimento é com a extração de sinônimos do enunciado. Nesse caso, estamos testando com o GPT e o Llama. É gerado um prompt pedindo as palavras mais importantes e pedindo sinônimos.

O terceiro grupo de experimentos é a reescrita do enunciado, que também é feita usando GPT e Llama.

Slides 25 e 26 – RESULTADOS – BASELINE

Chegamos aos resultados. Antes de tudo, no texto eu apresentei os resultados das métricas olhando os primeiros 5 e 20 resultados retornados. Como os resultados são semelhantes, vou focar nessa apresentação apenas nos primeiros 5 resultados retornados.

Primeiro, vamos ver os resultados apenas dos cenários base.

1. No grupo 1 de queries, percebemos que a configuração atual é cerca de 5% melhor do que o BM25 na precisão e no recall, ou seja, em geral ela traz mais documentos relevantes do que o BM25. Por outro lado, a ordenação oferecida pelo BM25 é melhor. Pela fórmula de cálculo, o MRR indica que, em regra, a configuração atual traz o primeiro documento relevante na terceira posição e o BM25, na segunda posição. Em relação a ordenação geral, o nDCG do BM25 é cerca de 6% maior do que a configuração atual.
2. Em outras palavras, o BM25, mesmo trazendo um pouco menos de documentos relevantes nas 5 primeiras posições, faz uma ordenação melhor do que o que é retornado.
3. Já em relação ao grupo 2 e 3, fica nítido a superioridade do BM25, especialmente em relação ao grupo 3. No grupo 2, que representa termos de pesquisa presentes no enunciado, trouxe mais documentos relevantes e manteve a ordenação mais ou menos com a mesma performance.
4. O grupo 3 é escancarado a superioridade do BM25. E isso ocorre por conta do operador AND usado na pesquisa atual.

Slide 27 – RESULTADOS – DOCT5QUERY

Tendo em vista os resultados dos slides anteriores, a partir de agora vou me ater mais a comparar os resultados com o BM25 do que com a pesquisa atual, principalmente porque o BM25 já ordena melhor os resultados e a ordenação dos documentos retornados é algo muito importante, principalmente por conta de questões relacionadas a experiência do usuário na ferramenta.

Agora vamos ver o que ocorre quando eu comparo os baselines com a expansão de documentos usando o método docT5query. Nesse cenário eu testei a expansão dos documentos com 1 e com 5 queries. Os resultados foram o seguinte:

1. No grupo 1 de queries, a expansão com apenas 1 query não teve muita diferença. Algumas métricas melhoraram, outras pioraram. Mas com 5 queries, tivemos uma melhora muito grande em todas as métricas, com 3 delas chegando na casa dos 40%. É um resultado muito bom, ainda mais considerando a facilidade que é de implementar isso.
2. Não vou mostrar mais os resultados para os grupos 2 e 3 de queries por questões de tempo e para simplificar a apresentação. Entretanto, em linhas gerais, eles seguem os resultados do grupo 1, mas com menos intensidade – justamente por serem de queries sintéticas geradas especificamente em cima do enunciado.

Slide 28 – RESULTADOS – SINONIMOS

Na geração de sinônimos das palavras mais importantes do enunciado, os testes foram feitos com 3 modelos: GPT-3.5, GPT-4o e Llama 3 70B. Podemos observar também melhorias consideráveis em relação em BM25, chegando a 15% dependendo da métrica.

Um resultado interessante é que o modelo maior (GPT-4o) não foi o que teve melhores resultados.

Os resultados para o grupo 2 e 3 de queries foram muito menos expressivos que para o grupo 1. No grupo 2, ficou da ordem de 5% de melhoria e, no grupo 3, chegou a até 2%, mas também teve resultados com ligeira piora.

Slide 29 – RESULTADOS – REESCRITA

Considerando que no teste anterior o GPT-4o não trouxe resultados significativos, optei por fazer a reescrita do enunciado só com o GPT-3.5 e o Llama 3 70B, até por uma questão de custo financeiro.

O resultado aqui foi bem ruim. Em geral trouxe menos documentos relevantes e piorou a classificação do primeiro documento relevante da lista.

Eu acho que cabe um *disclaimer* aqui em relação a isso. A gente está usando um qrels que tem dados de pesquisa reais no grupo 1 e dados sintéticos nos grupos 2 e 3. No grupo 1, as queries foram extraídas do log como queries frequentemente buscadas e elas possuem a característica de usar termos que efetivamente se encontram nos documentos, que são documentos especializados. Reescrever o documento alterando as palavras vai ter o efeito de inserir um pouco de ruído na base, o que justificaria esse resultado.

Esse *disclaimer* é importante pois implica que, mesmo não funcionando para esse conjunto de dados, dependendo da base pode ser interessante ter reescritas de um campo. Vai depender da forma como o usuário interage com a ferramenta.

Em certa medida, isso ocorreu também no cenário anterior de geração de sinônimos. Mas naquele cenário como também é reindexada a palavra original, ela acaba aumentando o seu peso no ranqueamento, compensando o ruído gerado.

Slide 30 – RESULTADOS – DOCT5QUERY + SINONIMOS

Por último, eu combinei a expansão dos dois melhores métodos, o docT5query e a geração de sinônimos. O efeito deles foi muito superior, com melhoria de quase 60% dependendo da métrica no grupo 1. Para os grupos 2 e 3 também houve melhorias, chegando a cerca de 15% no grupo 2 e a cerca de 5% no grupo 3.

Aqui, prevalece também os comentários em relação aos modelos usados na geração de sinônimos: o modelo maior não foi o que trouxe melhores resultados

Slide 31 – CONCLUSÕES

Concluindo o trabalho, dá pra gente sintetizar o que foi identificado e sugerir algumas recomendações:

1. O primeiro ponto é que a pesquisa atual é muito suscetível ao problema de descasamento de vocabulário, pois usa o AND como default. Então lá no grupo de queries 3 o resultado dela é muito ruim
   1. A sugestão então é alterar o operador default para OR e avaliar novamente a pesquisa
2. Dos métodos de expansão de documentos, tivemos bons resultados com o docT5query e a extração de sinônimos.
   1. Uma possibilidade de intervenção é a criação de campos para manter essas informações. E esses campos seriam usados apenas na busca para alterar o score
   2. Na extração de sinônimos, não precisamos nos preocupar com os custos do GPT-4o, podemos optar por um modelo mais barato (GPT-3.5) ou gratuito (Llama 3 70B).

Slide 32 – AGRADECIMENTOS