UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Geração Automática de Verbetações para Recuperação de Informações no Domínio Jurídico Brasileiro

Kenzo Miranda Sakiyama

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Kenzo Miranda Sakiyama

Geração Automática de Verbetações para Recuperação de Informações no Domínio Jurídico Brasileiro

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Roseli Aparecida Francelin Romero
Coorientador: Prof. Dr. Rodrigo Frassetto Nogueira

USP – São Carlos
Setembro de 2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Sakiyama, Kenzo Miranda
S158g Geração Automática de Verbetações para Recuperação de Informações no Domínio Jurídico Brasileiro / Kenzo Miranda Sakiyama; orientadora Roseli Aparecida Francelin Romero; coorientador Rodrigo Frassetto Nogueira. -- São Carlos, 2023.
107 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Processamento de Linguagem Natural. 2. Redes Neurais. 3. Recuperação da Informação. I. Romero, Roseli Aparecida Francelin, orient. II. Nogueira, Rodrigo Frassetto, coorient. III. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Kenzo Miranda Sakiyama

Automated Keyphrase Generation for Brazilian Legal Information Retrieval

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Roseli Aparecida Francelin Romero
Co-advisor: Prof. Dr. Rodrigo Frassetto Nogueira

USP – São Carlos
September 2023

AGRADECIMENTOS

Agradeço, em primeiro lugar, aos meus orientadores profa. Dra. Roseli A. F. Romero e prof. Dr. Rodrigo Nogueira. Em particular, gostaria de agradecer a profa. Dra. Roseli A. F. Romero pela confiança, acolhimento e atenção durante todo o período da proposta, além de me permitir interagir com outras áreas além da minha área de pesquisa como robótica. Agradeço também ao prof. Dr Rodrigo Nogueira, pelas discussões e sugestões técnicas apresentadas que permitiram guiar o desenvolvimento desta pesquisa sem complicações.

Em seguida, gostaria de agradecer aos meus colegas e amigos do Laboratório de Aprendizado de Robôs (LAR), que tornaram a minha jornada mais divertida e tranquila através do compartilhamento de experiências. Em particular, quero agradecer aos meus amigos Iury Batista Andrade Santos e Raphael Montanari pelos conselhos e acolhimento inicial, que me permitiram inserir à pós-graduação e à USP mais facilmente. Também gostaria de agradecer a minha família pelo suporte e confiança contínuo ao longo de toda minha graduação e pós-graduação.

Por fim, quero agradecer à CAPES, FAPESP, CeMEAI e ao ICMC pelo apoio técnico e financeiro, e todos os seus funcionários pela atenção dada em todo o período de estudo.

RESUMO

SAKIYAMA, K. M. **Geração Automática de Verbetações para Recuperação de Informações no Domínio Jurídico Brasileiro**. 2023. 107 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

A grande quantidade de processos jurídicos em trânsito no Brasil, evidencia a grande lentidão do sistema judiciário brasileiro. Sendo assim, há uma grande necessidade em desenvolver formas de automatizar e melhorar processos existentes. Os recentes avanços em Processamento de Linguagem Natural (PLN), possibilitam a aplicação dos métodos do estado da arte para automatizar tarefas em diferentes domínios. Assim, neste trabalho, abordamos o problema da automatização da escrita de verbetações: sequência de termos-chave presentes em documentos utilizados em tribunais de todo o Brasil. Para tanto, propusemos a utilização de um *framework* texto-para-texto baseado em *Transformers* generativos. Avaliamos diferentes modelos generativos (PTT5, mT5, OPT e BLOOM) e comparamos seus desempenhos para a tarefa proposta. O modelo PTT5 foi escolhido como gerador de verbetações, pois alcançou uma pontuação BLEU de 37,54% no conjunto de teste, superando os demais modelos avaliados em até 24,6%. Por fim, para avaliar a influência e a qualidade das verbetações geradas, realizamos quatro experimentos baseados em um caso real de recuperação de informações no domínio jurídico. Empregando métodos tradicionais de recuperação de informações (TF-IDF e BM25); em combinação com as verbetações originais, geradas, ou ambas; observamos ganhos estatisticamente significativos (p-valor $< 0{,}05$) em todos os experimentos realizados.

Palavras-chave: Aprendizado de Máquina, Aprendizado Profundo, Processamento de Linguagem Natural, Geração de Texto, Recuperação de Informações..

ABSTRACT

SAKIYAMA, K. M. **Automated Keyphrase Generation for Brazilian Legal Information Retrieval**. 2023. 107 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2023.

The large quantity of legal processes in transit in Brazil highlights the slowness of the Brazilian judicial system. Therefore, there is a great need to develop ways to automate and improve existing processes. The recent advancements in Natural Language Processing (NLP) enable the application of state-of-the-art methods to automate tasks in different domains. Thus, in this work, we address the problem of automating the writing of keyphrases: a sequence of key terms present in documents used in courts throughout Brazil. For this, we proposed the use of a text-to-text framework based on generative Transformers. We evaluated several generative models (PTT5, mT5, OPT, and BLOOM) and compared their performances for the proposed task. PTT5 was chosen as the keyphrase generator, as it achieved a BLEU score of 37.54% on the test set, outperforming the other evaluated models by up to 24.6%. Finally, to assess the influence and quality of the generated keyphrases, we performed four experiments based on a real case of information retrieval in the legal domain. By using traditional information retrieval methods (TF-IDF and BM25); in combination with the original, generated keyphrases, or both; we observed statistically significant gains (p-value < 0.05) in all experiments.

Keywords: Machine Learning, Deep Learning, Natural Language Processing, Text Generation, Information Retrieval..

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AB2L	Associação Brasileira de *Lawtechs* e *Legaltechs*
AM	Aprendizado de Máquina
BERT	Modelo codificador *Bidirectional Encoder Representations from Transformers*
BLEU	Métrica para avaliação de geração textual *Bilingual Evaluation Understudy*
BLOOM	Modelo gerador de texto *BigScience Large Open-science Open-access Multilingual Language Model*
BM25	Método de ranqueamento tradicional, baseado em representações *bag of words*
CNJ	Conselho Nacional de Justiça
DL	*Deep Learning*
FLOPs	*Floating-point Operations Per Second*
GPT	Modelo gerador de texto *Generative Pre-Trained Transformer*
MRR	Métrica para avaliação de ranqueamento *Mean Reciprocal Rank*
nDCG	Métrica para avaliação de ranqueamento *Normalized Discounted Cumulative Gain*
OPT	Modelo gerador de texto *Open Pre-Trained Transformers*
PLN	Processamento de Linguagem Natural
RI	Recuperação de Informações
ROUGE	Métrica para avaliação de geração textual *Recall-Oriented Understudy for Gisting Evaluation*
STF	Supremo Tribunal Federal
STJ	Supremo Tribunal de Justiça
T5	Modelo gerador de texto codificador-decodificador baseado em *Transformers*
TF-IDF	Método de representação textual *Term Frequency - Inverse Term Frequency*, baseado em representações *bag of words*
TREC	*Text Retrieval Conference*

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

1.1 Introdução

O processo de Recuperação de Informações (RI) pode ser definido como a busca por informações em grandes conjuntos de dados não-estruturados (frequentemente textos), para satisfazer uma consulta de um usuário. Sendo assim, os processos de RI podem ser encontrados em diferentes domínios. No Brasil, os sistemas de recuperação são frequentemente usados em sistemas de busca profissional utilizados, por exemplo, por profissionais do direito (CANEDO *et al.*, 2021). Nesse cenário, um profissional, com conhecimento do domínio, utiliza ferramentas de busca textual para atender suas necessidades de informação e assim realizar suas tarefas.

O uso de sistemas como esses torna-se necessário devido à grande abundância de documentos para análise jurídica no Brasil. Segundo dados de 2022, disponibilizados pelo Conselho Nacional de Justiça (CNJ) (CNJ, 23), ao final de 2021 haviam 77,3 milhões de processos em trânsito no judiciário brasileiro. Os dados representaram um aumento de 10,4% em relação ao ano anterior. Além disto, o relatório disponibilizado também reporta que o tempo médio de duração dos processos judiciais está entre meses e, em casos extremos, anos. Ainda sobre tais ações judiciais, é importante ressaltar que todas demandam custos financeiros tanto para as partes envolvidas, quanto para o sistema judiciário brasileiro ao longo de toda sua duração. Desta forma, a fim de aprimorar os sistemas existentes e automatizar os processos, muito esforço têm sido direcionado ao campo jurídico no Brasil.

Em seu trabalho de 2021, Canedo *et al.* (CANEDO *et al.*, 2021) apresentou um estudo no qual foram entrevistados 107 órgãos brasileiros pertencentes a diferentes setores (executivo, legislativo, judiciário, etc). Seu estudo foi realizado com o intuito de investigar os sistemas de busca por jurisprudências utilizados nos órgãos públicos brasileiros. Jurisprudências consistem em conjuntos de decisões e aplicações da lei, realizadas por tribunais brasileiros. Os autores apontaram um grande uso de técnicas de Aprendizado de Máquina (AM) e Processamento de

Linguagem Natural (PLN) para análise de documentos, incluindo: vetorização, classificação, agrupamento de documentos e estimativa de similaridade. Outra constatação é que 69% dos órgãos entrevistados utilizam sistemas de busca baseados em operadores lógicos (booleanos) para realizar a recuperação de jurisprudência.

As descobertas de Canedo *et al.* fomentam os resultados de Maia e Bezerra (MAIA; BEZERRA, 2020). Em seu trabalho, os autores identificaram um crescimento médio anual de 18,92% na produção científica relacionada à área de jurimetria entre 2002 e 2019 no Brasil. A jurimetria, por sua vez, busca aplicar uma análise quantitativa em processos de Direito, auxiliando na análise do grande volume de documentos existentes através da aplicação de técnicas e modelos estatísticos. Desta forma, nota-se que a jurimetria, além de relacionar-se à aplicações de AM e ciência de dados, também está relacionada a busca textual (subtarefa de RI), visto que a mesma requer filtragens iniciais para delimitar o escopo das análises textuais a serem feitas.

Outra evidência do grande interesse pela integração de processos de inteligência artificial ao Direito é o destaque da Associação Brasileira de *Lawtechs* e *Legaltechs* (AB2L). Fundada 2017, a associação reúne atualmente (em 2023) mais de 600 empresas que prestam serviços de análise, recuperação e compilação de dados legais e jurimetria [1]. Tais empresas focam em apresentar soluções para seu público de interesse, buscando oferecer ferramentas de busca e análise de documentos jurídicos que atendam às necessidades de profissionais de Direito.

Paralelo ao cenário discutido, entre 2017 e 2023, houveram grandes avanços no estado da arte de PLN promovendo grande interesse em transformar pesquisas na área em produtos como o *ChatGPT* [2]. Desde seu surgimento em 2017, a arquitetura *Transformer* (VASWANI *et al.*, 2017) vem predominando no estado da arte em diferentes tarefas de PLN, substituindo as tradicionais redes neurais recorrentes. Além de conseguirem processar sequências de texto mais eficientemente e paralelizável do que redes neurais recorrentes, modelos baseados em *Transformers* conseguem extrair representações textuais com maior riqueza semântica, dado que consideram um contexto maior e bidirecional em que as palavras se encontram para gerar suas representações.

Em 2020 foram apresentados modelos de linguagem baseados em *Transformers* (SOUZA; NOGUEIRA; LOTUFO, 2020; CARMO *et al.*, 2020), porém ajustados para o idioma português brasileiro. Conforme apontado pelos trabalhos, modelos ajustados para o idioma-alvo tendem a apresentar desempenhos superiores a versões multi-linguais dos mesmos em tarefas no idioma desejado. Desta forma, tais trabalhos tornam possível atingir resultados compatíveis com o estado da arte em diferentes tarefas de PLN em português brasileiro, devido à grande capacidade de transferência de aprendizado dos modelos apresentados.

Além disso, os modelos mencionados encontram-se disponibilizados publicamente[3]

[1] <https://ab2l.org.br/ecossistema/radar-de-lawtechs-e-legaltechs/>
[2] <https://openai.com/blog/chatgpt>
[3] <https://huggingface.co/models>

para desenvolvimento de pesquisa e aplicações. Sendo assim, existem muitas possibilidades e oportunidades para o desenvolvimento de aplicações que empregam o estado da arte de PLN, visando automatizar tarefas em diferentes domínios no idioma de interesse.

Tendo em mente as necessidades de automatização de processos jurídicos e as oportunidades apresentadas pelos modelos do estado da arte de PLN, a seguir será apresentada proposta deste projeto de Mestrado.

1.2 Apresentação da Proposta

Tendo apresentado o contexto em que este projeto se encontra, esta Seção visa apresentar a pesquisa desenvolvida. Em linhas gerais, a proposta se encaixa em dois contextos: automatização de processos no meio jurídico e busca jurisprudencial. Sendo assim, formulamos a seguinte questão de pesquisa: "O uso de técnicas avançadas de PLN permitiria a automatização da escrita de textos jurídicos em Português Brasileiro, de forma a gerar texto similar ao de especialistas do domínio?"

Partindo desta questão, este trabalho consiste em propor uma solução para automatização da escrita (geração) de um importante campo textual observado na estrutura de documentos legislativos em português brasileiro: as verbetações.

Figura 1 – Exemplo de ementa. A verbetação está destacada em negrito.

DIREITO CONSTITUCIONAL, TRIBUTÁRIO, PREVIDENCIÁRIO E PROCESSUAL CIVIL. CONTRIBUIÇÃO SOCIAL. ART. 2º DA LEI Nº 7.856, DE 25 DE OUTUBRO DE 1989. MAJORAÇÃO DE 8% PARA 10%, DA ALÍQUOTA DA CONTRIBUIÇÃO SOCIAL. SUCUMBÊNCIA. HONORÁRIOS ADVOCATÍCIOS. CUSTAS PROCESSUAIS.
1. Firmou-se em Plenário do Supremo Tribunal Federal o entendimento no sentido de que, em se tratando de "lei de conversão da Medida Provisória nº 86, de 25 de setembro de 1989, da data da edição desta é que flui o prazo de noventa dias previsto no art. 195, § 6º, da CF, o qual, no caso, teve por termo final o dia 24 de dezembro do mesmo ano, possibilitando o cálculo do tributo, pela nova alíquota, sobre o lucro da recorrente, apurado no balanço do próprio exercício de 1989"(RE 197.790-6-MG, Rel. Min. ILMAR GALVÃO).
2. Adotados os fundamentos desse precedente, o RE, no caso, é conhecido e provido.
3. As autoras ficaram vencedoras, quanto a outros pedidos, nas instâncias ordinárias, sem que o R.E. abordasse esses pontos. Sendo maior a sucumbência da ré, pagará honorários advocatícios às autoras.
4. Custas em proporção.

Os documentos os quais estamos interessados em analisar consistem ementas de processos legislativos do Brasil. As ementas consistem resumos de processos judiciais e são utilizadas por tribunais de todo o Brasil, visando fornecer uma representação concisa de decisões judiciais.

A Figura 1 apresenta um exemplo de ementa vinculada a um processo civil. Observando o exemplo, é possível observar que as ementas tendem a seguir uma estrutura bem definida:

1. **Verbetação (Palavras-chave):** sequência de termos em maiúsculo que buscam apresentar as palavras-chave da decisão;
2. **Parágrafos Enumerados:** parágrafos que descrevem as teses da decisão. Frequentemente o último parágrafo apresenta a decisão do tribunal.

Ao escrever verbetações, um especialista deve especificar termos-chave visando fornecer uma visão geral dos conteúdos mais importantes do processo jurídico em análise. O objetivo das verbetações é auxiliar em tarefas de busca e recuperação de jurisprudências (GUIMARÃES; SANTOS, 2016). Assim, por conta da forma e do estilo linguístico empregado, nota-se que a escrita de verbetações se assemelha às tarefas de extração de palavras-chave e de sumarização.

No entanto, embora semelhantes às palavras-chave, as verbetações podem conter frases pequenas. Além disto, por consistirem apenas de termos-chave, as verbetações tem um estilo linguístico único e diferem também de resumos escritos em linguagem natural e fluida. Um último aspecto que torna a tarefa de escrita de verbetações mais desafiadora é o fato de a maioria dos termos presentes nas verbetações não estarem necessariamente presentes diretamente no corpo do resumo. Assim, sua escrita requer certo grau de conhecimento do sistema judiciário brasileiro, já que não é uma tarefa puramente extrativa.

Dados os componentes de uma ementa apresentados anteriormente, e a questão de pesquisa de interesse, buscamos evidências para validar a seguinte hipótese: “O uso de uma abordagem de aprendizado supervisionado, baseada no estado da arte de PLN, viabilizaria a geração automática de verbetações?”. Para a investigação de tal hipótese, investigamos o uso de uma abordagem supervisionada de aprendizado profundo para gerar verbetações automaticamente. Ao extrair as verbetações e parágrafos das ementas, nossa proposta consistiu em preparar pares de entrada-saída (parágrafos enumerados e verbetações) para alimentar um gerador de texto supervisionado e baseado em *Transformers*.

Sendo assim, a principal contribuição desta proposta está no estudo da possibilidade de automatizar a escrita de verbetações utilizando aprendizado supervisionado. Detalhamos e avaliamos diferentes estratégias para a geração das verbetações, de forma que os resultados apresentados possam ser reproduzidos por diferentes tribunais (a depender da disponibilidade de documentos), e possam ajudar a automatizar tarefas no domínio jurídico brasileiro.

1.3 Objetivos

Após apresentarmos a contextualização necessária e a proposta, nesta seção são apresentados objetivo geral e os específicos desta Dissertação de Mestrado.

1.3.1 Objetivo Geral

Recapitulando os pontos apresentados na Seção anterior, nesta proposta, o interesse principal está na investigação da viabilidade da automatização da escrita de verbetações de processos jurídicos brasileiros através do uso de aprendizado profundo. Em específico, dada a tarefa de interesse, investigamos a possibilidade de utilizar o estado da arte de PLN (*Transformers*) na resolução da mesma.

Com base nos sucessos observados por modelos baseados em *Transformers* em tarefas de geração textual e na disponibilidade de modelos ajustados para português, espera-se que os mesmos também sejam capazes de gerar verbetações de qualidade similar às geradas por humanos. Desta forma, apresentados os resultados, os modelos produzidos podem ser integrados a sistemas existentes, de forma a auxiliar o processo de escrita de parte das ementas judiciais e melhorar ferramentas de busca jurisprudencial existentes (em virtude do papel das verbetações), atuando na aceleração de análises jurídicas em todo o Brasil. Vale destacar que este trabalho traz uma contribuição inédita no cenário brasileiro e pode ser facilmente replicada por diferentes tribunais do Brasil a depender da disponibilidade de documentos.

1.3.2 Objetivos Específicos

Os objetivos específicos são listados a seguir.

1. Investigação da influência do uso de verbetações em tarefas de recuperação de informação envolvendo ementas judiciais;
2. Apresentação de uma proposta inovadora e facilmente replicável para automatizar a geração de verbetações para documentos legislativos em português Brasileiro utilizando *Transformers*;
3. Avaliação de quatro modelos baseados em *Transformers* geradores de texto para a tarefa proposta;
4. Investigação de duas técnicas de decodificação para geração de verbetações;
5. Avaliação das verbetações geradas empregando métricas de geração de texto e tarefas de RI baseadas em problemas reais.

1.4 Organização do Documento

Este trabalho está organizado do seguinte modo. No Capítulo 2, conceitos de RI e de geração textual são introduzidos pois serão referenciados ao longo do trabalho. No Capítulo 3, são apresentados exemplos de trabalhos relacionados aos temas de interesse desta pesquisa. Em seguida, a metodologia aplicada para realização deste trabalho é apresentada no Capítulo 4. O

Capítulo 5 apresenta os resultados obtidos das avaliações das verbetações geradas pelos modelos utilizados. Finalmente, o Capítulo 6 apresenta as conclusões finais e os trabalhos futuros.

CAPÍTULO

2

FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo, algumas definições necessárias para o entendimento deste trabalho são apresentadas. Para isto, este capítulo está organizado em duas Seções, voltadas às duas grandes áreas abordadas nesta Dissertação: RI e *Transformers* . A Seção 2.1 apresenta conceitos relativos a área de Recuperação de Informações (RI). Em seguida, a contextualização de modelos *Transformers* para tarefa de geração de texto é apresentada na Seção 2.2.

2.1 Recuperação de Informações

Nesta Seção, serão discutidos conceitos introdutórios e necessários para a realização de tarefas de RI. As Subseções 2.1.1, 2.1.2, 2.1.3 e 2.1.4 apresentam conceitos necessários para a execução e avaliação de processos de RI. Ao fim, a Subseção 2.1.5 apresenta exemplos de métodos de RI.

2.1.1 Ranqueamento Textual

O problema de recuperação *ad-hoc* de documentos (ou ranqueamento textual) pode ser formalizado da seguinte forma: dado um *corpus* de documentos textuais $D = \{d_0, d_1, ..., d_n\}$ e uma consulta Q formulada em linguagem natural, o objetivo é retornar um subconjunto de D que satisfaça a consulta Q, baseado em um critério escolhido. O critério é utilizado para ordenar os documentos associados à consulta Q através da geração de pontuações (*scores*), as quais são associadas a cada documento. Exemplos de critérios são: similaridade semântica entre d e Q e relevância estimada.

Em outras palavras, consultas consistem em descrições de necessidades de informação de um usuário. Desta forma, focando na tarefa de ranqueamento textual, um sistema de recuperação de informações deve iterar sobre um conjunto de documentos buscando pelos textos que satisfaçam as necessidades especificadas do usuário. É importante ressaltar que tais coleções de

documentos textuais podem assumir diferentes tamanhos. Assim, sistemas de RI devem buscar escalabilidade, de maneira a ser possível aplicá-los tanto em pequenas coleções de documentos de empresas até grandes conjuntos de páginas *web*.

Voltando a atenção ao texto dos documentos em si, os mesmos podem assumir diferentes tamanhos. Frases, parágrafos, páginas, até livros completos podem ser utilizados para construir corpus para busca textual. Portanto, os algoritmos que constituem sistemas de RI para busca textual devem estar preparados para trabalhar com documentos dos diferentes tamanhos mencionados, ou permitir que os documentos possam ser divididos em segmentos menores (frases por exemplo) (YILMAZ *et al.*, 2019).

As definições apresentadas podem ser facilmente associadas a motores de busca populares (Ex: *Google* e *Bing*). Entretanto, a tarefa de ranqueamento textual pode ser aplicada a cenários diferentes. Considerando um caso de uso real de um profissional de advocacia, os documentos podem consistir nas ementas dos processos jurídicos e as consultas poderiam descrever os temas (ou tópicos) que o advogado deseja encontrar nos documentos recuperados. Este será o cenário de recuperação de informações a ser investigado por esta proposta.

2.1.2 O Conceito de Relevância

Até este ponto, foram feitas múltiplas menções à "relevância" de um documento porém sem apresentar uma definição para a mesma. Relevância é um conceito utilizado para quantificar o quanto um documento atende à necessidade de informação especificada por um usuário. Geralmente definida de forma binária, a relevância especifica se um determinado documento atende ou não a necessidade de informação descrita pelo usuário. Sendo assim, uma modelagem comum empregada por algoritmos de ranqueamento baseados em aprendizado de máquina é tratar a tarefa como um problema de classificação binário (LIN; NOGUEIRA; YATES, 2021). Entretanto, a popularidade desta definição não exclui a possibilidade de caracterizar a relevância utilizando escalas ordinais (ex: "não relevante", "relevante" e "muito relevante").

No caso binário, por se tratar de apenas duas possibilidades ("sim" ou "não") pode parecer que o processo de anotação de conjuntos de dados para a tarefa de ranqueamento de documentos seja simples. Entretanto, pessoas tendem a buscar por características diferentes para atender suas necessidades. O que é relevante para um anotador pode não ser para outra. Logo, o conceito de relevância tende a ser subjetivo. Assim, um anotador acaba inevitavelmente introduzindo seus vieses em sua anotação (a sua própria definição de "relevância"). Este fato acaba implicando em uma taxa de consenso baixa ao gerar julgamentos de relevância para a tarefa em análise (HARMAN, 2011).

As anotações de relevância (também chamadas de julgamentos de relevância) são comumente especificadas através de triplas (q, d, r) chamadas *qrels*. Para um exemplo, a anotação r indica se o documento d é relevante ou não para a consulta q. Como frequentemente utiliza-se a

definição binária de relevância, o *qrels* podem conter apenas triplas contendo documentos que sejam "relevantes" para a consulta especificada.

Conjuntos de dados anotados são necessários para a etapa de aprendizado de algoritmos de aprendizado de máquina supervisionados. Os *qrels*, por sua vez, cumprem este papel para algoritmos de ranqueamento supervisionados, sendo utilizados tanto para treinamento como avaliação dos ranqueadores. Assim, as anotações de relevância são utilizadas para treinar poderosos ranqueadores baseados em redes neurais e computar as métricas utilizadas para avaliar e comparar o desempenho de ranqueadores. Exemplos de métricas serão apresentadas na Subseção 2.1.4.

2.1.3 Anotações de Relevância

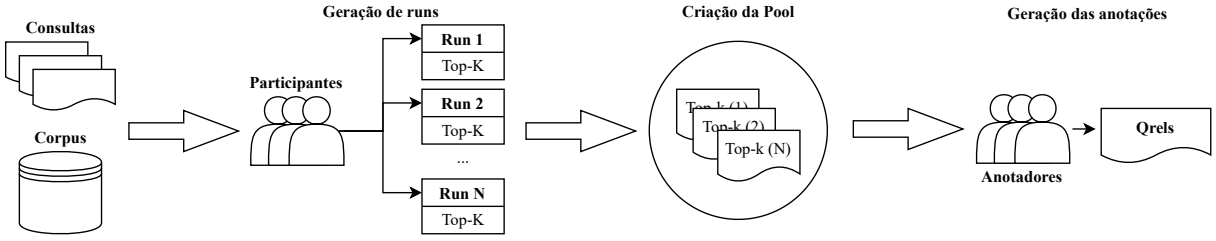


Figura 2 – Visão geral da metodologia de anotação TREC.

Com base no que foi apresentado sobre a definição e os usos das *qrels*, resta abordar como gerar tais anotações. Um exemplo de metodologia que pode ser aplicada para a anotação de *qrels* é a aplicada na *Text Retrieval Conference* (TREC)[1]. Realizada anualmente, esta conferência tem como objetivos gerais a construção de corpus para tarefas de RI de diferentes áreas de conhecimentos, promover e incentivar pesquisas em RI e aproximar pesquisas acadêmicas e aplicações no mundo real na área de estudo. A Figura 2 apresenta uma visão geral da metodologia de anotação. Seus componentes serão discutidos a seguir.

A metodologia TREC para anotação (SANDERSON, 2010) sugere a participação conjunta de múltiplos pesquisadores para a geração das *qrels*. Partindo das definições de consultas Q e conjuntos de documentos D, os mesmos são fornecidos para os pesquisadores participantes. Cada pesquisador (ou grupo de pesquisa) fica encarregado de desenvolver e aplicar um modelo de ranqueamento textual e gerar *runs*. Tais *runs* correspondem ao resultado da aplicação dos algoritmos de ranqueamento sobre o conjunto D fornecido, consistindo geralmente nos 1.000 (top-1.000) documentos melhor ranqueados de acordo com seus critérios utilizados pelos pesquisadores.

Fica a cargo da organização do evento combinar as *runs* geradas para gerar as anotações finais. Esta etapa de combinação é denominada por *pooling*. A motivação para esta etapa está na redução do viés proveniente das amostras (TONON; DEMARTINI; CUDRÉ-MAUROUX,

[1] <https://trec.nist.gov/>

2015). A técnica de *pooling* utilizada pela TREC é a *Depth@K*. Segundo esta metodologia, os K documentos melhor posicionados pelos ranqueadores avaliados são extraídos e um novo conjunto é formado a partir da união dos K documentos de cada *run*. Enfim, chega a etapa de anotação humana, em que especialistas analisam a relevância dos documentos do conjunto final à consulta em análise e geram seus julgamentos para compor os *qrels*. É importante observar que, mesmo com as limitações no número de documentos, a etapa de anotação continua sendo um trabalho que consome muito tempo.

2.1.4 Avaliação de Sistemas de Ranqueamento

Comentados os conceitos fundamentais relacionados à tarefa de ranqueamento textual, os próximos conceitos a serem apresentados são as métricas encarregadas de avaliar quantitativamente a performance dos ranqueadores. Para a definição das mesmas, será utilizada a mesma nomenclatura utilizada pelo *survey* de (LIN; NOGUEIRA; YATES, 2021). Seja $R = \{(i, d_i)\}_{i=1}^{l}$ uma lista ranqueada de l documentos, cujos *ranks* i foram determinados a partir de pontuações produzidas um algoritmo de ranqueamento. Observe que uma lista R de documentos é criada a cada consulta Q a ser avaliada.

Antes de abordar as definições das métricas, há um conceito preliminar que precisa ser definido. Frequentemente o número de documentos a serem utilizados para computar as métricas é limitado por um limiar. Um limiar k estabelece que apenas os k-primeiros documentos (baseados em suas pontuações) serão utilizados para o cálculo de uma métrica. Desta forma, a lista R passa a ser expressa por $\{(i, d_i)\}_{i=1}^{k}$, onde $k \leq l$. A nomenclatura utilizada na literatura para indicar o limiar é expressar a métrica avaliada como sendo "*Métrica@k*". A definição do k fica a cargo dos desenvolvedores da aplicação baseada em ranqueamento. Para competições ou conferências destinadas a ranqueamento textual, é frequente a utilização de um $k = 1000$. Por outro lado, para aplicações *web* que utilizam ranqueamento, é frequente utilizar valores entre 10 e 20 uma vez que os textos precisam ser apresentados em uma única página. Logo, o valor do limiar k é um parâmetro que pode ser ajustado conforme as necessidades da aplicação a ser avaliada.

Feita a apresentação das nomenclaturas e definições básicas associadas às métricas de ranqueamento, a seguir serão apresentadas as métricas a serem avaliadas por este projeto de mestrado:

- **Revocação**

A métrica Revocação para ranqueamento textual é definida pela expressão:

$$Revocação(R, Q) = \frac{\sum_{(i,d) \in R} rel(Q, d)}{\sum_{d \in D} rel(Q, d)}, \tag{2.1}$$

na qual $rel(Q,d)$ indica a relevância binária de d comparado à Q. A Revocação busca quantificar a fração dos documentos relevantes a Q em D (o conjunto inteiro de documentos) presentes na lista R. Uma vantagem da utilização da métrica está na facilidade de interpretá-la, já que seu valor expressa "Qual porcentagem dos documentos relevantes à consulta foi retornada pelo método?". Entretanto, é preciso ter em mente que a métrica não considera a ordem em que os documentos relevantes foram posicionados e não é preparada para trabalhar com julgamentos de relevância não binários.

Conforme anteriormente mencionado, o número de documentos a serem considerados para o cálculo da métrica pode ser limitado por um limiar k. Por exemplo, para um $k = 1000$ (utilizado em avaliações no *MS MARCO* (NGUYEN *et al.*, 2016)), a Revocação@1000 corresponderá a fração de todos os documentos relevantes retornados nos 1000 documentos melhor posicionados no ranqueamento avaliado.

- **Precisão**

  Similar a métrica Revocação, a Precisão é definida pela expressão:

$$Precisão(R,Q) = \frac{\sum_{(i,d)\in R} rel(Q,d)}{|R|}, \tag{2.2}$$

  na qual $rel(Q,d)$ apresenta o mesmo significado da métrica anterior. Porém, o denominador corresponde ao tamanho da lista de documentos ordenados R. Assim como a Revocação, a Precisão também é facilmente interpretável através da pergunta: "Dentre os documentos retornados para análise, qual fração deles é relevante para a consulta?". Além disto, a Precisão também compartilha as mesmas desvantagens da Revocação: não considera a ordem dos documentos e também não trabalha com julgamentos de relevância não binários.

- ***Mean Average Precision (MAP)***

  Para entender a métrica *Mean Average Precision* (MAP), é necessário primeiro definir a *Average Precision* (AP). Esta métrica, por sua vez, é definida pela expressão:

$$AP(R,Q) = \frac{\sum_{(i,d)\in R} Precisão@i(R,Q) \times rel(Q,d)}{\sum_{d\in D} rel(Q,d)}, \tag{2.3}$$

  cujos componentes foram discutidos pelas métricas anteriores. Ela representa a Precisão média obtida considerando limiares i que consistem nas posições em que aparecem documentos relevantes. A métrica assume valores maiores (próximos de um) na media em que documentos relevantes são posicionados nas primeiras posições da lista ordenada R. Por avaliar diferentes limiares dentro de seu cálculo, o AP é frequentemente utilizado considerando limiares "@k"maiores (por exemplo: 1000). Assim como as métricas anteriores, AP não lida com definições não binárias de relevância. Enfim, MAP consiste na média das métricas AP obtidas ao avaliar todas as consultas Q do conjunto de avaliação.

- ***Reciprocal Rank (RR)***

  O *rank* recíproco (*Reciprocal Rank*) é dado pela expressão:

$$RR(R,Q) = \frac{1}{rank_i}, \tag{2.4}$$

  na qual $rank_i$ indica a primeira posição ocupada por um documento relevante para a consulta Q especificada na lista R. Assim, a métrica RR considera apenas as posições (*ranks*) associadas aos documentos *relevantes* em seu cálculo. Portanto, a métrica também considera apenas a definição binária para relevância: um documento é relevante ou não para Q. Desta forma, se um documento relevante foi posicionado na primeira posição $i = 1$, o RR assume valor 1. Por outro lado, caso o primeiro documento relevante seja posicionado na posição $i = 5$, temos que $RR = 1/5 = 0.25$.

  Em outras palavras, o valor de RR assume valores maiores (próximos de 1) caso os documentos relevantes sejam posicionados nas primeiras posições pela estratégia de ranqueamento. É importante ressaltar que a métrica considera apenas a primeira ocorrência de um documento relevante em R. Este fato torna a métrica inadequada para avaliar sistemas destinados a usuários que desejam mais de um resultado relevante para suas consultas.

  Assim como a Revocação, as posições a serem avaliadas podem ser limitadas pelo limiar k. Além disto, a métrica é comumente apresentada como resultado da média de RR para diferentes Q de um conjunto de avaliação. Neste caso, a métrica recebe o nome de *Mean Reciprocal Rank* (MRR).

- **nDCG (*Normalized Discounted Cumulative Gain*)**

  A nDCG é uma métrica expressa por dois compoenntes. O primeiro componente é o *Discounted Cumulative Gain* (DCG) expresso pela expressão:

$$DCG(R,Q) = \sum_{(i,d)\in R} \frac{2^{rel(Q,d)} - 1}{log_2(i+1)}, \tag{2.5}$$

  na qual $rel(Q,d)$ representa a pontuação de relevância associada ao documento d. Diferente das demais métricas apresentadas, o DCG permite a utilização de valores não binários para $rel(Q,d)$, permitindo a utilização de valores contínuos que quantifiquem a relevância. A motivação para o DCG está em penalizar a relevância de documentos relevantes caso os mesmos não apareçam nas primeiras posições de R. Assim, ao contrário das outras métricas apresentadas, o DCG considera tanto a relevância dos documentos quanto o *rank* associados aos mesmos. Enquanto o numerador da Equação 2.5 é diretamente proporcional à relevância de d para Q, o denominador é proporcional a posição em que d foi alocado em R. Desta forma, a relevância atribuída é ponderada pela posição em que o documento aparece em R.

O segundo componente do nDCG é o IDCG, o qual representa o DCG "ideal" de R. Sendo assim, ele consiste no cálculo do DCG posicionando todos os documentos relevantes nas primeiras posições de R. Combinando DCG e IDCG, o nDCG é expresso por:

$$nDCG(R,Q) = \frac{DCG(R,Q)}{IDCG(R,Q)}. \tag{2.6}$$

A motivação para a utilização do IDCG está em normalizar a métrica no intervalo $[0,1]$. A métrica nDCG é frequentemente utilizada para avaliar a qualidade de buscas na *web*. Logo, quando utilizada em conjunto de um limiar k, o valor tende a refletir o número de resultados a serem mostrados na página contendo os resultados da busca (geralmente $k \leq 20$).

Por fim, serão apresentados comentários finais sobre as métricas apresentadas. Um ponto importante a ser considerado ao utilizar as métricas é que as mesmas assumem que o algoritmo utilizado para ranqueamento produza pontuações decrescentes conforme a relevância de d para Q diminua. Este fato torna as métricas sensíveis a documentos com pontuações iguais. Por mais extremo que seja o caso de um documento relevante e um não relevante terem pontuações iguais, este evento acaba impactando nos valores gerados pelas métricas.

Um segundo ponto a ser considerado é a forma com que as métricas são agregadas para avaliar múltiplas consultas Q. Conforme anteriormente mencionado, as métricas apresentadas são utilizadas para avaliar os documentos ranqueados (lista R) para uma única consulta Q. A solução empregada na literatura para avaliar múltiplas consultas de um mesmo conjunto de avaliação é computar a média das métricas obtidas em todas as consultas avaliadas. Desta forma, é importante ter em mente que a média não ponderada não considera aspectos como a dificuldade da consulta avaliada e o assunto da mesma.

Por fim, é importante ressaltar que os valores observados pelas métricas são dependentes dos conjuntos de documentos nos quais os ranqueadores textuais foram avaliados. Logo, não faz sentido comparar valores de RR entre corpus distintos. Comparar valores observados em corpus diferentes (ex: *MS MARCO* e *TREC*) é útil apenas para avaliar a robustez dos métodos ao serem expostos a dados diferentes.

2.1.5 Algoritmos para Ranqueamento Textual

O problema de ranqueamento textual pode ser abordado de diferentes formas. Porém, o objetivo de algoritmos de ranqueamento textual é frequentemente o mesmo: gerar pontuações (valores reais) que quantifiquem a relevância de um documento para um consulta. Tendo gerado estas pontuações, o ranqueamento consiste em ordenar os documentos baseadas nas mesmas. Enfim, esta Subseção busca apresentar metodologias comuns para ranqueamento textual, apresentando suas vantagens e desvantagens.

2.1.5.1 Modelo Booleano

O modelo de busca booleano consiste num método de recuperação de documentos baseado em consultas expressas através de expressões lógicas como conjunção (`AND`), disjunção (`OR`) e negação (`NOT`). O método trata documentos como sendo conjuntos de palavras, e utiliza relações de pertinência de conjuntos para produzir subconjuntos a partir das consultas. Exemplos de consultas Q são apresentados a seguir.

$$\text{Cobrança } \texttt{AND} \text{ Indevida} \tag{2.7}$$

$$\text{Divórcio } \texttt{OR} \text{ 'Cobrança indevida'} \tag{2.8}$$

$$\texttt{NOT} \text{ Negado} \tag{2.9}$$

Observe que as consultas 2.7, 2.8 e 2.9 são elaboradas a partir de palavras. Desta forma, no método de busca booleana, as consultas Q são representadas como sendo operações lógicas entre os termos presentes na consulta. Dado um conjunto de documentos D, a consulta 2.7 retorna documentos que possuem os termos "Cobrança" e "Indevida" simultaneamente. A consulta 2.8, por sua vez, retorna a união do conjunto de documentos que possui o termo "Divórcio" com o conjunto de documentos que possuí os termos "Cobrança indevida" aparecendo na ordem apresentada. Por fim, a consulta 2.9 retorna documentos que não possuem o termo "Negado".

Embora seja simples, o modelo booleano não gera pontuações que quantifiquem a relevância de um documento para uma consulta. Logo, os documentos recuperados não poderiam ser ordenados, já que o modelo recupera apenas os documentos que possuem termos em comum com a consulta utilizada. Além disto, outra limitação é a construção das consultas, dado que o método requer que as mesmas sejam expressas através de operações lógicas ao invés de texto em linguagem natural. Por fim, a construção das consultas também requer cuidado na escolha dos termos que as compõem, uma vez que os mesmos precisam estar presentes nos documentos de análise. Não serão avaliados modelos booleanos neste projeto devido a sua baixa performance em relação às demais técnicas a serem descritas.

2.1.5.2 Modelo Probabilístico

Os modelos probabilísticos de recuperação de informações representam um conjunto de técnicas que buscam ranquear documentos estimando sua relevância em relação a uma consulta. Considerando um problema binário onde, um documento pode ser relevante ($R = 1$) ou não ($R = 0$), a relevância de um documento d dado uma consulta q pode ser expressa por:

$$R_{d,Q} = P(R = 1|d, Q) \tag{2.10}$$

e esta representação é o alicerce do *Probability Ranking Principle* (PRP) apresentado por (ROBERTSON, 1977).

Dentre os modelos probabilísticos de recuperação de informação que mais se destacam está o modelo *Okapi Best Match 25* (BM25) (ROBERTSON; WALKER, 1999). Devido a sua simplicidade, tal modelo é frequentemente utilizado como método de base comparativa para métodos de ranqueamento de documentos, além de ser amplamente utilizado em mecanismos de pesquisa (YANG; FANG; LIN, 2018; OUNIS *et al.*, 2005).

Um conceito preliminar a ser discutido, antes da explicação do funcionamento do BM25, é o tradicional método de representação textual *Vector Space Model* (VSM). Ao utilizar este método, documentos são representados como vetores cujas dimensões representam palavras presentes no vocabulário do *corpus* textual analisado. Em outras palavras, cada dimensão $C = \{c_1, c_2, ..., c_n\}$ do vetor representa uma característica ou aspecto do documento em análise. Por fim, a quantidade de dimensões n corresponde ao tamanho do vocabulário definido para o conjunto de documentos analisados.

Existem duas variações principais de representação textual baseada em VSM: Binária (B) e Frequência de Termos (FT). Na primeira, cada dimensão denota a presença ou não da palavra correspondente a dimensão no documento. Já a segunda, cada dimensão corresponde a frequência do termo no documento ponderada pela frequência inversa relativa a todos os documentos. A representação FT mais comum é a através de um vetor esparso *Term Frequency - Inverse Document Frequency* (TF-IDF), apresentada na Equação (2.11).

$$TF - IDF = TF_{d,c} \times log\left(\frac{|D|}{df_c}\right) \tag{2.11}$$

onde $TF_{d,c}$ representa a frequência de um termo (palavra) c em um documento d, $|D|$ denota a quantidade de documentos analisados e df_c representa a frequência de c em todos os documentos de D. A ponderação visa diminuir a magnitude de termos que aparecem frequentemente nos documentos do corpus D.

Observe que foram apresentadas apenas representações que consideram apenas unidades de palavras. Os modelos apresentados podem ainda ser expandidos para considerar combinações de múltiplas palavras (*n-grams*) ao custo de aumentar o número de dimensões n em troca de desempenho.

Utilizando a formulação apresentada, é possível empregar o TF-IDF para tarefas de busca ou ranqueamento textual. A partir das representações TF-IDF geradas para a consulta e para o documento, aborda-se o problema de ranqueamento utilizando uma função de similaridade. Tal estratégia é a base para ranqueamentos baseados em similaridade, que serão discutidos na próxima Seção.

Tendo contextualizado o TF-IDF, o modelo BM25 consiste em uma função de ranquea-

mento que produz uma estimativa da relevância para um documento baseado em uma consulta, empregando conceitos da representação VSM discutida anteriormente. Uma representação desta função é apresentada em 2.13.

$$IDF(d,q) = log\left(\frac{|D|}{df_q}\right) \tag{2.12}$$

$$score(d,Q) = \sum_{q \in Q} IDF(d,q).\frac{(k_1+1).TF_{d,q}}{TF_{d,q}+k_1.(1-b+b.\frac{|D|}{avg_{dl}})} \tag{2.13}$$

Observe que muitos dos termos que compõem 2.13 foram apresentados na Equação 2.11 que apresenta uma representação TF-IDF, porém os termos q, que integram a consulta Q, são analisados em vez dos termos c que compõem o documento. $TF_{d,q}$ representa a frequência do termo q no documento d e avg_{dl} representa o tamanho médio (quantidade de palavras) dos documentos presentes em D. A Equação 2.12 representa a ponderação de frequência inversa para o termo q. Por fim, ambos k_1 e b são parâmetros positivos e calibráveis que podem ser ajustados conforme a necessidade da busca. O parâmetro K_1 pondera a frequência $TF_{d,q}$. Já o valor b controla a importância dada ao tamanho do documento. Assim, o desempenho do modelo é sensível á calibração de seus parâmetros. Além disto, a representação esparsa utilizada tem como desvantagens notáveis a incapacidade de capturar noções de posicionamento e semântica dos termos dos documentos e ser sensível ao tamanho do vocabulário criado (maldição da dimensionalidade).

2.1.5.3 Modelos Baseados em Similaridade

Como discutido nas seções anteriores, representações esparsas (como TFIDF) podem ser facilmente utilizadas em técnicas de recuperação de informações. Entretanto, desde o surgimento de representações textuais densas como *word embeddings* (MIKOLOV *et al.*, 2013; PENNINGTON; SOCHER; MANNING, 2014) e os *Transformers* já mencionados, houve um grande esforço no desenvolvimento de técnicas de ranqueamento de documentos que utilizassem estas representações mais ricas. A grande vantagem destas representações é a capacidade de conter implicitamente informações semânticas do texto original, ao contrário das representações esparsas. Mesmo com esta diferença, as representações citadas preservam a ideia de criar uma representação vetorial para textos.

Quando se tem interesse em comparar representações vetoriais de consultas e documentos, o problema de recuperação *ad-hoc* documentos continua o mesmo, porém a forma com que a ordenação é feita é diferente. Neste cenário, utiliza-se uma transformação $\eta : [p_1, ..., p_n] \rightarrow \mathbb{R}^n$, capaz de transformar uma sequência de palavras em uma representação vetorial. A partir das representações obtidas, aborda-se o problema de ranqueamento utilizando uma função de similaridade ϕ (geralmente semelhança de cossenos). Desta forma, ranqueia-se os documentos baseando-se na semelhança entre o vetor da consulta Q e vetor do documento d, de forma que os

documentos mais semelhantes fiquem nos ranques mais altos. Observe que esta formulação se assemelha bastante a um problema de k-vizinhos mais próximos. Em termos gerais, busca-se estimar a relevância do documento através da expressão:

$$P(R = 1|d, Q) = \phi(\eta(Q), \eta(d)) \tag{2.14}$$

A transformação η pode consistir em uma representação esparsa como TFIDF, uma simples média das *word embeddings* das palavras do texto e até uma representação baseada em *Transformers* (REIMERS; GUREVYCH, 2019). Já como função de similaridade ϕ, podem ser usadas funções simétricas como distância euclidiana e semelhança de cossenos. Assim, tanto a escolha da função de similaridade quanto a de representação textual devem ser feitas com cuidado, pois impactam diretamente na tarefa de recuperação de informações.

2.2 Geração de Texto usando *Transformers*

Esta Seção foca na apresentação de uma visão geral da arquitetura *Transformers* bem como na sua utilização para tarefas de geração textual. Em primeiro lugar, as Subseções 2.2.1 e 2.2.2 introduzem a tarefa de geração textual em conjunto de exemplos de aplicações da mesma, além de exemplificar como avaliar quantitativamente textos gerados. Em seguida, a Subseção 2.2.3 apresenta uma introdução à arquitetura *Transformer*. O seu funcionamento será discutido em maiores detalhes na Subseção 2.2.4. Por fim, a Subseção 2.2.5 discute a utilização de *Transformers* para geração de texto.

2.2.1 Geração de Texto e suas Aplicações

Dentro do contexto de PLN e aprendizado de máquina, a tarefa de geração de texto consiste em, como o próprio nome sugere, desenvolver sistemas capazes de gerar textos automaticamente de forma a imitar características da linguagem natural humana (tom, estilo linguístico, etc). Desta forma, pesquisa e desenvolvimento de tais sistemas é de grande interesse, uma vez que a geração de texto pode ser aplicada em um vasta variedade de tarefas (FLORIDI; CHIRIATTI, 2020). Como exemplo temos tradução automática, sumarização de documentos, geração de sugestões de escrita, desenvolvimento de *chatbots*, etc.

2.2.2 Avaliação de Geração de Texto

Para avaliar o desempenho de sistemas em tarefas de aprendizado supervisionado de maneira quantitativa, é necessária a utilização de métricas especializadas. Considerando geração de texto supervisionada, o cenário não é diferente. Porém, um fato que torna a avaliação de textos gerados particularmente difícil esta no fato de frequentemente não existir uma única referência para comparação. Por exemplo, ao traduzir um texto do inglês para português, nota-se que não

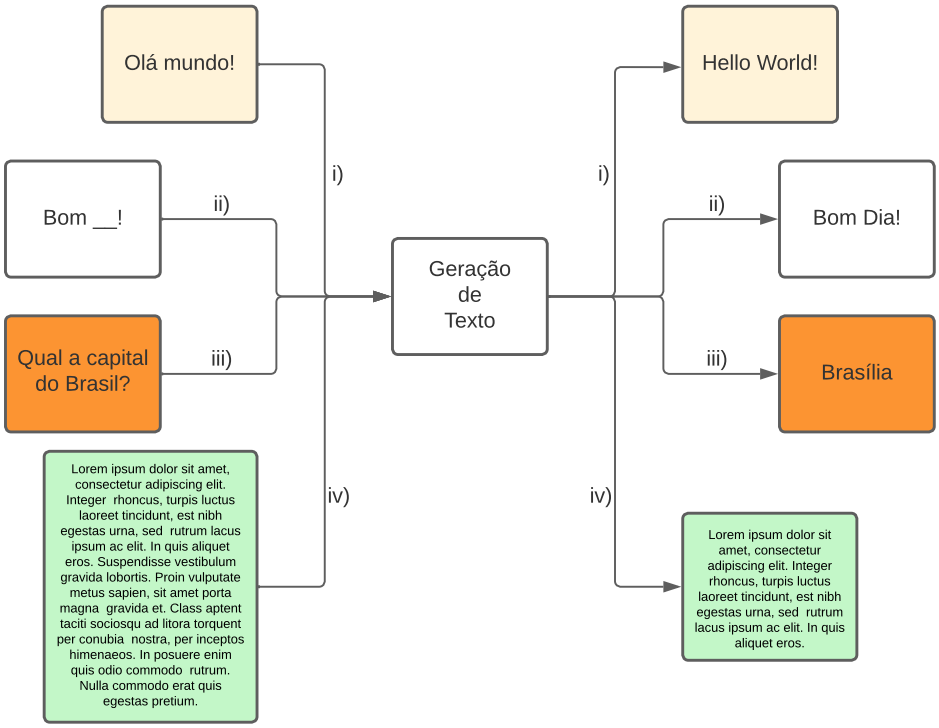


Figura 3 – Exemplos de tarefas que podem ser modeladas como geração de texto: i) tradução, ii) sugestões de escrita, iii) sistemas de perguntas e respostas (ex: *chatbots*) e iv) sumarização.

há apenas uma única tradução possível. Em outro exemplo, em tarefas de sumarização, também é possível sumarizar um mesmo texto de inúmeras formas.

Em um cenário ideal, a avaliação da qualidade de textos artificiais deve ser feita por seres humanos. Entretanto, uma avaliação humana acaba sendo muito custosa o que faz com que muito esforço seja dedicado a elaboração de formas automáticas de avaliação (BELZ; REITER, 2006). Sendo assim, métricas foram desenvolvidas especialmente para avaliar quantitativamente o desempenho de geradores textuais considerando todas as particularidades e variações da tarefa.

A seguir, serão discutidos exemplos de métricas automáticas cujos valores buscam quantificar o quanto os textos gerados artificialmente se aproximam de referências humanas. Como características comuns das métricas a serem discutidas, é possível destacar que elas utilizam casamento de cadeias de texto (*string-matching*) ou n-gramas em seu cálculo. Além disto, as mesmas são frequentemente utilizadas para comparação de sistemas de aprendizado de máquina voltados a geração textual (CELIKYILMAZ; CLARK; GAO, 2020).

2.2.2.1 *Bilingual Evaluation Under-study (BLEU)*

Bilingual Evaluation Under-study (BLEU) (PAPINENI *et al.*, 2002) é uma métrica proposta inicialmente para avaliação de sistemas de tradução. Em linhas gerais, seu funcionamento consiste em mensurar a interseção entre as palavras (ou tokens) presentes no texto gerado e as palavras contidas em um ou mais textos de referência gerados por humanos. Na prática,

as palavras correspondem a n-gramas (entre uma e quatro n-gramas) dos textos tokenizados. Seus valores estão limitados entre 0 e 1, de maneira que, quanto mais próximo de 1, melhor a qualidade do texto gerado.

Além disso, também é importante destacar que a métrica é calculada considerando considerando o corpus de avaliação completo. Sendo assim, as estatísticas são acumuladas com base em todos os textos gerados e todas as referências sem a realização de uma agregação (por exemplo: média) para gerar a pontuação final.

Formalmente, o BLEU está definido da seguinte forma:

$$BLEU = Penalidade \times \left(\prod_{i=1}^{4} Precisão_i \right)^{1/4} \tag{2.15}$$

$$Penalidade = \min\left(1, \exp\left(1 - \frac{\text{tamanho da referência}}{\text{tamanho do texto gerado}}\right)\right) \tag{2.16}$$

$$Precisão_i = \frac{\sum_{\text{tg} \in \text{Gerados}} \sum_{i \in \text{tg}} \min(c^i_{gerado}, c^i_{ref})}{C^i_t = \sum_{\text{tg'} \in \text{Gerados}} \sum_{i' \in \text{tg'}} c^{i'}_{gerado}}, \tag{2.17}$$

na qual i corresponde a i-ésima das i-gramas, *tg* representa um texto gerado pelo sistema a ser avaliado, c^i_{gerado} denota a quantidade de ocorrências simultâneas da i-grama na referência e no texto gerado, c^i_{ref} corresponde a quantidade de ocorrências da i-grama no texto de referência, $c^{i'}_{tg}$ computa a quantidade de n-gramas no texto gerado e C^i_t informa o número total de i-gramas dos textos gerados.

O componente de *Penalidade* (Equação 2.16) da fórmula existe para penalizar a geração de textos curtos, uma vez que seu valor decresce quanto menores os textos gerados. Como o número de termos presentes no texto gerado faz parte do denominador da Expressão 2.17, caso este valor seja pequeno, torna-se fácil maximizar a precisão. Assim, a penalidade atua para favorecer a geração de textos mais longos. Os tamanhos são calculados considerando todos os exemplos (somatório dos tamanhos) dos exemplos gerados e de referência. Já a Equação 2.17, descreve um cálculo de precisão, no qual verifica-se qual porcentagem das n-gramas, do texto gerado, ocorre simultaneamente no texto gerado e no texto de referência.

Enfim, a Equação 2.15 apresenta a combinação dos componentes anteriormente citados. Conforme é possível observar, a métrica BLEU corresponde a média harmônica das precisões das n-gramas (usualmente, trabalhos na literatura utilizam entre 1 e 4) penalizada pelo tamanho do texto gerado.

A vantagem da utilização do BLEU está na simplicidade e rapidez do cálculo da métrica. Além disto, seu valor está positivamente correlacionado com avaliações humanas (PAPINENI *et al.*, 2002). No entanto, é importante destacar que a limitação principal da métrica está no fato de a mesma não considerar aspectos semânticos dos textos comparados uma vez que seu

funcionamento é baseado no casamento de n-gramas. Além disto, a métrica é sensível ao método de tokenização empregado.

2.2.2.2 Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (LIN, 2004) corresponde a um conjunto de métricas voltados à avaliação de geração de texto. Ao contrário do BLEU, a métrica opta por focar mais em revocação (*recall*) nas comparações entre textos gerados e referências. Além disto, a métrica pode ser computada por exemplo para posterior agregação utilizando média (ou mediana) para gerar sua pontuação final. Em sua forma mais tradicional, o ROUGE é definido pela equação:

$$ROUGE - i = \frac{quantidade_{casamento}(i-grama)}{quantidade(i-grama)}, \tag{2.18}$$

na qual o numerador corresponde a quantidade de i-gramas presentes tanto no texto gerado quanto no texto de referência e o denominador representa a quantidade de i-gramas presentes no texto de referência. Assim como o BLEU, opta-se por calcular o ROUGE para n-gramas entre um e quatro (ROUGE-1, ..., ROUGE-4). Ao utilizar a quantidade de i-gramas do texto gerado no denominador, obtemos a Precisão ROUGE-i. Também é possível gerar uma pontuação F ponderando os valores de precisão e revocação ROUGE.

Outro exemplo de variação da métrica é o ROUGE-l. ROUGE-l avalia o maior casamento de sequência de tokens, entre os textos de referência e gerados, empregando o problema da Maior Subsequência Comum (MSC) considerando uma sequência de tokens. Para esta variação também é possível computar precisão, revocação e medida F. Assim, a revocação ROUGE-l é computada pela razão entre o comprimento da MSC e o número de unigramas do texto gerado. Já a precisão ROUGE-l divide o comprimento da MSC pelo número de unigramas do texto referência.

As descrições apresentadas valem apenas para o cenário onde existe apenas uma referência. Para realizar a avaliação de um texto gerado considerando múltiplas referências, computa-se as pontuações ROUGE para cada referência (considerando o mesmo texto gerado), e toma-se como pontuação final a pontuação máxima obtida. Formalmente, para múltiplas referências, modifica-se o ROUGE da seguinte forma:

$$ROUGE - i_{mult} = argmax_i(ROUGE - i(\text{referência}_n, \text{texto gerado}), \text{para N referências}. \tag{2.19}$$

O ROUGE compartilha as mesmas vantagens e desvantagens da pontuação BLEU, sendo rápido de calcular e simples de interpretar porém sensível à tokenização. Sendo assim, a escolha entre as métricas depende do que se prioriza maximizar na comparação entre texto gerado e referências (precisão ou revocação de n-gramas).

2.2.3 Introdução aos Transformers

Apresentada em 2017, a arquitetura *Transformers* (VASWANI *et al.*, 2017) mostrou ser uma grande evolução no estado da arte na área de processamento de linguagem natural. Originalmente proposto para tarefas de tradução (uma tarefa sequência-a-sequência), o *Transformer* é uma rede neural codificadora-decodificadora, capaz de gerar representações densas que, por sua vez, podem ser usadas em diversas tarefas. Embora sua versão original utilize blocos codificadores e decodificadores, os módulos podem ser utilizados separadamente.

O grande diferencial das representações textuais produzidas pela arquitetura é a sensibilidade ao contexto em que o token aparece no texto. Ao contrário de representações estáticas para *tokens* como o *GloVe* (PENNINGTON; SOCHER; MANNING, 2014), as representações geradas pelos *Transformers* são dinâmicas. Ou seja, as representações geradas para um *token* podem mudar dependendo dos demais tokens presentes na sequência. Além disto, os *Transformers* são capazes de realizar processamento sobre sequências textuais de forma mais eficiente e com melhor desempenho do que as redes neurais recorrentes antes utilizadas, tomando o seu lugar em diferentes tarefas de PLN.

(a) Comprei uma *manga* na feira.
(b) A *manga* da camisa está dobrada.

Figura 4 – Exemplo de uma mesma palavra (“manga”) usada em contextos diferentes.

A Figura 4 exemplifica o caso mencionado. Neste caso, a palavra “manga” é apresentada em dois contextos diferentes e com significados diferentes. Neste exemplo, a palavra analisada será tratada como um único token. Uma representação estática geraria uma representação igual para “manga” nos dois casos. Por outro lado, as representações para ‘manga’ geradas pelos *Transformers* serão diferentes para a) e b), uma vez que os contextos onde a palavra se encontra são diferentes.

Uma característica marcante da arquitetura está no seu treinamento baseado em modelagem de linguagem. Seguindo esta metodologia de treinamento, modelos de linguagem são treinados de forma não supervisionada sobre um grande volume de textos. Assim, o também chamado de treinamento auto-supervisionado (*self-supervised*) consiste em utilizar as próprias entradas para determinar automaticamente a função objetivo do treinamento. Geralmente, a função de objetivo do treinamento consiste em predizer probabilidade condicional de uma palavra (ou *token*), dada uma sequência de palavras que a antecedem. O treinamento baseado em modelagem de linguagem de um *Transformer* será discutido em mais detalhes nas próximas seções pois faz parte desta proposta de mestrado.

Os *Transformers*, por sua vez, são compostos formados por dois componentes principais: bloco codificador e bloco decodificador. A seguir será apresentada uma visão geral dos

componentes.

- **Codificador:** Recebe como entrada uma sequência de palavras e tem como objetivo construir uma sequência de *features* (ou características) que contenham conhecimento inferido a partir das entradas originais. Em outras palavras, o bloco codificador atua como um extrator de *features* textuais. Pode ser utilizado individualmente (sem o componente decodificador) para tarefas que requerem compreensão e inferências a partir das entradas da arquitetura (ex: classificação de texto, reconhecimento de entidades nomeadas, comparação semântica entre sentenças, etc) (ex: BERT (DEVLIN *et al.*, 2018))
- **Decodificador:** Utiliza como entrada as *features* geradas pelo componente codificador, em um conjunto de entradas próprias. Utiliza suas entradas para gerar uma nova sequência alvo análoga a sequência de entrada do codificador. Desta forma, o componente decodificador é preferencialmente empregado em tarefas voltadas a geração de sequências (ex: tradução de texto e sumarização), também podendo ser utilizado sem o componente codificador (ex: GPT-2 (RADFORD *et al.*, 2019)).

2.2.4 Arquitetura dos Transformers

Na Figura 5 é apresentada uma arquitetura dos modelos baseados em *Transformers*. Os elementos principais desta arquitetura e de seu funcionamento são apresentados nas Seções a seguir.

2.2.4.1 Representação das Entradas

Conforme apresentado na Figura 5, tanto o bloco codificador quanto o bloco decodificador utilizam representações especiais (*embeddings*) como entradas iniciais. *Embeddings* consistem em vetores unidimensionais densos, os quais são frequentemente associados à sequências de *tokens*. Assim como outros métodos tradicionais, a arquitetura *Transformers* também requer uma etapa de *tokenização* para transformar textos em sequências de *tokens*. Um exemplo de método de *tokenização* é o *WordPiece* aplicado pelo popular modelo BERT (DEVLIN *et al.*, 2018). Tal método constrói seu vocabulário de *tokens* a partir de sub-palavras que sejam mais frequentes no corpus textual analisado. Assim, uma palavra não é necessariamente mapeada para um único *token*. Este caso é ilustrado na Figura 6

A arquitetura *Transformers* utiliza uma combinação de três *embeddings* para gerar suas representação de entrada:

- ***Embeddings* de *tokens***: vetor denso associado a cada *token* pertencente ao vocabulário do método de *tokenização* empregado.
- ***Embeddings* de Segmento**: vetor unidimensional denso que é utilizado quando o processo de inferência do *Transformer* utiliza uma tarefa de predição sobre pares de segmentos

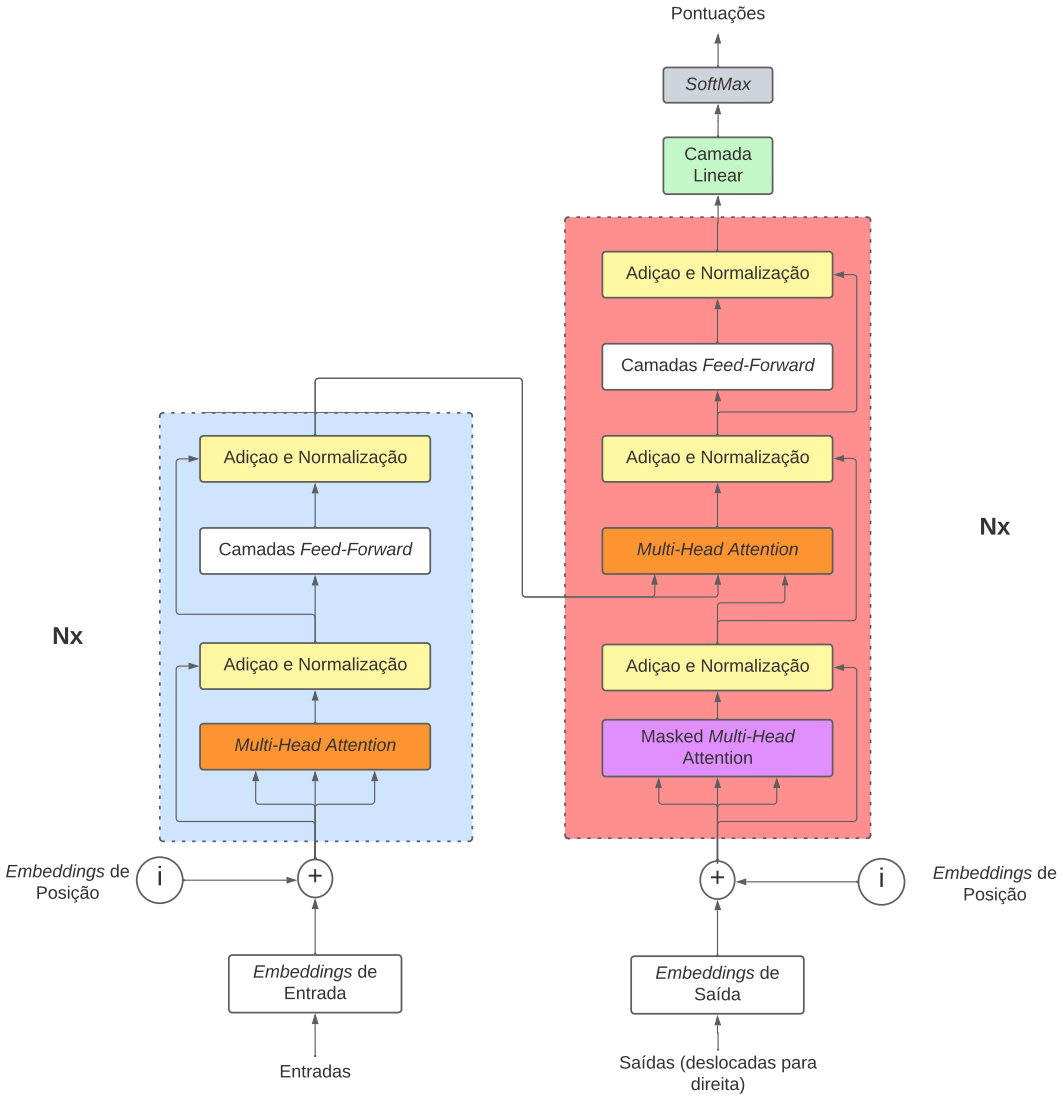


Figura 5 – Representação da arquitetura *Transformers*. Codificador à esquerda e Decodificador à direita. Adaptado de: (VASWANI *et al.*, 2017).

```
Eu trabalho com NLP ⇒ 'Eu', 'trabalho', 'com', 'N', '##LP'
```

Figura 6 – Exemplo de tokenização utilizando o método *WordPiece*. No exemplo, a sigla NLP foi decomposta em dois tokens distintos ('N' e '##LP') pelo algoritmo de tokenização.

textuais. Nesta tarefa, os *tokens* são divididos em dois segmentos S_A e S_B e o objetivo é inferir uma relação entre os dois segmentos (ex: predizer se S_A antecede S_B). Assim, *tokens* pertencentes ao primeiro segmento são associados a um vetor V_A, enquanto os *tokens* do segundo segmento são associados a um vetor V_B.

- ***Embeddings* de Posição**: vetores unidimensionais que são utilizados para indicar a posição do *token* dentro de uma sequência de *tokens*. São construídos de forma que o vetor associado a posição i da sequência seja diferente do vetor associado a posição $i + 1$ e assim

sucessivamente. Como exemplo, BERT utiliza as funções seno e cosseno para gerar uma representação única para cada elemento da sequência.

As *embeddings* são então combinadas através da soma elemento a elemento para servirem de entrada para os blocos codificador e decodificador. É importante ressaltar que a quantidade de dimensões destas representações é um parâmetro customizável e deve ser a mesma para cada tipo de *embedding* para permitir as operações elemento a elemento. Por exemplo, a versão **base** do BERT utiliza 768 dimensões para suas *embeddings*. Assim, a soma dos três tipos de *embeddings* será referida como *embeddings* de entrada nas próximas seções.

2.2.4.2 Mecanismo de Atenção

O mecanismo de atenção é o principal responsável por gerar as representações densas para cada *token* fornecido como entrada para o modelo *Transformer*. De forma geral, seu funcionamento consiste em gerar representações densas a partir de três representações intermediárias: *queries* (Q) e pares *key-value* (K e V) . Assim, a primeira etapa do seu funcionamento consiste em gerar representações Q, K e V a partir das *embeddings* de entrada comentadas anteriormente. Seja X uma sequência de *embeddings* de entrada. As representações Q, K e V são geradas por:

$$\begin{aligned} Q &= XM^Q \\ K &= XM^K \\ V &= XM^V \end{aligned} \tag{2.20}$$

As representações geradas pelas matrizes M^Q, M^K e M^V possuem as mesmas dimensões das entradas X. Tendo gerado as representações para *query*, *key* e *value*; o mecanismo de atenção é expresso por:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{2.21}$$

A Equação 2.21 descreve uma *attention-head* e seus componentes serão discutidos a seguir. A multiplicação das representações Q pela transposição das representações K consiste na comparação par a par das representações das *queries* e *keys*, através de produtos escalares. Em outras palavras, a matriz resultante QK^T consiste em uma matriz quadrada que armazena valores que quantificam a similaridade par a par dos *tokens* da sequência de entrada. Em seguida, os valores desta matriz são divididos pelo escalar $\sqrt{d_k}$ [2] com o objetivo de controlar a escala dos valores da matriz e proporcionar gradientes mais estáveis.

Considerando a escala dos valores de QK^T, a próxima etapa consiste em aplicar a função *softmax* nos valores da matriz com o objetivo de gerar pontuações entre 0 e 1 que estimem a

[2] d_k corresponde ao número de dimensões dos vetores Q, K e V.

"atenção a ser dada" a um determinado *token* da sequência de entrada. Por fim, as representações finais são obtidas multiplicando as pontuações *softmax* pelas representações V. Em outras palavras, o objetivo da multiplicação é ponderar os elementos da sequência V de forma que os elementos de maior importância tenham mais participação na geração das representações finais.

Vale observar que a Equação 2.21 descreve o funcionamento de *um único mecanismo de atenção* (ou *attention-head*). Os autores da arquitetura *Transformer* identificaram que é benéfico ter múltiplos mecanismos de atenção operando simultaneamente, porém com matrizes M^Q, M^K e M^V diferentes. Assim, a ideia é ter múltiplas *attention-heads* operando e que cada mecanismo busque características diferentes durante a etapa de aprendizado. Desta forma, a atenção final de um modelo *Transformer*, chamada de *Multi-Head Attention* (Figura 5), é expressa por:

$$\begin{aligned} MultiHead(X) &= Concat(A_1, \ldots, A_h)M^O \\ \text{onde } A_i &= Attention(XM_i^Q, XM_i^K, XM_i^V) \end{aligned} \tag{2.22}$$

De forma resumida, a Equação 2.22 revela que a saída gerada pela atenção de um *Transformer* consiste na concatenação das saídas de h mecanismos de atenção distintos, multiplicados por uma matriz M^O. Esta multiplicação final tem por objetivo reduzir as dimensões das atenções concatenadas, de maneira que a saída final tenha as mesmas dimensões de uma única *attention-head*. Utilizando o modelo BERT **base** como exemplo, as dimensões de saída do módulo de atenção são 768 (mesmo número de dimensões de suas *embeddings* de entrada).

2.2.4.3 Blocos Codificador e Decodificador

Tendo descrito o funcionamento do módulo *Multi-Head Attention* na subseção anterior, as demais operações, ilustradas na Figura 5, são aplicadas de maneira sequencial. Focando no componente codificador de um *Transformer*, o funcionamento de um único bloco codificador é dado por:

$$\begin{aligned} X^1 &= MultiHead(X) \\ X^2 &= Normalização(X^1 + X) \\ X^3 &= ReLU(ReLU(X^2W_1 + b_1)W_2 + b_2) \\ X^f &= Normalização(X^3 + X^2) \end{aligned} \tag{2.23}$$

Na Equação anterior, X^1, X^2, X^3 indicam representações intermediárias enquanto X^f consiste na representação final gerada pelo codificador. X corresponde a soma das *embeddings* de *tokens*, de segmento e de posição anteriormente apresentados (Seção 2.2.4.1). Pela Equação 2.23 é possível observar que mesmo utilizando o mecanismo de atenção, o bloco codificador se assemelha a redes neurais *feed-forward* tradicionais. A atenção é aplicada para gerar a representação intermediária X^1. Em seguida, ocorre a aplicação de duas técnicas especiais para geração da representação X^2. Tanto a representação X^2 quanto a X^f utilizam conexões residuais,

seguidas de uma normalização de camada (*Layer Normalization*) (BA; KIROS; HINTON, 2016) para sua criação. A representação intermediária X^3, por sua vez, é gerada por duas transformações lineares, com a aplicação da função de ativação *ReLU* sobre as saídas das mesmas.

É importante ressaltar que a Equação 2.23 descreve apenas um bloco codificador. Utilizando o BERT novamente como exemplo, o mesmo utiliza 12 blocos de *Encoders* em seu modelo **base**, aplicados sequencialmente. Outro aspecto interessante do codificador está no fato de o bloco ter acesso a todos os *tokens* da sequência ao mesmo tempo para gerar as representações finais. Além disto, o número de dimensões da saída dos *Encoders* não é alterado.

O funcionamento de um bloco decodificador é similar ao bloco codificador, com quatro mudanças principais. O decodificador utiliza um próprio conjunto de *embeddings* análogas as apresentadas na Seção 2.2.4.1. Além disto, a entrada do decodificador é processada por uma versão modificada do módulo *Multi-Head Attention* anteriormente descrito. Como o componente decodificador é utilizado geralmente para tarefas de geração de sequências, é interessante que o mesmo gere um novo *token* olhando apenas para os *tokens anteriormente gerados* sem olhar para todos os *tokens* da sequência. Assim, ao contrário do codificador, o componente decodificador limita o número de *tokens* em que serão aplicados o mecanismo de atenção nesta primeira etapa. Desta forma, este módulo modificado recebe o nome de *Masked Multi-Head Attention* devido ao fato de ocultar parte dos *tokens* da sequência de entrada.

A terceira diferença está no fato de o decodificador utilizar as representações geradas pelo codificador como representações Q e K para o seu módulo *Multi-Head Attention* (Figura 5). As saídas do módulo *Masked Multi-Head Attention* (seguida pela adição e normalização das conexões residuais) são utilizadas como representações V. Por fim, a última diferença está na existência de um módulo linear seguido de uma função *softmax* ao final do decodificador. Estes dois componentes são utilizados para predizer um *token* a ser gerado a partir da sequência gerada até o momento em tarefas de geração textual. Tendo gerado um novo *token*, o mesmo passa a compor a entrada do módulo decodificador.

2.2.4.4 Transferência de Aprendizado

Para finalizar a apresentação dos *Transformers*, será discutido um último conceito que foi importante para popularização da arquitetura em tarefas de PLN. Este conceito recebe o nome de Transferência de Aprendizado (*Transfer Learning*). Conforme o nome sugere, a técnica busca fazer com que o conhecimento obtido em um domínio seja transferido para outro. Em outras palavras, espera-se que o conhecimento obtido em um domínio A seja generalizado para um domínio B. Neste ponto, é importante observar que tal transferência é apenas possível caso os domínios estejam relacionados de alguma forma (ZHUANG *et al.*, 2020).

A técnica é particularmente útil em cenários em que um domínio possui uma abundância de dados e domínio da tarefa alvo não. Assim, a utilização da técnica para aprendizado supervisionado consiste em treinar um modelo em um domínio com grande quantidade de dados, e

depois treiná-lo novamente no conjunto de dados destinado a tarefa alvo. Tendo apresentado esta contextualização, é possível realizar uma comparação entre *Transformers* e modelos de visão computacional. Em visão computacional, o procedimento de transferência de aprendizado é tradicional e utilizado da seguinte forma. Um modelo de visão computacional é inicialmente treinado em um grande conjunto de dados (como o *ImageNet* [3]), com o objetivo de aprender a extrair características comuns das imagens do conjunto de dados (ex: formas e contornos). Para o conjunto de dados *ImageNet*, tal treinamento inicial consiste em um problema de classificação de imagens com 1.000 distintas. Tendo feito este treinamento inicial, o modelo treinado pode ser aplicado em outras tarefas de visão computacional com sutis alterações em sua arquitetura (troca da camada linear de saída para um problema de classificação, por exemplo). Exemplos de tarefas de visão computacional são: classificação, segmentação e detecção de objetos.

Para o contexto de PLN, a transferência de aprendizado utilizando *Transformers* é feita de forma semelhante. Como exemplo, o modelo *BERTimbau* (SOUZA; NOGUEIRA; LOTUFO, 2020), baseado no componente codificador dos *Transformers*, foi treinado inicialmente no conjunto de dados *brWaC* (FILHO *et al.*, 2018). Tal corpus é composto por textos em português de diferentes temas, e possui 2.86 Bilhões de tokens. De forma semelhante ao que é feito em visão computacional, o treinamento é feito em grandes conjuntos textuais visando aprender estruturas linguísticas comuns da linguagem alvo (português). O treinamento dos *Transformers* nesta etapa é baseado em modelagem de linguagem, conforme anteriormente discutido. Assim como o modelo BERT, no qual o *BERTimbau* foi baseado, as representações para os *tokens* podem ser utilizadas em diferentes tarefas de PLN: classificação, reconhecimento de entidades nomeadas, agrupamentos, etc.

Em ambas as situações apresentadas, os modelos tiveram uma etapa de treinamento inicial em grandes conjuntos de dados com o objetivo de aprender características comuns de suas áreas de interesse (imagens e textos respectivamente). Nos dois casos, o treinamento inicial não necessariamente corresponde a tarefa final na qual o modelo de aprendizado de máquina será aplicado. Além disto, o pré-treinamento em grandes bases de dados possibilita boa performance na tarefa final, na qual os modelos serão aplicados, mesmo que a mesma não tenha uma grande quantidade de dados disponível para o treinamento supervisionado. Após esta etapa de treinamento inicial, os modelos pré-treinados podem ser disponibilizados publicamente[4,5]. Este aspecto é importante pois permite que pesquisadores tenham acesso a uma grande coleção de modelos pré-treinados, prontos para serem aplicados em suas tarefas alvo.

[3] <https://www.image-net.org/>
[4] <https://pytorch.org/hub/>
[5] <https://huggingface.co/models>

2.2.5 Utilização de Transformers para Geração de Texto

Conforme apontado pela Seção 2.2.3, a arquitetura *Transformer* foi proposta para a tarefa de tradução. No entanto, a mesma arquitetura pode ser facilmente aplicada para outras tarefas que envolvem geração de sequências (ex: sumarização), apenas alterando os pares de entradas e saídas. Em específico, muita atenção vem sido dada para geração de texto utilizando *Transformers* após os sucessos apresentados pelos modelos da família GPT (RADFORD *et al.*, 2019; FLORIDI; CHIRIATTI, 2020).

A geração de texto feita por *Transformers* é frequentemente denominada auto-regressiva. Esta denominação está baseada no fato da geração de texto empregada pela arquitetura assumir que a distribuição de probabilidade, que descreve uma sequência de palavras, ser composta por produtos de probabilidades condicionais das palavras antecedentes. Utilizamos o termo "palavra" neste caso para facilitar o entendimento, porém na prática os modelos geradores geram tokens que nem sempre coincidem com palavras da língua em análise. No caso dos *Transformers*, o modelo utiliza um contexto de tokens anteriores (de tamanho customizável como parâmetro) para predizer um próximo token mais provável dado este contexto. Formalmente, é possível descrever a geração condicional pela expressão:

$$P(w|W_0) = \prod_{t=1}^{T} P(w_t|[w_{t-1},...,w_0],W_0), \tag{2.24}$$

na qual para uma sequência de palavras de tamanho T, determinamos a probabilidade de uma palavra em um instante t ocorrer a partir das probabilidades condicionais das palavras anteriores ocorrerem. W_0 representa um contexto inicial opcional.

A seguir serão discutidos exemplos de métodos de decodificação utilizados para geração de texto. O papel dos métodos de decodificação é guiar a geração de texto a partir das probabilidades estimadas pelos modelos em análise. Em outras palavras, os métodos de decodificação são responsáveis por transformar as representações internas utilizadas por modelos de linguagem em texto legível por humanos.

2.2.5.1 Decodificação Gulosa

Conforme discutido na Subseção 2.2.4.3, ao empregar a arquitetura codificador-decodificador completa, um token é predito ao se aplicar uma função *softmax* sobre as saídas finais do bloco decodificador (também chamadas de *logits*). Para então predizer um token, o *Transformer* deve escolher um dentre todos os armazenados dentro do seu vocabulário. Sendo assim, a decodificação gulosa propõe escolher o token predito como sendo o que maximiza a probabilidade condicional em um dado instante. Em outras palavras, o token predito é dado por:

$$w_t = argmax_w P(w|w_{t-1},...,w_0), \tag{2.25}$$

para w_t igual a uma palavra do vocabulário do modelo. Esta consiste na abordagem mais simples para decodificação em geração textual, porém tem como desvantagem a possibilidade de escolher sequências de menor probabilidade ao considerar uma sequência de tokens completa, uma vez que considera apenas os máximos locais.

Como proposta para solução da desvantagem apresentada, existe a possibilidade de utilizaram algoritmo *beam-search* em conjunto da decodificação. Esta modificação propõe a predição de um token a partir da análise dos tokens mais prováveis em predições (ou instantes) anteriores. Desta forma, busca-se estimar o token mais provável com base em uma pequena "memória" de predições anteriores. O tamanho desta memória é customizável e recebe o nome de número de *beams*.

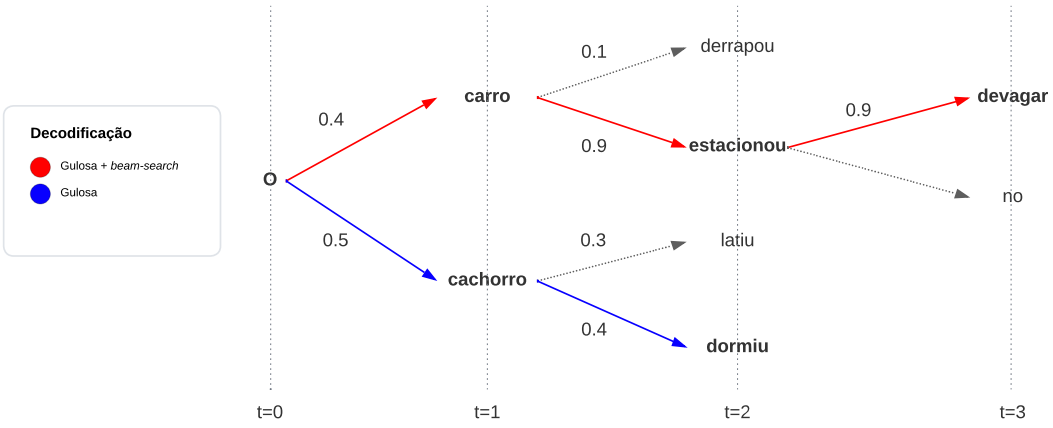


Figura 7 – Comparação da decodificação gulosa usando ou não *beam-search*.

A Figura 7 apresenta uma comparação das duas versões da decodificação gulosa: com e sem *beam-search*. Considerando um número de *beams* igual a dois e um token inicial "O", observamos que a decodificação gulosa sem *beam-search* produz a sequência ("cachorro", "dormiu") com probabilidade 0.2. No entanto, ao utilizar *beam-search*, a sequência produzida ("carro", "estacionou") tem uma probabilidade maior (igual a 0.36). Embora produza sequências de maior probabilidade, é necessário ter em mente que o uso de *beam-search* introduz um parâmetro a mais no sistema de geração textual (número de *beams*).

2.2.5.2 Decodificação Utilizando Amostragem

Por mais promissora que a geração automática de texto seja, ela ainda sofre de problema como texto repetitivo, incoerente e previsível por consequência da frequente escolha dos tokens mais prováveis (HOLTZMAN *et al.*, 2019). Como forma de amenizar os problemas citados, é possível escolher os tokens preditos com base em suas probabilidades condicionais $P(w|w_{t-1})$. Observa-se que, ao utilizar a distribuição de probabilidades, a geração textual torna-se não-determinística.

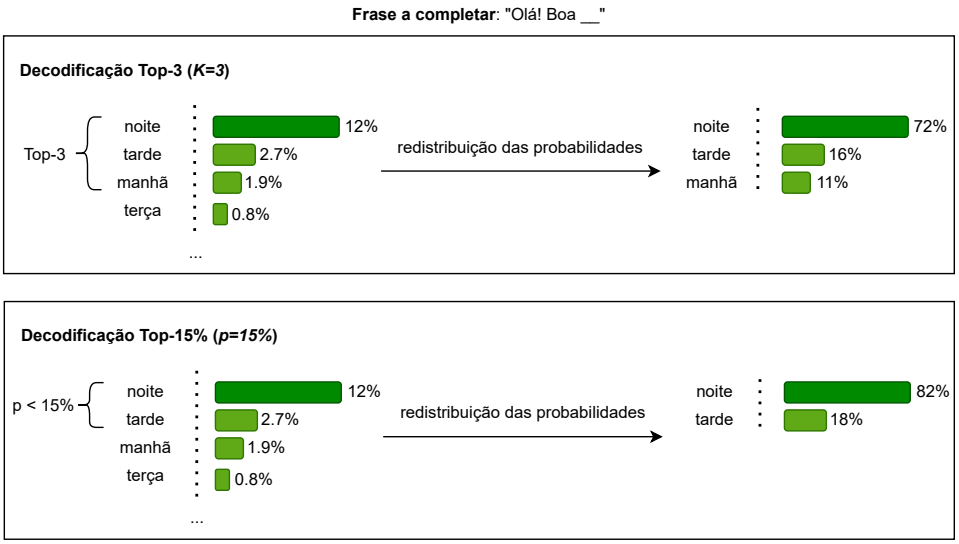


Figura 8 – Comparação entre as decodificações *top-K* e *top-p*.

Uma das formas mais práticas e efetivas de aplicar a decodificação com amostragem é utilizar o esquema *top-k* (FAN; LEWIS; DAUPHIN, 2018b) . A abordagem apresentada pelos autores consiste em filtrar os K tokens mais prováveis em um dado instante, e redistribuir as suas probabilidades entre os mesmos. Outra abordagem para decodificação não-determinística é a amostragem *top-p* (HOLTZMAN *et al.*, 2019). Este esquema de amostragem, por sua vez, propõe que a escolha do token predito seja limitada a um conjunto de tokens mais prováveis cujas probabilidades somadas sejam inferiores ao limiar p estabelecido. A maior vantagem do esquema está no fato de o número de tokens que podem ser escolhidos em um dado instante é dinâmico, já que as distribuições de probabilidade variam a cada instante de predição. A Figura 8 exemplifica os dois métodos ddiscutidos.

2.3 Considerações Finais

O presente Capítulo apresentou os conceitos básicos das duas grandes áreas em que esta Dissertação de Mestrado se situa: recuperação de informações e geração de texto. Quanto a primeira área de interesse, foram apresentados desde conceitos introdutórios até abordagens tradicionais para ranqueamento textual. Já a respeito da geração de texto, a área foi introduzida e exemplos de suas aplicações foram apresentados. Em seguida, foi discutido o atual estado da arte em geração textual e como o mesmo pode ser aplicado. O próximo Capítulo discutirá exemplos de trabalhos relacionados às áreas de interesse desta Dissertação.

CAPÍTULO

3

TRABALHOS RELACIONADOS

Neste capítulo, são apresentados alguns trabalhos relevantes para o desenvolvimento desta pesquisa. Os trabalhos são organizados em duas seções, que focam nas duas grandes áreas de interesse desta pesquisa: geração de texto e recuperação de informações no domínio jurídico.

3.1 Geração de Texto

Apresentada em 2017, a arquitetura *Transformer* (VASWANI *et al.*, 2017) passou a dominar o estado da arte de diferentes tarefas de PLN. Originalmente utilizada para tradução (uma tarefa de geração textual sequência-para-sequência), o *Transformer* é uma rede neural codificadora-decodificadora, que utiliza mecanismos de atenção para gerar representações sensíveis ao contexto para uma sequência de tokens que podem ser usados em diversas tarefas de PLN. Embora a arquitetura original utilize uma composição de dois módulos (codificador-decodificador), os mesmos podem ser utilizados separadamente.

Esse é o caso de modelos *Transformer* que utilizam apenas blocos decodificadores, como o *Generative Pretrained Transformers* (GPT) (RADFORD *et al.*, 2019). Usados principalmente para tarefas de geração de texto, os modelos baseados em GPT alcançaram grande popularidade devido a sua capacidade de realizar uma ampla variedade de tarefas, como responder perguntas, realizar traduções, resumir textos, atuar como *chatbots*, etc (FLORIDI; CHIRIATTI, 2020). Além disso, os modelos desta linhagem também apresentaram potencial de aprender utilizando poucos e até nenhum exemplo (aplicações *few-shot* e *zero-shot*) (BROWN *et al.*, 2020). Tais fatos, em 2022, levaram a diferentes tentativas de replicar o desempenho do GPT, porém usando alternativas de código aberto.

Open Pretrained Transformers (OPT) (ZHANG *et al.*, 2022) é um exemplo desses esforços. Desenvolvido pela *Meta*, o trabalho apresentou uma coleção de oito *Transformers* com arquiteturas semelhantes ao GPT. Os modelos foram criados e pré-treinados com o objetivo de

aproximar o número de parâmetros e o desempenho dos modelos do GPT-3. Outra iniciativa que segue um objetivo semelhante é o *BigScience Large Open-science Open-access Multilingual Language Model* (BLOOM) (SCAO *et al.*, 2022), criado por uma colaboração de centenas de pesquisadores de diferentes nacionalidades.

A arquitetura completa do *Transformer* (codificador-decodificador) também é utilizada em várias tarefas de PLN. Em 2020, Raffel *et al.* (RAFFEL *et al.*, 2020) apresentou um estudo propondo a unificação de uma série de tarefas de PLN em uma única tarefa de geração de texto (também chamada de texto-para-texto ou sequência-para-sequência). Por exemplo, uma tarefa de classificação, considerando o *framework* proposto, tornaria-se uma tarefa de geração textual na qual o objetivo é gerar o texto do rótulo da classe predita. Um resultado interessante dos autores é que, considerando a tarefa texto-para-texto proposta, a configuração original codificador-decodificador obteve desempenhos melhores do que os modelos somente decodificador avaliados. Este trabalho, por sua vez, foi posteriormente expandido por Wue *et al.* (XUE *et al.*, 2020), com o objetivo de adicionar suporte múltiplas línguas, porém preservando a proposta original.

Embora os exemplos de abordagens de geração textual apresentados sejam diferentes em termos de arquiteturas e escala (número de parâmetros), todos lidam com problemas comuns relativos à qualidade do texto gerado artificialmente. Textos gerados por modelos similares aos citados são frequentemente simplistas, incoerentes ou acabam sendo repetitivos (HOLTZMAN *et al.*, 2019). Há também a possibilidade do gerador textual "alucinar", gerando textos contraditórios, sem sentido e sem embasamento ou evidências (JI *et al.*, 2022; MAYNEZ *et al.*, 2020). Por mais que modelos mais recentes e de maior escala (número de parâmetros) (OPENAI, 2023) tenham mitigado parte dos problemas apontados, alucinações ainda persistem (BUBECK *et al.*, 2023).

Como forma de mitigar os primeiros desafios citados (textos repetitivos e previsíveis), algumas iniciativas foram propostas visando tornar a geração textual não-determinística (HOLTZMAN *et al.*, 2019; FAN; LEWIS; DAUPHIN, 2018b). Tais propostas surgiram como alternativas aos métodos mais simples de geração textual (também chamados de decodificação gulosa), argumentando que a busca por escolher sempre palavras (ou tokens) mais prováveis é uma das principais causas de textos repetitivos.

Outro exemplo de pesquisa visando a mitigação de textos repetitivos é o *contrastive-search* (SU *et al.*, 2022). Proposto em 2022, a estratégia consiste em uma modificação na escolha de palavras (ou tokens) preditos um gerador textual, a qual visa aumentar a variabilidade do texto mantendo sua coerência. Para este fim, os autores sugeriram penalizar, durante a decodificação ou treino não supervisionado do modelo de linguagem, as pontuações *softmax* dos tokens mais prováveis pela semelhança em relação aos outros tokens dentro do contexto visando minimizar a ocorrência de textos repetidos. Para atingir tal objetivo, a introdução de um parâmetro *alpha* foi realizada que, por sua vez, controla a importância a ser dada a similaridade aos demais tokens do contexto.

Tendo apresentado os geradores textuais que constituem o estado da arte e suas peculiari-

dades, a seguir serão apresentados exemplos de aplicações dos mesmos na área de interesse deste trabalho. Antes, é importante destacar que verbetações, como as estudadas por este trabalho, existem apenas em documentos utilizados nos tribunais do Brasil e, para nosso conhecimento, este é o primeiro estudo que emprega *Transformers* para geração automática de verbetações. Sendo assim, este trabalho apresenta uma contribuição inédita, apresentando um primeiro esforço em direção a automatização da escrita deste campo textual usando técnicas modernas de PLN.

Retomando as discussões a respeito de trabalhos relacionados, nota-se que, embora tenha sido apresentada em 2017, já existem vários trabalhos visando aplicar a arquitetura *Transformer* para geração de texto no domínio jurídico. Este fato pode ser visto no trabalho de Feijo e Moreira (FEIJO; MOREIRA, 2019), que aplicaram a arquitetura completa para resumir as decisões do Supremo Tribunal Federal (STF), obtendo resultados superiores a métodos extrativos tradicionais considerando a métrica ROUGE avaliada. Com um objetivo similar, Yoon *et al.* (YOON *et al.*, 2022) também obtiveram sucesso aplicando arquiteturas baseadas em *Transformers* (BERT2BERT e BART (ROTHE; NARAYAN; SEVERYN, 2020; LEWIS *et al.*, 2019)) para realizar sumarização de casos jurídicos em coreano, disponibilizados a partir de uma plataforma pública da Coreia do Sul.

Peric *et al.* (PERIC *et al.*, 2020) propuseram a utilização de *Transformers* para gerar opiniões a cerca de casos jurídicos oriundos do *U.S Circuit Court*, empregando uma arquitetura codificador-decodificador baseada no *Transformer-XL* (DAI *et al.*, 2019). Outro exemplo é o trabalho de Huang *et al.* (HUANG *et al.*, 2021). Em seu trabalho, os autores propuseram uma solução para as sub-tarefas de Previsão de Julgamento Legal (PJL) utilizando a abordagem texto-para-texto do modelo T5.

Como um último exemplo de trabalho que utiliza *Transformers* no domínio legal, encontra-se o trabalho de Althammer *et al.* (ALTHAMMER *et al.*, 2021). Os autores investigaram o potencial de utilizar um *Transformer* sumarizador como parte de um *pipeline* de recuperação de informações no domínio legal desenvolvido para a competição *Competition on Legal Information Extraction/Entailment* (COLIEE) de 2021. Neste contexto, a sumarização foi utilizada visando reduzir o tamanho dos documentos que irão para o próximo estágio do *pipeline*, com o objetivo de reduzir os requisitos computacionais.

3.2 Recuperação de Informações

Ao trabalhar com RI, uma prática comum é usar um método simples e computacionalmente eficiente para realizar uma filtragem inicial dos resultados. Neste contexto, o método tradicional Okapi BM25 (ROBERTSON; WALKER, 1999) é frequentemente usado nesta função, sendo utilizado em conjunto de ranqueadores textuais do estado da arte baseados em *Transformers* (LIN; NOGUEIRA; YATES, 2021). Considerando recuperação de informações em documentos jurídicos, o tradicional modelo mantém a sua importância como um referencial

competitivo (ROSA *et al.*, 2021b) para comparações. Pradeep *et al.* (PRADEEP *et al.*, 2020) também observou resultados competitivos para o BM25 ao competir nas tarefas *Health Misinformation and Precision Medicine Tracks* da edição de 2020 da *Text Retrieval Conference* (TREC) trabalhando em um domínio e tarefa diferentes (textos a respeito de desinformação a respeito do COVID-19).

Em 2020, Gomes e Ladeira (GOMES; LADEIRA, 2020) apresentaram uma aplicação de RI para recuperação de jurisprudências disponibilizadas pelo mecanismo de busca do Supremo Tribunal de Justiça (STJ), aplicando o método BM25 no domínio jurídico. O trabalho avaliou técnicas tradicionais de ranqueamento (BM25 e representações densas *Word2Vec*) em um conjunto de dados composto por ementas de documentos do STJ. Os resultados apresentados pelos autores demonstram que os métodos de RI avaliados foram capazes de superar a performance de consultas booleanas utilizadas por padrão no sistema estudado considerando a métrica nDCG@25 (*Normalized Discounted Cumulative Gain*) avaliada.

Ainda considerando RI sobre documentos em português, Oliveira *et al.* (OLIVEIRA; JUNIOR, 2018) e Souza *et al.* (SOUZA *et al.*, 2021) avaliaram o impacto de diferentes técnicas de *stemming* para português na tarefa de ranqueamento de documentos legislativos. *Stemming* (ou stemização) consiste em técnicas que visam reduzir palavras a sua raiz, o que é frequentemente útil como forma de pré-processamento para aplicação de técnicas tradicionais de PLN. As técnicas de *stemming*, por sua vez, foram utilizadas em conjunto de variações do tradicional método de RI BM25. Os trabalhos focaram em jurisprudências do Tribunal de Justiça do Estado de Sergipe e documentos oriundos da Câmara de Deputados do Brasil, respectivamente. Ambos os trabalhos reportaram que a redução de dimensionalidade causada pelo uso do *stemming* (redução do vocabulário para modelos baseados em *bag-of-words*) gera resultados positivos apenas em alguns casos.

Ao trabalhar com sistemas de recomendação, Ostendorff *et al.* (OSTENDORFF *et al.*, 2021) apresentou uma avaliação de vários métodos de representação de documentos para a tarefa de recomendação de literatura jurídica. Esse trabalho usou dados disponíveis publicamente pela *Harvard Law School Library* e pela Suprema Corte dos EUA para os experimentos. Segundo seus experimentos, a média de representações *fastText* de cada token (BOJANOWSKI *et al.*, 2017) (treinados nos corpora usados) obteve melhor desempenho no problema estudado em termos das métricas de RI avaliadas.

Em contrapartida ao trabalho anteriormente citado, os trabalhos de Mandal *et al.* (MANDAL *et al.*, 2021), Lima *et al.* (LIMA; COSTA; ARAÚJO, 2021) e Pedroso *et al.* (PEDROSO; LADEIRA; FALEIROS, 2019) apresentaram resultados que favorecem métodos de representação esparsa para representação de documentos jurídicos. Mandal *et al.* (MANDAL *et al.*, 2021) avaliaram 56 métodos de representação para busca por similaridade usando documentos dos Casos da Suprema Corte da Índia e, apesar de avaliar métodos densos modernos como o BERT (DEVLIN *et al.*, 2018), os autores obtiveram resultados mais favoráveis (em termos de correlação

de *Pearson*) usando representações esparsas *bag-of-words*.

Lima *et al.* (LIMA; COSTA; ARAÚJO, 2021) realizaram um trabalho semelhante ao anterior, porém usando documentos do Tribunal de Justiça do Rio Grande do Norte (TJRN). Neste trabalho, os autores também obtiveram resultados favoráveis para representações esparsas ao usá-las para agrupar documentos jurídicos semelhantes, argumentando que agrupar documentos jurídicos pode depender mais de aspectos sintáticos do que semânticos.

Pedroso *et al.* (PEDROSO; LADEIRA; FALEIROS, 2019), por sua vez, compararam os métodos esparsos BM25 e TFIDF aos métodos semânticos *Latent Semantic Indexing* (LSI) e *Latent Dirichlet Allocation* (LDA) utilizando uma tarefa de RI no domínio legal. Ao utilizar consultas e documentos oriundos do Ministério Público do Distrito Federal e Territórios (MPDFT), os autores obtiveram resultados melhores em termos de *Normalized Discounted Cumulative Gain* (nDCG) para o método BM25, embora sem diferenças significativas na métrica avaliada.

3.3 Considerações Finais

Este Capítulo apresentou exemplos de trabalhos relacionados diretamente ao tema abordado nesta Dissertação de Mestrado. Foram apresentadas as estratégias mais atuais utilizadas para geração textual, bem como exemplos de trabalhos de RI que aplicam *Transformers*. Os modelos apresentados na discussão a respeito de geração de texto representam o estado-da-arte da tarefa, e serão posteriormente avaliados para geração de verbetações. Conforme mencionado, por mais que existam aplicações dos modelos em Direito, não existem trabalhos explorando a utilização de *Transformers* para geração de verbetações. Tal fato reforça a contribuição inédita deste trabalho.

Conforme apresentado anteriormente, as verbetações são criadas com o intuito de auxiliar em tarefas de RI. Sendo assim, optamos por também avaliar as verbetações geradas utilizando uma tarefa de busca sobre ementas. Logo, os trabalhos relacionados de RI serão usados para motivar as escolhas de métodos de RI para esta avaliação final das verbetações artificiais. No próximo Capitulo, será apresentada a metodologia adotada para alcançarmos o objetivo proposto anteriormente.

CAPÍTULO

4

METODOLOGIA

Neste Capítulo, são apresentados os vários componentes que compõem a metodologia desta pesquisa. Inicialmente, na Seção 4.1 é discutida a coleta dos dados utilizados durante o desenvolvimento do projeto. Em seguida, na Seção 4.2 descreve-se o pré-processamento dado aos documentos coletados. A Seção 4.3 aborda a metodologia empregada para geração automática de verbetações. Por fim, a Seção 4.4 descreve a avaliação final das verbetações geradas utilizando uma tarefa de RI.

4.1 Aquisição de Dados

Em 2022, como iniciativa em direção à transparência administrativa, o Supremo Tribunal de Justiça (STJ) tornou pública a plataforma Dados Abertos [1]. A plataforma corresponde a um domínio público para compartilhamento de decisões judiciais de diversos tribunais do Brasil, todas julgadas por ministros do STJ. Além da transparência, os Dados Abertos também foram planejados visando diminuir a utilização de *web scrappers* no portal oficial do STJ além de fomentar pesquisa e desenvolvimento de ferramentas baseadas em inteligência artificial que dependem da disponibilidade de dados como a desenvolvida por esta Dissertação de Mestrado.

O STJ é organizado em três seções. Cada uma delas é especializada em temas jurídicos específicos como impostos (Primeira Seção), comercio (Segunda Seção) e crimes no geral (Terceira Seção) (STJ, 23). Assim, os documentos disponibilizados abrangem uma grande variedade de temas do domínio jurídico brasileiro. Coletou-se um total de 726.384 documentos da plataforma (coleta feita em agosto de 2022) e foram analisadas as ementas dos processos coletados.

[1] <https://dadosabertos.web.stj.jus.br/>

4.2 Pré-processamento Inicial dos Dados

Os documentos presentes no portal Dados Abertos foram disponibilizados no formato *JavaScript Object Notation* (JSON). Tendo feito o *download* dos processos das diferentes fontes, os documentos foram concatenados em um único arquivo para a realização do próximo pré-processamento. Removemos então, documentos duplicados com base nos identificadores únicos de processo presentes nos metadados dos mesmos e com base no texto das decisões. No total, 144.518 documentos permaneceram após a etapa de deduplicação.

4.3 Geração de Verbetação

Nesta Seção, a metodologia empregada para geração automática de verbetação é discutida em maiores detalhes.

4.3.1 Preparação de Exemplos para Geração de Texto

A aplicação de técnicas modernas de PLN frequentemente requer que os dados tenham um tratamento especial antes de serem analisados. Tendo realizado o pré-processamento inicial, foi realizado um novo pré-processamento sobre os dados de-duplicados com foco no texto dos documentos. Partindo das ementas dos processos, removemos URLs dos textos das ementas e extraímos as verbetações e parágrafos enumerados nesta etapa, identificando frases compostas por caracteres em maiúsculo no início do texto das ementas.

Devido a simplicidade do método de separação de verbetações e parágrafos enumerados empregado, ementas cujas verbetações não seguiam a estrutura padrão (termos em caixa alta) foram descartadas neste pré-processamento. Assim, durante esta etapa, ementas sem verbetação (ou com falha na separação da verbetação) foram removidas, resultando em um conjunto final de 111.964 documentos. Desta forma, as verbetações originais (escritas por especialistas), passam a compor o conjunto de referência (ou conjunto ouro) a ser utilizado para avaliar a qualidade das verbetações geradas pelas estratégias estudadas tanto durante o treinamento supervisionado quanto na avaliação.

Concluímos o pré-processamento para geração de texto dividindo o corpus em três conjuntos: treino (70%), validação (10%) e teste (20%). Ao dividir os conjuntos, preservamos as proporções das origens (seções de julgamento) dos documentos. A divisão foi feita pois nosso objetivo é utilizar uma solução de DL supervisionada para a tarefa de geração de verbetação e a partição dos conjuntos será utilizada na validação cruzada. Na Tabela 1, são apresentadas as proporções para cada origem considerando todos os documentos coletados. Conforme mostrado nesta tabela, foram observadas proporções semelhantes para cada seção de julgamento.

Para visualizar o tamanho dos parágrafos enumerados e verbetações, a Tabela 2 apresenta estatísticas descritivas para tokens separados por espaço no conjunto de treino (78.375 exemplos).

Tabela 1 – Distribuição dos documentos por origem.

Origem	Exemplos
Primeira Seção	38,210 (34.12%)
Segunda Seção	33,881 (30.26%)
Terceira Seção	39,873 (35.61%)
Total	111,964

No total, o conjunto de treinamento possui em torno de 1,6 milhão de tokens. Os parágrafos enumerados têm uma média de 203,26 tokens e as verbetações têm uma média de 55,84 tokens. Conforme mostrado pelos valores de desvio padrão, os tamanhos dos parágrafos enumerados e verbetações variam dentro de uma margem considerável.

Tabela 2 – Estatísticas descritivas (média, desvio padrão e quartis) para tokens separados por espaço do conjunto de treino.

	Média	Desvio.	25%	50%	75%
Parágrafos enumerados	203.262	183.332	92	155	253
Verbetação	55.842	32.136	34	49	69

4.3.2 Transformers para Geração de Texto

Conforme mencionado anteriormente, a maioria dos termos apresentados nas verbetações não está diretamente no corpo da ementa (parágrafos enumerados) e, analisando os documentos do conjunto de validação, observamos que apenas em torno de 10% dos termos presentes nas verbetações estão localizado no corpo da ementa. Portanto, optamos por tratar a geração das palavras-chave discutidas como uma geração de sequência-a-sequência (ou texto-para-texto) utilizando a arquitetura *Transformer*.

Desta forma, o corpo das ementas (parágrafos enumerados apresentados na Seção 4.1) é usado como entrada para os modelos do *Transformer*, e as verbetações originais são usadas como as saídas esperadas. Assim, avaliamos quatro *Transformers* geradores de texto para a solução do problema. Dividimos os modelos avaliados em dois grupos, baseados na arquitetura dos mesmos: codificadores-decodificadores (utilizam blocos codificadores e decodificadores) e decodificadores (utilizam apenas blocos decodificadores). Os modelos escolhidos serão discutidos a seguir.

- **Codificadores-decodificadores:** Escolhemos dois modelos baseados no *framework* texto-para-texto T5: PTT5(CARMO *et al.*, 2020) e mT5(XUE *et al.*, 2020). O PTT5 foi pré-treinado no corpus *brWaC* (FILHO *et al.*, 2018) em Português Brasileiro. O corpus possui 2,7 bilhões de tokens e consiste em páginas da web obtidas utilizando filtros de alta qualidade. Utilizamos a versão *base* (220M de parâmetros) do PTT5. Quanto ao mT5, o modelo foi pré-treinado usando uma versão multilíngue (compreendendo 101 idiomas) do conjunto de dados C4 utilizado no pre-treino do T5(RADFORD *et al.*, 2019) original.

O corpus C4 multilíngue tem 6,3T tokens no total. Utilizamos a versão de 300M de parâmetros do modelo.

- **Decodificadores:** Avaliamos também dois modelos multilíngues *decoder-only*: OPT (ZHANG *et al.*, 2022) e BLOOM (SCAO *et al.*, 2022). Os modelos seguem a arquitetura *decoder-only* da família GPT e foram pré-treinados em coleções de texto baseadas na combinação de vários corpora de diferentes domínios textuais. Eles foram pré-treinados empregando corpora contendo 180B e 341B tokens, respectivamente. A respeito do tamanhos dos modelos, utilizamos a versão com 350M de parâmetros (OPT) e a versão com 560M de parâmetros (BLOOM).

Todos os tamanhos de modelo, tanto para modelos T5 quanto para os modelos que utilizam apenas decodificadores, foram escolhidos considerando a nossa disponibilidade de recursos computacionais. Optamos por não avaliar a geração de verbetações utilizando *few-shot* (ou *zero-shot*) empregando GPT-3 ou GPT-4 (RADFORD *et al.*, 2019; OPENAI, 2023), em razão do custo computacional e financeiro associado a esta tarefa.

4.3.3 Avaliação dos Textos Gerados

Para avaliarmos a qualidade dos textos gerados e comparar os diferentes modelos geradores de texto supervisionados estudados, empregamos uma métrica especializada. Sendo assim, utilizamos a pontuação *Bilingual Evaluation Understudy* (BLEU) (POST, 2018). Conforme discutido na Seção 2.2.2.1, a métrica estima a qualidade do texto gerado, comparando os textos gerados com referências geradas por humanos. Assim, comparamos as verbetações geradas com as verbetações de referência (escritas por especialistas) extraídas dos documentos descritos nas Seções anteriores. Realizamos a avaliação utilizando o pacote Python *sacrebleu* [2] e usamos decodificação gulosa (sem *beam-search*) para gerar as verbetações para avaliação. Para cálculo do BLEU, utilizamos os parâmetros padrão do pacote mencionando, empregando n-gramas de um a quatro tokens e o tokenizador *13a - Moses*.

Escolhemos utilizar apenas esta métrica, pois ela cumpre o objetivo de estimar a qualidade do texto gerado, sendo altamente correlacionada com avaliações humanas (PAPINENI *et al.*, 2002). Sendo assim, o seu cálculo é suficiente para comparar, de maneira quantitativa, os diferentes *Transformers* (geradores de verbetações) estudados. Além disto, optamos por não avaliar métricas baseadas em modelos pré-treinados como *BERTScore* (ZHANG *et al.*, 2019) ou *COMET* (REI *et al.*, 2020). Por falta de tempo, consideramos mais adequado (e prático) a avaliação de uma métrica que não dependa de um modelo de linguagem pré-treinado.

[2] <https://github.com/mjpost/sacrebleu>

4.3.4 Detalhamento do Treinamento Supervisionado

A seguir são descritos o processo de treinamento adotado e hiper-parâmetros utilizados para os modelos baseados em *Transformers*.

- **Codificador-Decodificador:** Considerando os modelos codificador-decodificador (PTT5 e mT5), foram utilizadas entradas e saídas de tamanhos 512 e 256 *sentencepiece* tokens (KUDO; RICHARDSON, 2018), respectivamente. Sequências de tokens mais curtas são preenchidas com tokens de *padding* e sequências mais longas são truncadas para o comprimento máximo estabelecido. Estes tamanhos de sequência foram suficientes para a maioria dos exemplos do conjunto de treino (88%). Ajustamos os modelos para a tarefa de geração de verbetação usando os seguintes hiper-parâmetros: taxa de aprendizado fixa de 1×10^{-3}, tamanho do lote igual a 256, decaimento de pesos igual a 1×10^{-2} e 20 épocas máximas de treinamento. Os hiper-parâmetros foram escolhidos com base nas implementações originais(CARMO *et al.*, 2020; XUE *et al.*, 2020). No entanto, observamos melhor desempenho no conjunto de validação ao usar uma taxa de aprendizado maior.

- **Decodificadores:** Para os modelos que utilizam apenas blocos decodificadores (OPT e BLOOM), combinamos as verbetações e os parágrafos enumerados em uma única entrada para o treino. Utilizamos um conjunto modificado de entradas porque a implementação de modelos decodificadores utilizada não permite sequências de entrada e saída de tamanhos diferentes para a tarefa de predição do próximo token. Assim, como uma etapa extra de pré-processamento, concatenamos as verbetações ao final das frases enumeradas e usamos os prefixos 'Documento:' e 'Verbetação:' para distinguir as seções durante o ajuste fino. Não investigamos mais variações de prefixos (também chamados de *prompts*) na literatura, pois esta tarefa foge do escopo deste projeto.

  Na Figura 9, é apresentado um exemplo de uma entrada preparada para o treinamento dos modelos OPT e BLOOM. Conforme discutido, não há uma 'saída esperada', os decodificadores visam apenas predizer os próximos tokens presentes na própria entrada. Não adicionamos os prefixos para treinamento do PTT5 e mT5, pois as métricas iniciais de treinamento (sem prefixos) já eram satisfatórias. Durante a avaliação, adicionamos apenas o prefixo 'Verbetação:' para a entrada dos modelos e extraímos o texto gerado após o prefixo para calcular as pontuações BLEU. Usamos um comprimento máximo de sequência de 768 e 616 para OPT e BLOOM, respectivamente. Novamente, sequências mais curtas são preenchidas e sequências mais longas são truncadas. Os valores foram escolhidos com base na disponibilidade de memória da GPU.

  Quanto aos hiper-parâmetros para treino dos modelos decodificadores, também seguimos valores semelhantes aos especificados para o pré-treinamento de modelagem de linguagem dos trabalhos originais: decaimento de pesos igual a 1×10^{-1}, tamanho do lote igual a

256, taxa de aprendizado de 1×10^{-4} com crescimento linear nas duas primeiras épocas e decaimento linear após e 20 épocas de treinamento máximo.

Figura 9 – Visualização da entrada fornecida para modelos compostos por apenas decodificadores. Os prefixos foram destacados em negrito.

"**Documento:** [texto dos parágrafos enumerados]. **Verbetação**: [texto da verbetação]"

Para todos os modelos estudados, usamos o objetivo de treinamento predição de próximo token (minimizando a função de perda de entropia cruzada) e monitoramos a métrica BLEU no conjunto de validação para interromper o treino após duas épocas sem melhoraria na métrica. Por fim, ajustamos todos os *Transformers* utilizando a biblioteca Python *huggingface*[3]. Foi utilizada uma GPU Tesla P100 com 16 GB de VRAM para todos os experimentos.

4.3.5 Comparação entre Verbetações Originais e Geradas

Partindo do modelo que atingiu a maior pontuação BLEU, realizamos análises comparativas entre o texto gerado e as verbetações originais. Para isto, realizamos comparações entre os tamanhos das verbetações geradas pelo melhor modelo e as verbetações originais. Além disto, comparamos a quantidade de palavras (tokens) copiadas e novas entre as verbetações considerando os parágrafos enumerados. Consideramos palavras como sendo "novas" (ou geradas), palavras que não constam nos parágrafos enumerados utilizados para a geração das verbetações artificiais.

4.4 Avaliação Utilizando RI

Para realizar a avaliação final dos textos gerados, nós os concatenamos ao seu documento original e simulamos um caso de uso real como sendo uma tarefa de RI. Para a geração das verbetações utilizadas nesta avaliação, empregamos tanto decodificação gulosa quanto decodificação usando amostragem. Os detalhes destas avaliações serão apresentados a seguir.

4.4.1 Formulação da Tarefa

Os documentos apresentados na Seção 4.1, além do texto da ementa, contêm metadados úteis, como o tema da decisão. Tais temas, também chamados de temas de recursos repetitivos, são representados por identificadores únicos que são mapeados para questões jurídicas comuns. Os documentos podem ter mais de um tema e existem mais de 1.000 temas diferentes listados no STJ [4]. Exemplos de temas são mostrados na Figura 10.

[3] <huggingface.co/>
[4] <https://scon.stj.jus.br/SCON/recrep/>

Tema 105: Ocorrência da decadência do direito de multar pelo estado, por falta de notificação do infrator, no prazo estabelecido em lei.

Tema 727: Possibilidade de técnicos de farmácia assumirem a responsabilidade técnica por drogaria, até a entrada em vigor da lei 13.021/2014.

Tema 1097: Necessidade de dupla notificação no caso de multa aplicada a pessoa jurídica proprietária de veículo fundamentada na ausência de indicação do condutor infrator.

Figura 10 – Exemplos de temas de recursos repetitivos, listados pelo STJ.

A partir destas informações, utilizamos a definição binária de relevância para formular uma tarefa de RI da seguinte forma: dado um documento de consulta Q, os documentos relevantes R a Q devem ser do mesmo tema que Q. Assim, esta formulação emula o caso de uso em que um advogado deseja buscar por documentos semelhantes a um documento em análise. A formulação apresentada é semelhante à utilizada por Ostendorff *et al.* (OSTENDORFF *et al.*, 2021), em que os autores criaram pares de relevância (documento consulta e documento relevante) usando decisões da Suprema Corte dos Estados Unidos a partir dos mesmos livros de casos ou categorias. Em resumo, utilizamos ementas para buscar ementas. Visando imitar o cenário real nos experimentos, empregamos ementas completas (contendo verbetações e parágrafos enumerados) a exceção do primeiro, o qual será discutido na Seção a seguir.

Considerando todos os documentos obtidos, apenas 801 têm anotações de tema (99,3% têm valores ausentes). Estes documentos foram removidos do conjunto de treinamento (treinamento de geradores de texto) e usados para compor pares de consulta e documentos relevantes nos experimentos posteriores. Filtramos os temas que apareceram pelo menos duas vezes, resultando em 482 consultas.

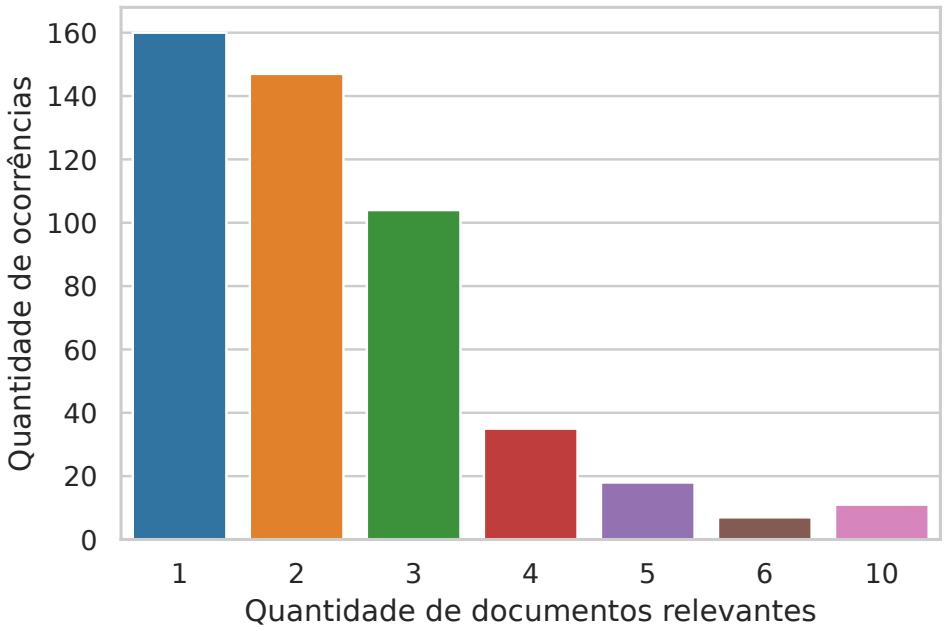


Figura 11 – Visualização do número de documentos relevantes (de mesmo tema) por documento consulta.

Na Figura 11 é apresentada uma visualização do número de documentos relevantes (mesmo tema) por documento de consulta. Conforme visto, a grande maioria das consultas tem entre 1 e 3 documentos relevantes (de mesmo tema). Além disso, no caso mais extremo, uma consulta pode ter até 10 documentos relevantes.

Por fim, para criar o corpus final de busca para a tarefa descrita, combinamos os 801 documentos com tema com o conjunto de teste apresentado na Seção 4.3.1, totalizando 23.194 documentos. Optamos por realizar tal combinação para tornar a tarefa de RI mais desafiadora uma vez que, no pior caso, os documentos podem apresentar falsos negativos prejudicando as métricas de RI. Os falsos negativos consistem em documentos com o mesmo tema da consulta, mas considerados não relevantes pois não possuem anotação.

4.4.2 Configuração Experimental

Conduzimos a avaliação usando RI através de quatro experimentos distintos. Os experimentos envolvem a utilização de método de decodificação simples (decodificação gulosa) e não determinísticos (decodificação com amostragem). Seus detalhes serão descritos a seguir.

1. **Comparação de documentos com e sem verbetações**: avaliamos a influência das verbetações na tarefa de RI descrita avaliando métricas de RI considerando documentos com e sem as suas verbetações originais (tanto para documentos de consulta quanto para documentos do corpus de busca). Este experimento foi realizado visando validar a utilidade das verbetações na estrutura das ementas, no contexto de uma tarefa de RI.

2. **Avaliação das verbetações geradas**: neste experimento, investigamos o efeito de usar as verbetações geradas artificialmente no lugar das originais. Concatenamos o texto gerado ao início dos documentos tanto da consulta quanto dos documentos do corpus. Desta forma, esperamos investigar se as verbetações artificiais trazem melhorias em relação ao não uso de verbetações. Neste experimento, utilizamos verbetações geradas utilizando decodificação gulosa sem *beam-search*.

3. **Combinação de verbetações**: com o objetivo de melhorar as verbetações originais, repetimos a avaliação de métricas de RI usando novas verbetações compostas pela concatenação das originais e das geradas. Como este procedimento pode ser usado para aprimorar documentos em corpus de pesquisa existentes (concatenando as verbetações geradas), o nosso interesse é investigar se a concatenação pode trazer melhorias nas métricas obtidas utilizando apenas as verbetações originais, já que há a potencial adição de mais termos discriminativos aos documentos. Assim como o experimento anterior, utilizamos decodificação gulosa para geração de verbetação.

4. **Experimentos com amostragem**: ao utilizar técnicas de decodificação baseadas em amostragem (FAN; LEWIS; DAUPHIN, 2018a), é possível gerar múltiplas verbetações para uma

mesma entrada. Tendo em mente esta ideia, dividimos os experimentos com amostragem em dois conjuntos:

a) *Investigando o efeito do número de repetições e do top-K*: partindo da hipótese de que é possível atingir desempenho compatível às verbetações originais utilizando decodificação com amostragem, investigamos o efeito de gerar múltiplas verbetações através da utilização da decodificação não-determinística. Geramos até 10 verbetações para cada entrada do corpus de busca, utilizando amostragem top-K, e as concatenamos as mesmas para gerar as métricas de RI. Repetimos 5 gerações com sementes de números aleatórios diferentes (1.000, 2.000, 3.000, 4.000 e 5.000) e agregamos os resultados para comparações. Também avaliou-se o efeito do K da amostragem top-K neste experimento, variando-se os valores de K entre 15, 50 e 100. Note que para este experimento, não estamos interessados em determinar o melhor número de repetições, nem o melhor valor K, e sim investigar o seu efeito na tarefa de RI proposta.

b) *Comparação entre Decodificação Gulosa e com Amostragem*: repetimos os dois experimentos anteriores (avaliação das verbetações geradas e combinação de verbetações), porém utilizando verbetações com amostragem no lugar das verbetações geradas com decodificação gulosa. Neste caso, o objetivo é investigar o quanto as métricas geradas com verbetações geradas empregando decodificação com amostragem diferem das geradas utilizando verbetação gulosa. Para esta última avaliação, utilizou-se apenas uma verbetação, obtida através da geração com amostragem (utilizando $K = 15$), em virtude dos resultados do experimento anterior.

Os dois últimos experimentos (combinação de verbetações e experimentos com amostragens) foram inspirados no trabalho *doc2query* (NOGUEIRA *et al.*, 2019). Neste trabalho, para cada documento de uma coleção, os autores geraram várias consultas relacionadas ao conteúdo do documento usando um modelo de sequência-para-sequência. As consultas são então concatenadas aos documentos de entrada com o objetivo de melhorar métricas de RI.

Para os experimentos com amostragem, utilizamos *contrastive-search* (SU *et al.*, 2022) com $alpha = 0.6$. Os valores de *alpha* (utilizado para *contrastive-search*) e K (utilizado para amostragem) escolhidos como parâmetros foram inspirados nos usados pelos trabalhos de (NOGUEIRA *et al.*, 2019; SU *et al.*, 2022). Por fim, quanto a decodificação gulosa, optamos por avalia-la sem a utilização de *beam-search*. Como a decodificação gulosa é a técnica de decodificação mais simples avaliada, optamos por não aumentar sua complexidade empregando *beam-search* para acentuar suas diferenças em relação a outra estratégia de decodificação estudada (top-K).

4.4.3 Métodos Avaliados

Avaliamos dois métodos tradicionais de RI para a tarefa proposta: TF-IDF e BM25 (ROBERTSON; WALKER, 1999). Os métodos foram escolhidos pois ainda fazem parte de mecanismos de busca populares (como Lucene[5]), são recorrentemente utilizados como *baselines* competitivas em RI (ROSA *et al.*, 2021b; PRADEEP *et al.*, 2020), e, como mostram trabalhos anteriores (LIMA; COSTA; ARAÚJO, 2021; MANDAL *et al.*, 2021), os métodos de representação esparsa tendem a ter melhor desempenho em tarefas semelhantes no domínio jurídico. A seguir serão apresentados mais detalhes de suas execuções.

4.4.4 Preparação de Exemplos para a Tarefa de RI

Para a utilização dos métodos esparsos de RI foram adicionadas novas etapas de limpeza e pré-processamento, sendo utilizadas em conjunto das descritas na Seção 4.2. Para ambos os métodos, os documentos foram tokenizados, utilizando lematização, e removendo *stop-words* e pontuações. Considerando TF-IDF, usamos n-gramas variando de 1 a 3 e um tamanho de vocabulário de 10.000 considerando os tokens que aparecem pelo menos 3 vezes no corpus de busca (documentos com anotação de tema e conjunto de teste). O pré-processamento foi feito usando *spacy*[6] e *sklearn*[7].

4.4.5 Ordenação dos Documentos

Ao utilizar representações TF-IDF para realizar a recuperação, calculamos a similaridade de cosseno entre consultas e documentos e ordenamos os documentos do corpus conforme suas semelhanças (ordem decrescente). Por outro lado, o BM25 usa o *Probality Ranking Principle* para estimar a relevância de um exemplo para uma consulta (CRESTANI *et al.*, 1998). Assim, os documentos são ordenados a partir de suas relevância estimadas. Utilizamos a implementação do TF-IDF e da métrica de distância (similaridade de cosseno) feitas pela biblioteca *sklearn*. Em relação ao BM25, utilizamos a implementação e parâmetros padrão da biblioteca *rank-bm25*[8].

4.4.6 Métricas Avaliadas

Para avaliar quantitativamente a tarefa de busca e, indiretamente, a qualidade das verbetações artificiais, avaliamos a execução dos métodos mencionados em termos de métricas tradicionais de RI. Assim, avaliamos o *Mean Reciprocal Rank* (MRR), *Normalised Discounted Cumulative Gain* (NDCG), Revocação, Precisão e *Mean Average Precision* (MAP).

Apenas os 10 primeiros documentos foram considerados para o cálculo das métricas. Escolhemos este limiar para simular um cenário restritivo, em que o usuário apenas aceita

[5] <https://lucene.apache.org/>
[6] <https://spacy.io/>
[7] <https://scikit-learn.org>
[8] <https://pypi.org/project/rank-bm25/>

investigar até 10 documentos para sua busca. Segundo (RUSSELL-ROSE; CHAMBERLAIN; AZZOPARDI, 2018), profissionais do Direito tendem a analisar, em sua grande maioria, até 50 documentos em suas consultas. Logo, estamos avaliando um cenário ainda mais desafiador do que o visualizado pelos autores.

Para a geração das métricas, e comparações utilizando testes de hipótese, foi utilizada a biblioteca *ranx* (BASSANI, 2022).

4.5 Considerações Finais

No presente Capítulo foram apresentadas as diferentes etapas que compõem a metodologia empregada neste trabalho, além de discutir como as mesmas se relacionam. Foram discutidos a coleta dos documentos utilizados nos experimentos, a metodologia empregada para geração das verbetações utilizando *Transformers* e a avaliação das verbetações (originais e artificiais). A avaliação da semelhança entre as verbetações geradas e as originais é feita por meio da métrica de geração de texto BLEU. A avaliação final associada a tarefa de RI descrita, é feita utilizando as métricas de RI descritas anteriormente. Por fim, o próximo Capítulo apresentará discussões baseadas nos resultados obtidos.

CAPÍTULO

5

RESULTADOS E DISCUSSÕES

Neste Capítulo são apresentados e discutidos os resultados do treinamento de geração de texto e avaliação usando IR. Os resultados obtidos do uso de modelos geradores de texto são comparados na Seção 5.1. Em seguida, os resultados obtidos utilizando RI, para todos os experimentos descritos na Seção 4.4, são apresentados e discutidos na Seção 5.2.

5.1 Avaliação da Geração de Texto

Nesta Seção são apresentados e discutidos os resultados da geração automática de verbetações utilizando os modelos supervisionados de *deep learning* descritos anteriormente.

5.1.1 Pontuações BLEU

Tabela 3 – Pontuações BLEU obtidas para cada *Transformer* avaliado nos conjuntos de validação e teste.

	BLEU	
	Validação	Teste
PTT5	37.194 ± 0.752	37.547 ± 0.782
mT5	30.887 ± 0.468	31.126 ± 0.413
BLOOM	15.895 ± 1.416	16.174 ± 3.645
OPT	11.032 ± 1.507	12.901 ± 2.046

A Tabela 3 mostra as pontuações BLEU obtidas para cada modelo avaliado. Repetimos o treinamento com cinco sementes de números aleatórios diferentes (1.000, 2.000, 3.000, 4.000 e 5.000) para gerar as métricas apresentadas. Conforme observado, o modelo PTT5 obteve as maiores pontuações BLEU para geração de verbetação. Apesar de ter mais parâmetros do que o PTT5 (300M em comparação a 220M) e ter sido treinado em um corpus maior (6,3T de tokens em comparação a 2,7B), a versão multilíngue do T5 obteve pontuações BLEU inferiores nos conjuntos de validação e teste. Ao trabalhar com *Transformers* e transferência de aprendizado

em PLN, modelos pré-treinados para o idioma-alvo (idioma da tarefa) tendem a ter desempenho superior a versões multilíngues dos mesmos (CARMO *et al.*, 2020; SOUZA; NOGUEIRA; LOTUFO, 2020; ROSA *et al.*, 2021a) e observamos o mesmo padrão em nossos experimentos. O par de melhores modelos (PTT5 e mT5) alcançou pontuações acima de 30% BLEU e a diferença entre suas pontuações BLEU é estatisticamente significativa de acordo com um teste-T pareado ($p < 0.05$).

Os modelos decodificadores avaliados tiveram um desempenho consideravelmente inferior quando comparados aos modelos baseados em T5, alcançando menos da metade das pontuações BLEU. Apontamos quatro possíveis explicações para seu desempenho. Em primeiro lugar, os modelos com maior número de parâmetros sofreram mais com a falta de dados para treinamento. Em segundo lugar, ao limitar o número de tokens de entrada (tamanho de sequência) dos modelos para acomodar as entradas na memória da GPU, o truncamento pode ter levado à perda de informação. Em terceiro lugar, ambos os modelos decodificadores foram pré-treinados em corpora multilíngues, nos quais textos em português representam apenas pequenas frações do total. No entanto, observamos que o modelo mT5 teve desempenho melhor embora seja multilíngue. Por fim, seguindo os resultados de (RADFORD *et al.*, 2019) e observando o desempenho dos modelos codificador-decodificador (PTT5 e mT5), os resultados sugerem que o componente codificador é importante para a geração de verbetações.

Considerando os resultados apresentados e discutidos, utilizamos verbetações geradas pelo modelo PTT5 para as próximas avaliações. Exemplos de verbetações geradas pelo melhor modelo, utilizando decodificação gulosa, são apresentados no Apêndice A.1. Ao avaliar-se qualitativamente, nota-se que com valores de BLEU próximos a 40%, as verbetações artificiais não apresentam erros ortográficos e léxicos, captam o estilo de escrita das verbetações e são bastante similares às verbetações originais escritas por humanos. Logo, com os resultados da geração textual podemos concluir que os resultados da geração de texto empregando *Transformers* foram positivos, apesar do pequeno número de exemplos do conjunto de treinamento (menos de 100K).

Tal sucesso, considerando o tamanho do conjunto de treino, pode ser atribuído principalmente ao ajuste do modelo de linguagem ao idioma Português Brasileiro tendo em vista a diferença do modelo vencedor PTT5 aos demais modelos multilinguais.

5.1.2 Comparação entre Verbetações Originais e Geradas

A análise da métrica BLEU permite estimar a quantidade de termos em comum entre as verbetações originais (referências) e as verbetações geradas ao comparar n-gramas contidas em ambas. A seguir, iremos dar continuidade à comparação entre as verbetações, analisando as verbetações originais e as verbetações geradas com decodificação gulosa para o conjunto de teste. A Figura 12 apresenta uma comparação entre o número de tokens das verbetações originais e das verbetações geradas. É possível observar que, por mais que as distribuições apresentadas

pelos dois histogramas sejam semelhantes, as verbetações geradas tiveram uma concentração maior de exemplos abaixo de 60 tokens. O efeito disto é perceptível na média, uma vez que a média de tokens separados por espaço das verbetações geradas é inferior à média dos tokens apresentados pelas verbetações originais (42.34 comparado a 48.28). Sendo assim, identificamos que as verbetações geradas com decodificação gulosa tendem a ter comprimento menor em tokens do que as verbetações originais.

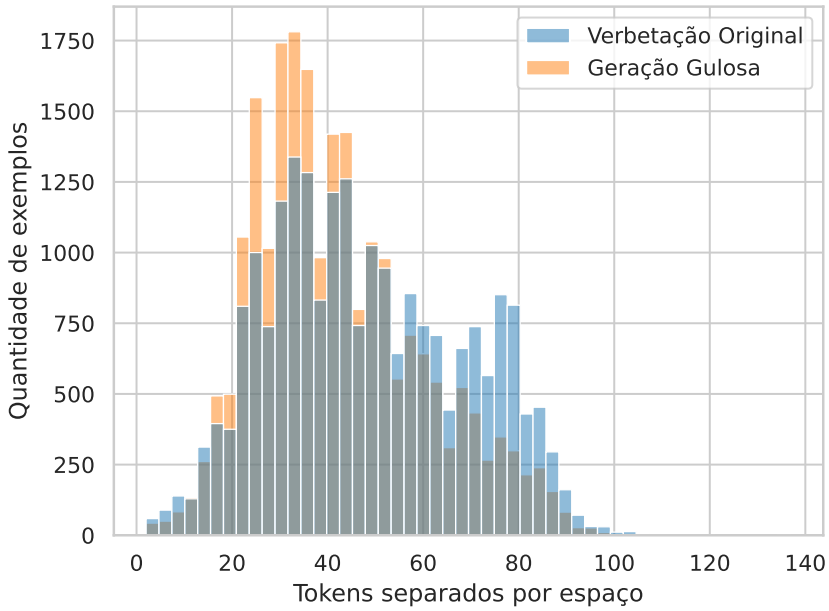


Figura 12 – Histogramas de comparação entre a quantidade de tokens das verbetações originais e das verbetações geradas.

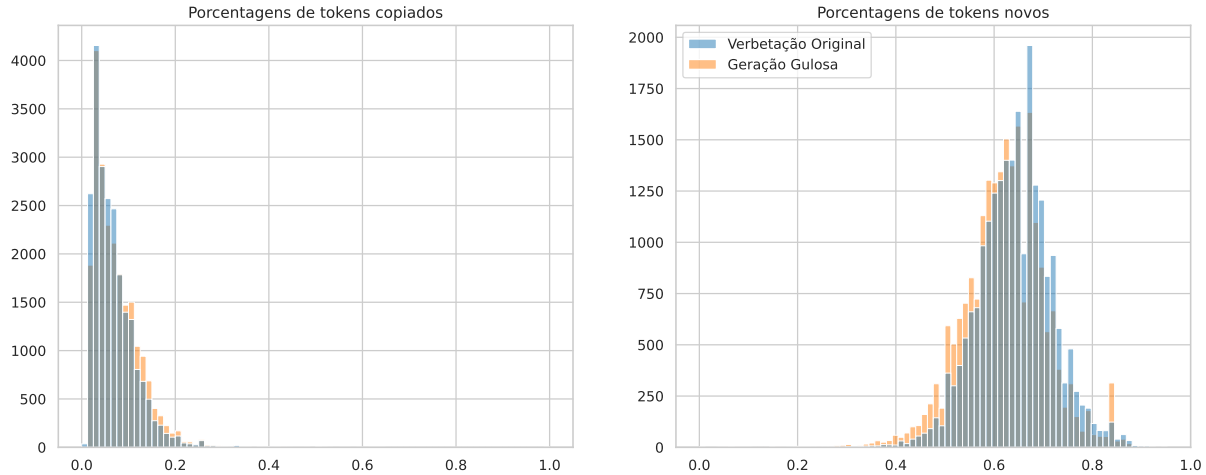


Figura 13 – Comparação entre porcentagens de palavras copiadas e geradas (novas), entre as verbetações geradas e as verbetações originais.

Em seguida, a Figura 13 apresenta novos histogramas comparando as porcentagens de palavras copiadas e geradas (novas) em relação aos parágrafos enumerados utilizados como entrada para geração de texto. É visível que as verbetações geradas e as verbetações originais

possuem distribuições de palavras copiadas e novas similares. Ao analisar as médias, e outras estatísticas descritivas, de palavras copiadas e geradas também observamos valores próximos conforme mostra a Tabela 4.

Tabela 4 – Estatísticas descritivas (média, desvio padrão e quartis) para a porcentagem de palavras copiadas e geradas considerando as verbetações geradas e originais.

	Tokens	Média	Desvio Padrão	25%	50%	75%
Verbetações Originais	Palavras Copiadas	0.067	0.047	0.033	0.055	0.089
	Palavras Geradas	0.641	0.076	0.594	0.644	0.690
Verbetações Geradas	Palavras Copiadas	0.067	0.047	0.033	0.055	0.091
	Palavras Geradas	0.631	0.080	0.581	0.631	0.681

5.2 Avaliação Utilizando RI

Os resultados obtidos para os quatro experimentos de IR descritos na Seção 4.4 são apresentados e discutidos nesta seção. As métricas apresentadas em todas as tabelas consistem nas médias obtidas para cada métrica considerando todas as 482 consultas. Além disto, foram realizados teste-T pareados para investigar a probabilidade de ocorrência da hipótese nula (não haver diferença significativa entre as médias) para cada comparação feita. Note que para os experimentos desta Seção, não estamos interessados em comparar os métodos de RI escolhidos, e sim investigar como os mesmos se comportam ao utilizar as verbetações originais ou geradas através de *deep learning*.

5.2.1 Comparação de Documentos Com e Sem Verbetações

Tabela 5 – Métricas de RI obtidas para cada modelo avaliado, avaliando a influência das verbetações originais. '*' indicam diferenças estatisticamente significativas, de acordo com um teste-T pareado (p-valor < 0.05).

	TF-IDF				
	MRR@10	**NDCG@10**	**R@10**	**P@10**	**MAP@10**
Sem verbetações	0.806	0.745	0.790	0.191	0.691
Verbetações originais	0.825*	0.780*	0.832*	0.201*	0.729*
	BM25				
Sem verbetações	0.819	0.805	0.878	0.212	0.754
Verbetações originais	0.879*	0.858*	0.916*	0.222*	0.815*

A Tabela 5 compara as métricas obtidas ao ordenar documentos com e sem as verbetações originais empregando os modelos TF-IDF e BM25. Podemos observar que o uso de verbetações traz melhorias para todas as métricas avaliadas, com maiores ganhos visíveis para o modelo BM25. Os resultados já eram esperados, uma vez que os métodos esparsos avaliados se beneficiam por terem mais termos discriminatórios no documento a ser indexado. Desta forma, os resultados indicam que o uso das palavras-chave tem, de fato, uma influência positiva na tarefa

de busca investigada. Observamos maior ganho na métrica R@10 (4,2 pontos percentuais) para TF-IDF e na MAP@10 (6,1 pontos percentuais) para BM25. Além disto, a diferença nas métricas (para ambos os modelos avaliados) é estatisticamente significativa para todas as métricas.

Vale a pena mencionar que, ao criar um corpus de pesquisa contendo o conjunto de teste (Seção 4.4), corremos o risco de introduzir falsos negativos no corpus, uma vez que não podemos garantir que os processos com temas ausentes do conjunto não possuam temas correspondentes às consultas. Apesar desta desvantagem, ainda observamos ganhos em todas as métricas avaliadas. Este fator também deve ser considerado ao interpretar os resultados dos demais experimentos.

5.2.2 Avaliação das Verbetações Geradas

Tabela 6 – Métricas obtidas para cada modelo de recuperação, ao utilizar documentos sem verbetação e utilizando as verbetações geradas com decodificação gulosa. '*' indicam diferenças estatisticamente significativas, de acordo com um teste-T pareado (p-valor < 0.05).

	TF-IDF				
	MRR@10	**NDCG@10**	**R@10**	**P@10**	**MAP@10**
Sem verbetações	0.806	0.745	0.790	0.191	0.691
Geração gulosa	0.822*	0.770*	0.815*	0.196*	0.718*
	BM25				
Sem verbetações	0.819	0.805	0.878	0.212	0.754
Geração gulosa	0.854*	0.819*	0.877	0.211	0.768

A Tabela 6 apresenta a avaliação das palavras-chave geradas com decodificação gulosa. No geral, ao adicionar as verbetações artificiais aos documentos, observamos melhorias em quase todas as métricas para ambos os cenários. Considerando o método TF-IDF, observamos diferença estatisticamente significativa em todas as métricas avaliadas.

Em relação ao método BM25, observamos ganhos em quase todas as métricas. Entretanto, a diferença nas métricas é estatisticamente significativa apenas para as métricas MRR@10 e NDCG@10. Considerando os 10 primeiros documentos ordenados, embora houveram melhorias para o TF-IDF, os resultados podem indicar que o modelo BM25 foi mais sensível a falsos positivos (falsos relevantes) e falsos negativos (falsos não relevantes), introduzidos corpus de busca por eventuais verbetações ruidosas geradas pelo PTT5. Sendo assim, observamos quase nenhuma modificação nas métricas baseadas em revocação e precisão.

As métricas obtidas para ambos os modelos foram inferiores às métricas obtidas utilizando as verbetações originais, apresentadas na Tabela 5. Tal fato era esperado, pois os resultados obtidos utilizando as verbetações originais atuam como um limite superior para as métricas deste experimento. No entanto, o uso das verbetações artificiais trouxe melhorias para ambos os métodos de recuperação quando comparado ao não uso de nenhuma verbetação, mostrando-nos a potencial viabilidade da proposta de geração automática das palavras-chave.

5.2.3 Combinação de Verbetações

Tabela 7 – Métricas de RI obtidas ao combinar verbetações originais e geradas com decodificação gulosa. '*' indicam diferenças estatisticamente significativas, de acordo com um teste-T pareado (p-valor < 0.05).

	TF-IDF				
	MRR@10	**NDCG@10**	**R@10**	**P@10**	**MAP@10**
Verbetação original	0.825	0.780	0.832	0.201	0.729
Combinação gulosa	0.838*	0.793*	0.839	0.202	0.743*
	BM25				
Verbetação original	0.879	0.858	0.916	0.222	0.815
Combinação gulosa	0.880	0.857	0.909	0.219	0.816

A Tabela 7 apresenta os resultados do terceiro experimento. Ao combinar os termos-chave originais e gerados, observamos pequenas melhorias em todas as métricas para a recuperação TF-IDF. Nesse caso, observamos diferenças significativas para as métricas MRR@10, NDCG@10 e MAP@10. Para as métricas R@10 e P@10, os valores permaneceram quase os mesmos.

A respeito do modelo BM25, não foram observadas melhorias visíveis. Nesta avaliação, não foram obtidas diferenças estatisticamente significativas para nenhuma métrica. Combinar a verbetação original e as geradas não trouxe nenhuma melhoria em relação ao uso de apenas a original. A mesma explicação apresentada no experimento anterior pode ser utilizada para justificar a ausência de ganhos nas métricas para o modelo.

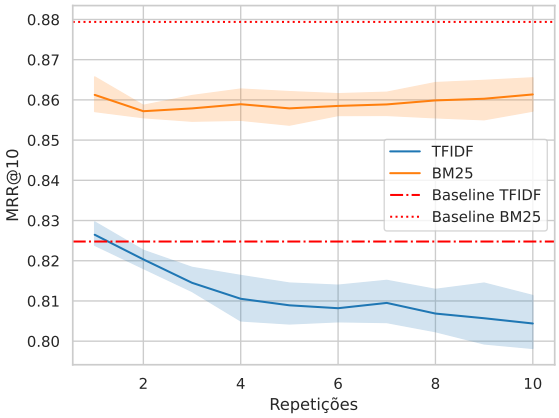
Embora não tenha havido melhorias para o modelo BM25, a combinação de verbetações proposta ainda pode ser utilizada para o método TF-IDF caso haja interesse em maximizar as métricas que obtiveram diferenças significativas.

5.2.4 Experimentos com Amostragem

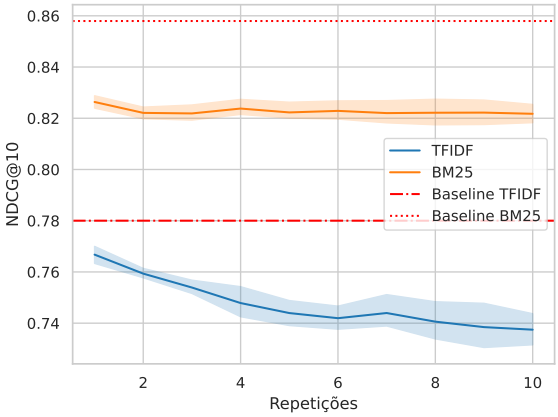
Nesta Subseção serão apresentados os resultados obtidos para os experimentos com geração de verbetações utilizando decodificação não-determinística.

5.2.4.1 Investigando o Efeito do Número de Repetições e do Top-K

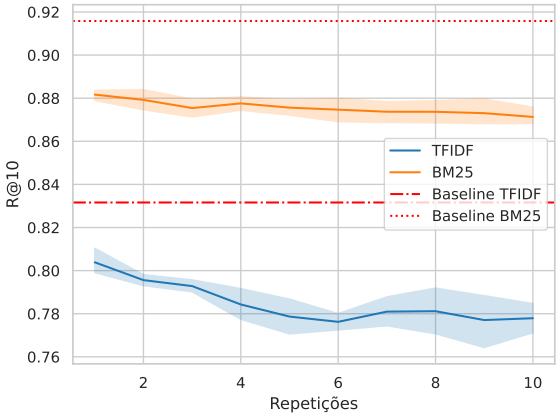
As Figuras 14, 15 e 16 ilustram as métricas obtidas ao gerar até 10 variações para cada entrada do corpus de busca para diferentes valores K. Ao realizar este experimento, a expectativa era de se observar um crescimento logarítmico na medida em que mais verbetações fossem concatenadas as entradas conforme observado por (NOGUEIRA *et al.*, 2019). Entretanto, este resultado não foi observado em nenhuma das métricas avaliadas. Ao contrário do esperado, nos piores casos, houve um decaimento nas métricas na medida em que novas repetições eram adicionadas aos textos de entrada. O decaimento é mais perceptível para o método TF-IDF, com reduções entre 2% e 3% em todas as métricas observadas. Os comportamentos mencionados foram observados para todos os valores de K avaliados.



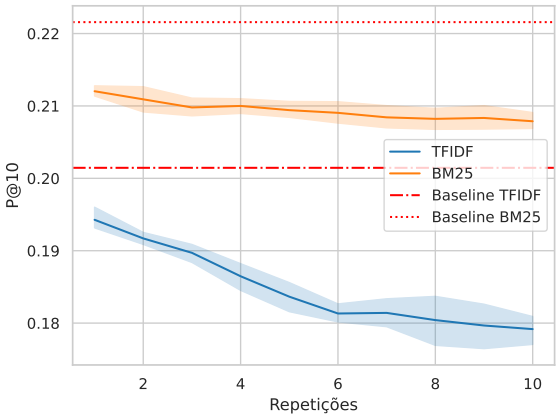
(a) MRR.



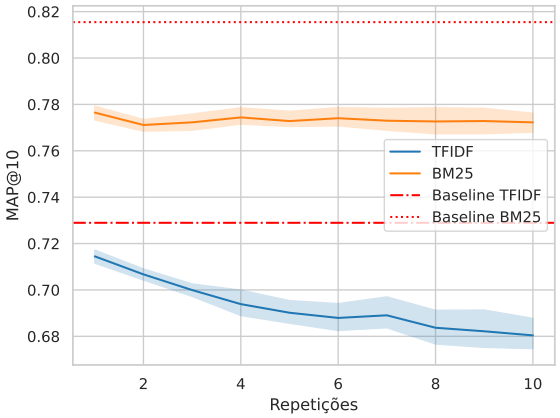
(b) NDCG.



(c) Revocação.



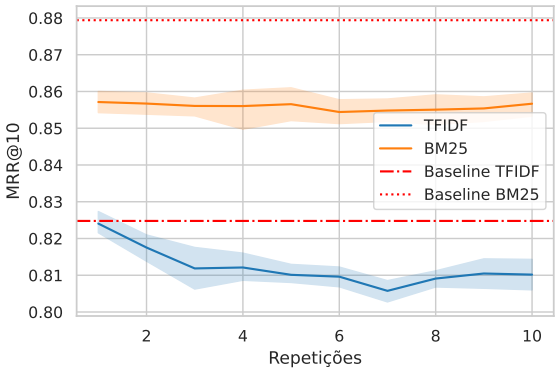
(d) Precisão.



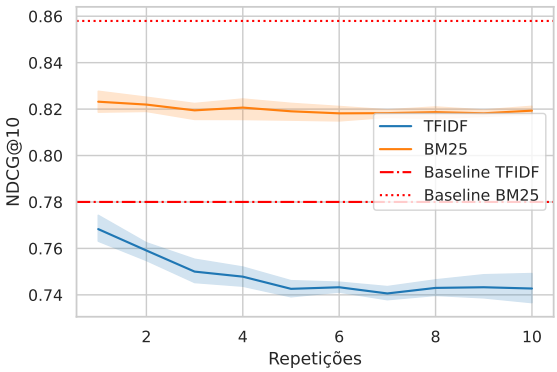
(e) MAP.

Figura 14 – Métricas obtidas para geração de verbetações utilizando amostragem utilizando amostragem **top-15**. O eixo horizontal mostra o número de variações de verbetação concatenadas à entrada para a geração. As linhas trastejadas indicam os valores obtidos ao utilizar as verbetações originais na tarefa de busca proposta. As regiões sombreadas indicam os intervalos de 95% de confiança considerando as 5 repetições da geração com decodificação utilizando amostragem.

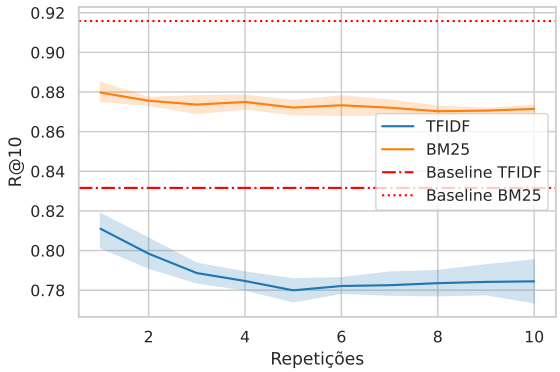
Nota-se que a formulação dos experimentos de (NOGUEIRA *et al.*, 2019) é diferente da empregada neste. Enquanto os autores avaliaram um cenário tradicional de RI, onde existem



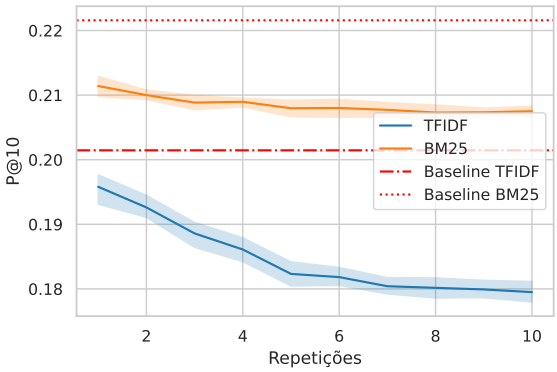
(a) MRR.



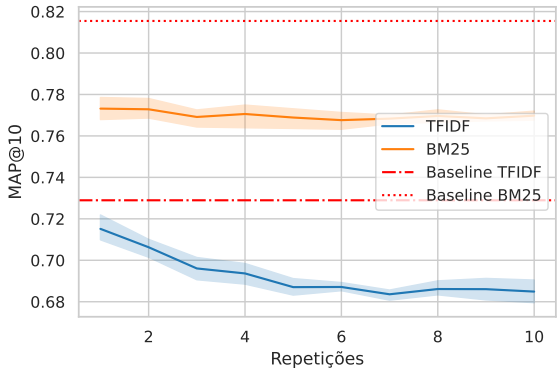
(b) NDCG.



(c) Revocação.



(d) Precisão.



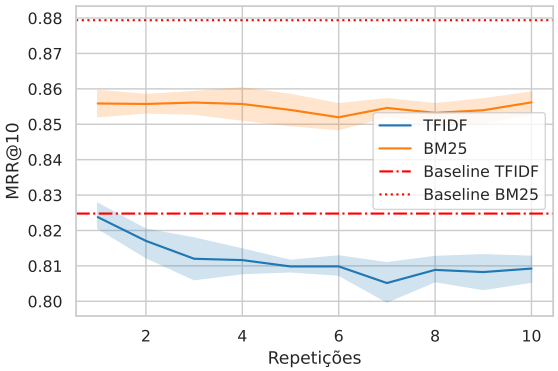
(e) MAP.

Figura 15 – Idem Figura 14, porém considerando amostragem **top-50**.

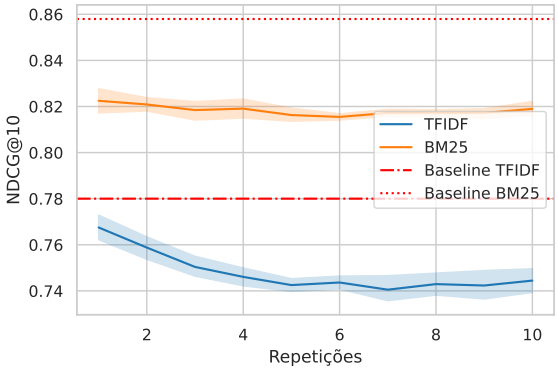
consultas genéricas formuladas em linguagem natural oriundas do buscador *Bing* [1], as consultas usadas nesta proposta são compostas por documentos inteiros e de um domínio específico (jurídico).

Retomando as discussões sobre o experimento com amostragem, ao aumentar os valores K, temos por consequência um aumento na variabilidade do texto gerado uma vez que os tokens a serem preditos são escolhidos dentro de um conjunto maior. Também era esperado um efeito positivo nas métricas em virtude da possibilidade da adição de mais termos discriminativos na

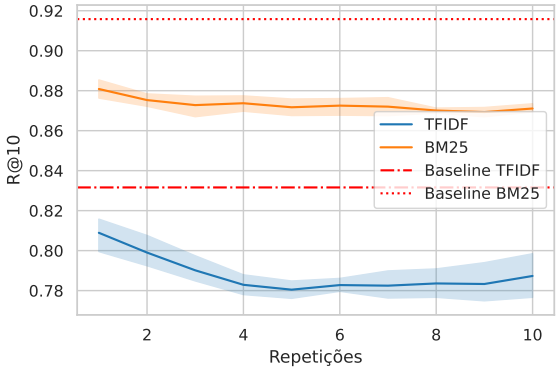
[1] <https://www.bing.com/>



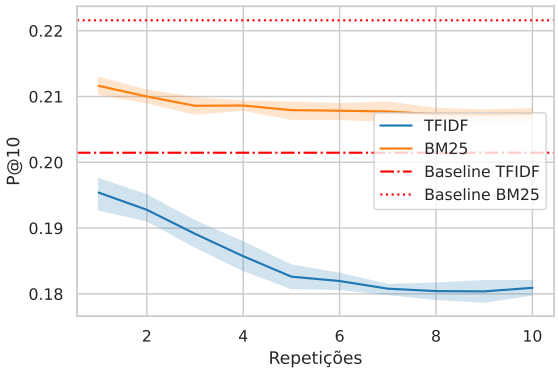
(a) MRR.



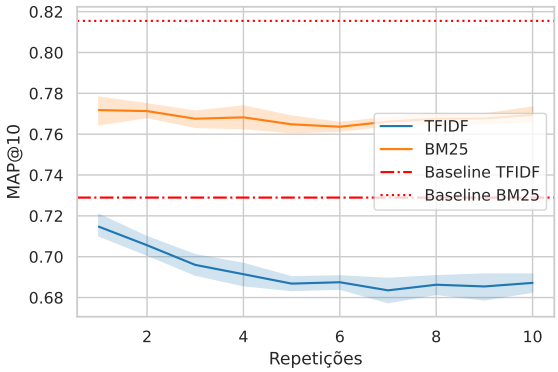
(b) NDCG.



(c) Revocação.



(d) Precisão.



(e) MAP.

Figura 16 – Idem Figura 14, porém considerando amostragem **top-100**.

geração. No entanto, o que observou-se foram métricas dentro dos mesmos intervalos ao se aumentar o valor de K de 15 a 100 (Figuras 14, 15 e 16). Logo, não há evidências que suportem a escolha de valores K superiores a 15.

Quanto aos modelos de RI avaliados, variando-se o número de repetições e valores K, nota-se que o método BM25 teve menor variância nas suas métricas, entretanto não obteve nenhuma melhora visível ao incorporar mais variações. Ao analisar as métricas do TF-IDF, observa-se que a performance do método deteriorou em todas as métricas ao adicionar novas repetições. Para ambos os algoritmos de RI estudados, as melhores métricas foram observadas

ao utilizar uma única verbetação proveniente da geração não-determinística. Os resultados observados para os dois métodos serão discutidos mais a fundo na Seção 5.2.4.3.

5.2.4.2 Comparação entre Decodificação Gulosa e com Amostragem

Tabela 8 – Repetição do primeiro experimento (verbetações artificiais no lugar das originais) utilizando verbetações geradas com amostragem. Caracteres sobrescritos indicam diferenças estatisticamente significativas, de acordo com um teste-T pareado (p-valor < 0.05).

	MRR@10	NDCG@10	R@10	P@10	MAP@10
TF-IDF					
a) Sem verbetações	0.806	0.745	0.790	0.191	0.691
b) Geração gulosa	0.822^{a}	0.770^{a}	0.815^{a}	0.196^{a}	0.718^{a}
c) Geração com amostragem	0.828^{a}	0.768^{a}	0.810^{a}	0.195^{a}	0.712^{a}
BM25					
a) Sem verbetações	0.819	0.805	0.878	0.212	0.754
b) Geração gulosa	0.854^{a}	0.819^{a}	0.877	0.211	0.768
c) Geração com amostragem	0.863^{a}	0.830^{ab}	0.890	0.213	0.779^{a}

Tabela 9 – Repetição do segundo experimento (combinação de verbetações) utilizando verbetações geradas com amostragem. Caracteres sobrescritos indicam diferenças estatisticamente significativas, de acordo com um teste-T pareado (p-valor < 0.05).

	MRR@10	NDCG@10	R@10	P@10	MAP@10
TF-IDF					
a) Verbetação original	0.825	0.780	0.832	0.201	0.729
b) Combinação gulosa	0.838^{a}	0.793^{a}	0.839	0.202	0.743^{a}
c) Combinação com amostragem	0.844^{a}	0.797^{a}	0.847	0.203	0.746^{a}
BM25					
a) Verbetação original	0.879	0.858	0.916	0.222^{b}	0.815
b) Combinação gulosa	0.880	0.857	0.909	0.219	0.816
c) Combinação com amostragem	0.884	0.862	0.917	0.221	0.820

As Tabelas 8 e 9 apresentam os resultados obtidos ao repetir os experimentos anteriores utilizando verbetações geradas com amostragem. Em ambos os experimentos, utilizamos uma única verbetação gerada através da amostragem. Optamos por utilizar apenas uma verbetação uma vez que não foi observado ganho nas métricas ao utilizar múltiplas verbetações e para utilizar uma configuração semelhante à utilizada nos experimentos com verbetação gerada pelo método guloso (usando uma única verbetação gerada). Além disto, utilizamos $K = 15$ em virtude da ausência de melhora nas métricas ao aumentar seu valor.

Considerando a Tabela 8, para os experimentos com verbetações geradas no lugar das originais, observamos resultados semelhantes aos obtidos para as verbetações geradas nos experimentos anteriores. As diferenças significativas observadas anteriormente para o TF-IDF também foram observadas para as verbetações geradas com amostragem para as mesmas métricas. Para o método BM25, novamente, não foram observadas diferenças significativas para as métricas

precisão e revocação. As justificativas apresentadas anteriormente também podem ser aplicadas para os experimentos com amostragem. No entanto, foi observada diferença estatisticamente significativa para o MAP@10, indicando que houve uma melhora na precisão ao considerar limiares de corte entre 1 e 10.

Agora, considerando a Tabela 9, os experimentos combinando as verbetações obtiveram resultados semelhantes aos observados no experimento com verbetações gulosas para ambos os modelos, apresentando apenas pequenas variações nos valores obtidos para as métricas. Novamente, observamos ganhos significativos em apenas duas métricas para o TF-IDF e, considerando o BM25, nenhum ganho significativo foi observado.

Para finalizar as discussões a cerca dos experimentos com amostragem, resta comparar os resultados obtidos com a utilização deste método com o método de geração baseado em decodificação gulosa. Conforme observado pelos resultados das Tabelas 8 e 9, na maioria dos casos, as verbetações geradas com amostragem obtiveram pequenos incrementos em relação às verbetações gulosas. No entanto, considerando um limiar de 5% para o teste de hipótese, apenas foram observados p-valores inferiores a 0.05 entre as decodificações no primeiro experimento, em uma única métrica (NDCG@10) e para apenas o método BM25. Desta forma, e considerando as demais métricas, não há evidência suficiente para ignorar a hipótese nula (métricas possuírem mesma média) observando as comparações entre as métricas das abordagens de decodificação. Logo, não há como justificar o uso de uma decodificação no lugar da outra tendo em vista os experimentos executados.

5.2.4.3 *Investigação dos Resultados de Amostragem*

Ao decorrer desta Seção, foram apresentados os resultados da aplicação de geração não-determinística de verbetações empregando amostragem top-K. Embora houvesse uma expectativa de melhora nas métricas de RI empregando amostragem, ao concatenar-se múltiplas variações de ementas nos documentos de consulta e do corpus de busca obtivemos resultados opostos. Além disto, ao avaliar valores maiores de K, a maior variabilidade no texto também não trouxe resultados positivos para as métricas de RI estudadas.

Buscando entender o motivo destes resultados abaixo do esperado, foi realizada uma inspeção de exemplos de verbetações geradas com amostragem para todos os experimentos realizados. Exemplos de verbetações geradas através de amostragem são apresentados no Apêndice A.2. A partir desta análise qualitativa, foram observados alguns padrões que podem ajudar a justificar os resultados dos experimentos.

Analisando verbetações geradas por amostragem top-15, é possível observar que o efeito principal da utilização da amostragem é a geração de paráfrases entre as diferentes gerações. Tal fato é facilmente perceptível na Figura 17, observando os temos “SERVIDOR PÚBLICO” e “AGENTE PÚBLICO”. Outro efeito interessante observado é a expansão de siglas como “CF/1998” para “CONSTITUIÇÃO FEDERAL DE 1998”. Outro efeito comum observado é

Original: CONFLITO NEGATIVO DE COMPETÊNCIA. JUSTIÇA COMUM ESTADUAL E JUSTIÇA DO TRABALHO. AÇÃO DE OBRIGAÇÃO DE FAZER C/C COBRANÇA. CONTRIBUIÇÃO SINDICAL DE SERVIDORES PÚBLICOS MUNICIPAIS. ART. 114, III, DA CF/1988 COM REDAÇÃO DADA PELA EC 45/2004. COMPETÊNCIA DA JUSTIÇA DO TRABALHO. PRECEDENTES.

1) CONFLITO NEGATIVO DE COMPETÊNCIA. CONTRIBUIÇÃO SINDICAL. SERVIDOR PÚBLICO ESTATUTÁRIO. ART. 114, III, DA CONSTITUIÇÃO FEDERAL DE 1988. EMENDA CONSTITUCIONAL 45/2004. COMPETÊNCIA DA JUSTIÇA DO TRABALHO.

2) CONFLITO NEGATIVO DE COMPETÊNCIA. CONTRIBUIÇÃO SINDICAL. AGENTE PÚBLICO MUNICIPAL. VÍNCULO CELETISTA ESTATUTÁRIO. COMPETÊNCIA DA JUSTIÇA DO TRABALHO.

Figura 17 – Visualização de duas verbetações geradas utilizando amostragem top-15. Nelas, é possível identificar efeitos efeitos do uso da amostragem como a geração de termos similares e expansão de siglas.

a reordenação de frases entre as diferentes verbetações. Os comportamentos observados são justificados pelo funcionamento de modelos de linguagem baseados em *Transformers* uma vez que, durante a geração de texto, os mesmos tendem a gerar tokens que aparecem em contextos semelhantes.

Original: TRIBUTÁRIO E PROCESSUAL CIVIL. RECURSO ESPECIAL. INTERPOSIÇÃO DE RECURSO, TEMPESTIVAMENTE, POR MEIO DE FAC-SIMILE. AUSÊNCIA DE APRESENTAÇÃO DA PETIÇÃO ORIGINAL, NO PRAZO PREVISTO NO ART. 2º DA LEI 9.800/99. RECURSO ESPECIAL INTERPOSTO VIA E-MAIL. INADMISSIBILIDADE. NÃO EQUIPARAÇÃO AO FAC-SIMILE. PRECEDENTES DO STJ. RECURSO ESPECIAL NÃO CONHECIDO.

Top-15) 5: TRIBUTÁRIO E PROCESSUAL CIVIL. RECURSO ESPECIAL, POR FAC-SIMILE. INTERPOSIÇÃO, NA ORIGEM, DE AGRAVO DE INSTRUMENTO, POR INTERPOSIÇÃO DE RECURSO POR MEIO DE FAC-SIMILE, ORIGINAIS APRESENTADOS DENTRO DO PRAZO LEGAL. IMPOSSIBILIDADE. ART. 2o DA LEI 9.800/99. RECURSO INTEMPESTIVO. PEÇA INCOMPLETA. NÃO APRESENTAÇÃO. PRESCRIÇÃO. PRAZO PARA INTERPOSIÇÃO DE RECURSO. PRECEDENTES DO STJ. ART. 543-C DO CPC/73. TERMO INICIAL. SÚMULA 83/STJ. AGRAVO INTERNO IMPROVIDO.

Top-50) TRIBUTÁRIO E PROCESSUAL CIVIL. RECURSO ESPECIAL, POR FAC-SIMILE, INTERPOSTO CONTRA ACÓRDÃO PUBLICADO NA VIGÊNCIA DO CPC/2015. RECURSO INTERPOSTO INTERPOSTO, POR FAX, POR E-MAIL. NÃO CABIMENTO. RECURSO CONTRA ACÓRDÃO PUBLICADO NA VIGÊNCIA DA LEI 9.800/99, QUE DETERMINOU A APRESENTAÇÃO DOS ORIGINAIS, NOS TERMOS DO ART. 2o DA LEI 9.800/99. PRECEDENTES DO STJ. RECURSO ESPECIAL NÃO CONHECIDO.

Top-100) TRIBUTÁRIO E PROCESSUAL CIVIL. RECURSO ESPECIAL, POR FAC-SIMILE, INTERPOSTO POR E-MAIL. POSSIBILIDADE. INTERPOSIÇÃO DE PARA APRESENTAÇÃO DE AGRAVO DE INSTRUMENTO. RECURSO INTERPOSTO DENTRO DO PRAZO LEGAL. ART. 2o DA LEI 9.800/99. POSSIBILIDADE. RECURSO ESPECIAL INTEMPESTIVO. ALEGADA OFENSA AO ART. 137 DO CTN. NÃO OCORRÊNCIA. APLICAÇÃO DO DISPOSTO NO ART. 927 DO CPC/2015. RECURSO ESPECIAL REPETITIVO JULGADO IMPROCEDENTE. PRECEDENTES DO STJ.

Figura 18 – Visualização de exemplos de verbetações geradas utilizando diferentes valores de K.

A Figura 18 ilustra o efeito da variação dos valores K. Ao aumentar-se o valor de K, os comportamentos apontados anteriormente se mantêm, e observamos de fato um aumento na variabilidade do texto. Entretanto, a possibilidade do modelo gerar texto desconexo em relação à entrada (contexto da geração de texto) também aumenta, o que pode ter prejudicado os métodos de RI estudados.

Em todos os exemplos apresentados na Figura 18, é possível notar a presença de termos e citações de artigos jurídicos que não constam na verbetação original. Por mais que tais termos

possam aparecer em contextos semelhantes, não podemos descartar a possibilidade de os mesmos terem sido originados por alucinações do modelo PTT5. Não é possível, sem conhecimento de domínio, afirmar que tais termos "novos" estejam de fato relacionados ao texto de entrada. Podemos apenas supor que os mesmos aparecem em contextos semelhantes. Note que esta mesma discussão é valida para a análise de verbetações geradas com decodificação gulosa.

As possíveis causas apontadas ajudam a compreender o porquê dos experimentos com amostragem não terem os resultados esperados. Conforme apontado nos experimentos anteriores, os métodos TF-IDF e BM25 se beneficiam da introdução de termos discriminativos. No entanto, conforme observado, as variações criadas por eventuais paráfrases e reordenações não foram suficientes para ajudar na tarefa de RI estudada. A possível introdução de termos não relacionados ao texto de entrada também pode ter prejudicado as comparações de ementas no processo de busca, tornando mais difícil distinguir os temas das mesmas. Além disto, observa-se que ao concatenar múltiplas variações de verbetações, adicionamos muitos termos repetidos aos documentos, o que pode ter influenciado negativamente os métodos esparsos de RI avaliados.

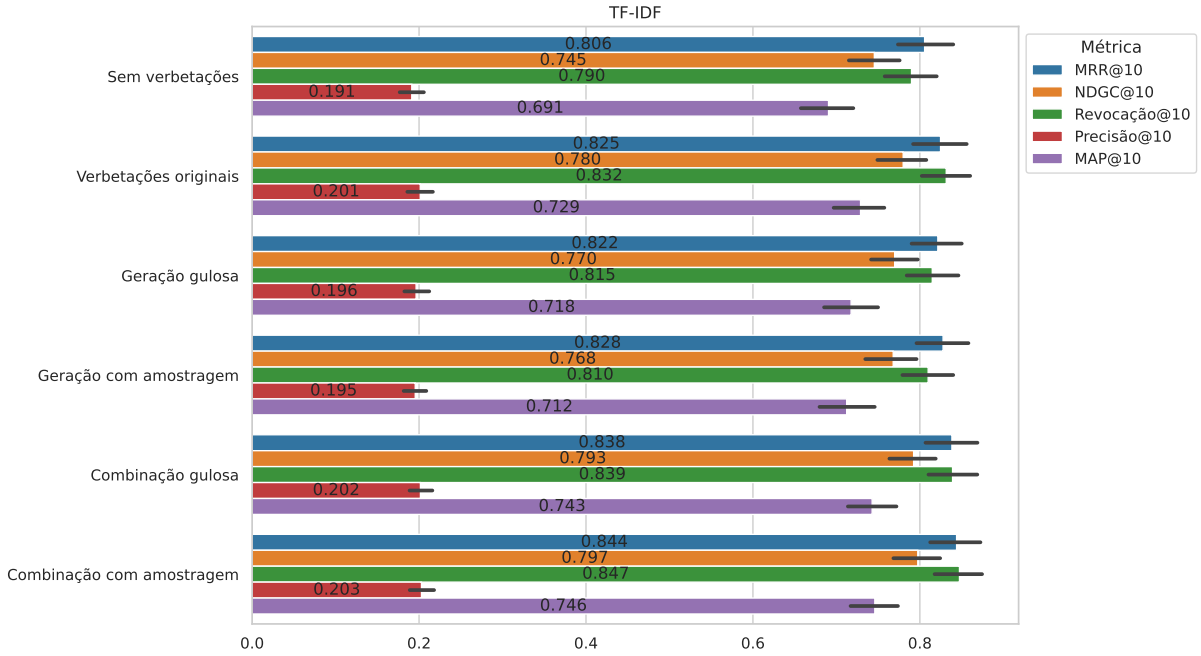
Além das justificativas apresentadas, a quantidade de dados para treino também pode ter tido um efeito negativo na geração não-determinística de verbetações. Embora os resultados para geração gulosa tenham sido melhores, a falta de variabilidade nos exemplos de treino, em virtude do tamanho pequeno, pode ter prejudicado a geração com amostragem top-K. Desta forma, é possível que os resultados possam ser melhorados com um aumento do conjunto de dados para o treino dos geradores de texto supervisionados. Sendo assim, em virtude dos comentários feitos, e tendo em mente a complexidade linguística dos documentos jurídicos estudados, podemos concluir com base nos resultados apontados que geração de variações de verbetações por meio de decodificação com amostragem é um problema consideravelmente difícil.

5.2.5 Compilação de Experimentos e Métricas

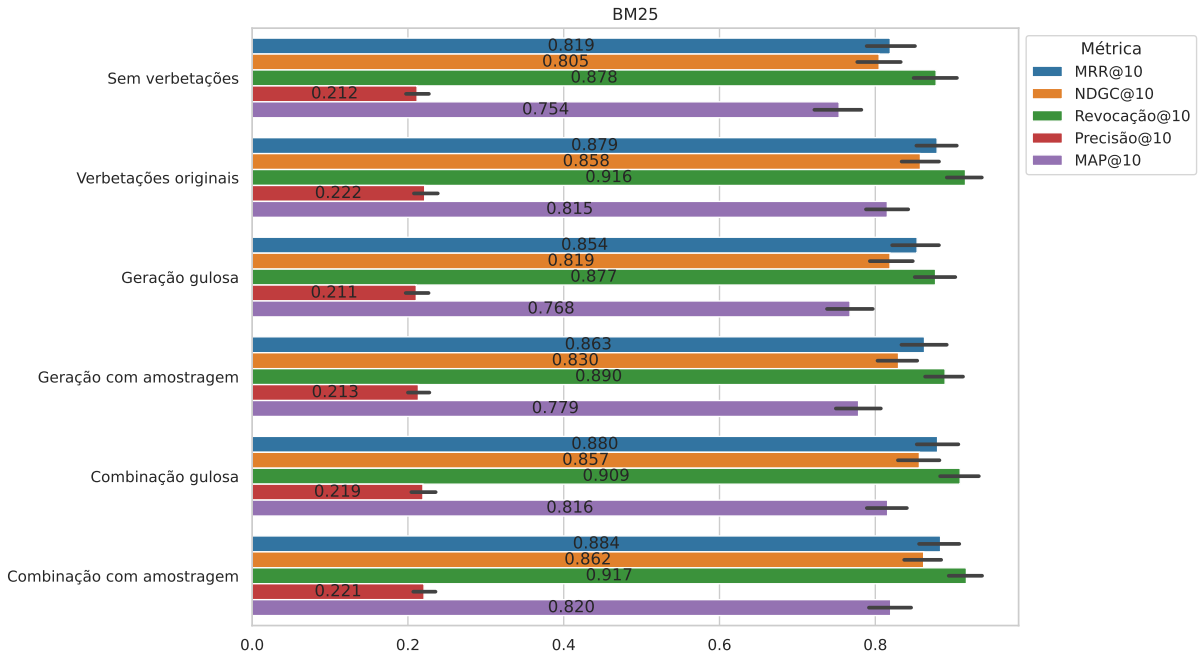
Tendo apresentado os resultados de todos os experimentos descritos na Seção 4.4, a Figura 19 sumariza todas as métricas obtidas. Através da análise dos gráficos é possível notar visualmente os principais resultados discutidos nesta Seção.

Assim, observa-se que todas as verbetações geradas trouxeram ganhos nas métricas de RI, em relação a realizar o procedimento de RI sem utilizar verbetações. Também podemos constatar visualmente, os desempenhos similares para as verbetações geradas com decodificação gulosa e com amostragem para tanto os experimentos sem e com combinação de verbetações. Ambas abordagens forneceram ganhos nas métricas em comparação ao não uso de verbetações.

Por fim, destacamos novamente os desempenhos obtidos ao realizar buscas com as verbetações originais. Conforme discutido nos experimentos, as métricas obtidas para as verbetações originais atuam como limites superiores para os demais experimentos. Este fato se reflete na superioridade das suas métricas nos gráficos, em relação às obtidas para as duas abordagens



(a) Métricas obtidas para todos os experimentos com TF-IDF.



(b) Métricas obtidas para todos os experimentos com BM25.

Figura 19 – Compilação de métricas de RI obtidas em todos os experimentos realizados.

de geração de verbetações. Mesmo combinando verbetações, os melhores resultados obtidos corresponderam apenas a pequenos ganhos em pontos percentuais em relação às verbetações originais (com destaque para os resultados do TF-IDF).

5.2.6 Investigando os Valores das Métricas

Pelas métricas apresentadas nas Tabelas 5, 6, 7, 8 e 9 observamos valores altos para a métrica MRR@10 em todos os experimentos realizados (acima de 0.8). Os resultados sugerem que uma quantidade considerável de consultas maximizaram a métrica mencionada. Considerando a recuperação BM25 sem verbetações, identificamos que 369 das 482 consultas obtiveram o MRR máximo. Como os métodos de recuperação esparsos avaliados podem ser influenciados por termos em comum entre consultas e documentos, decidimos investigar a porcentagem destes tokens.

Para este fim, para cada consulta, determinamos a porcentagem de tokens únicos separados por espaços presentes tanto na consulta quanto nos documentos relevantes para a mesma. Consideramos apenas o texto dos parágrafos enumerados e ignoramos as verbetações para esta análise. A porcentagem foi obtida ao dividir o número de tokens na interseção dividido pelo número de tokens únicos na consulta. A Figura 20 apresenta as porcentagens de tokens em comum para consultas fáceis (máximo MRR@10 para a recuperação BM25 sem palavras-chave) e difíceis (não-máximo MRR@10).

Nota-se que para as consultas fáceis os valores percentuais de tokens em comum têm mediana acima de 80%. Por outro lado, para as consultas mais difíceis, a mediana está próxima de 48%. Uma explicação para os altos valores de interseção, segundo um especialista do STJ contatado, é o uso frequente de *templates* pré-definidos para a escrita de processos judiciais como os investigados neste trabalho.

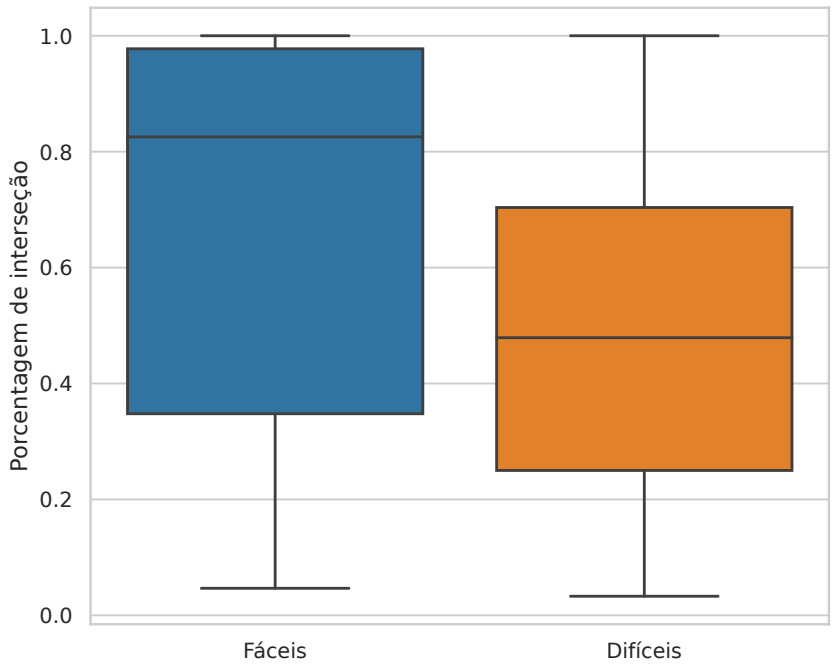


Figura 20 – Porcentagens de tokens em comum entre consultas e documentos para consultas fáceis e difíceis.

5.3 Considerações Finais

Ao longo deste Capítulo foram apresentados os resultados obtidos para os dois conjuntos de experimentos principais desta Dissertação. Na Seção 5.1 foram apresentados e discutidos as pontuações BLEU obtidas pelos *Transformers* avaliados e o melhor modelo obtido foi escolhido para gerar as verbetações para os próximos experimentos. Na Seção 5.2 foram apresentados os resultados obtidos para os experimentos de RI propostos. Obtivemos ganhos positivos para todos os experimentos com verbetações geradas com decodificação gulosa. Porém, para geração não-determinística os resultados foram contrários às expectativas e possíveis justificativas para este caso foram apresentadas e discutidas. No próximo Capítulo serão apresentadas as conclusões finais deste trabalho.

CAPÍTULO

6

CONCLUSÃO

Neste trabalho foram utilizados *Transformers* para automatizar a escrita de verbetações contidas em ementas de processos jurídicos brasileiros. Através de documentos disponibilizados pela plataforma Dados Abertos do STJ, foram treinados quatro modelos baseados em *Transformers* para a geração de verbetações. O melhor modelo foi o PTT5 que obteve uma pontuação BLEU acima de 37% para ambos os conjuntos de validação e teste. Ao comparar as verbetações geradas pelo melhor modelo com as verbetações originais observamos que as verbetações artificiais conseguiram captar as características principais das verbetações escritas por humanos, mesmo tendo sido geradas a partir de um conjunto de treino modesto em tamanho.

Com a finalidade de avaliar o desempenho dos métodos investigados, novos experimentos foram realizados utilizando as verbetações geradas pelo melhor modelo. Nesta avaliação, investigamos o uso das verbetações originais, das geradas e das combinações das mesmas em uma tarefa que emula um caso real de busca de jurisprudências. Para isso foram realizados quatro experimentos. No primeiro experimento, ao comparar documentos sem e com verbetações, foi constatado que as verbetações tem impacto positivo na tarefa de RI estudada, trazendo melhoras entre 1% e 6% nas métricas dois modelos de RI avaliados (TF-IDF e BM25).

No segundo experimento, ao empregar decodificação gulosa para geração das verbetações, foram obtidos ganhos para os dois métodos em relação a não usar verbetações. Embora as métricas observadas para as verbetações artificiais tenham sido inferiores às obtidas utilizando as verbetações originais, foram obtidos ganhos significativos para os dois modelos avaliados.

O terceiro experimento consistiu em combinar verbetações geradas com decodificação gulosa com as originais. Os resultados apresentados mostram que é possível obter métricas melhores combinando as verbetações para o método TFIDF. Por outro lado, para o método BM25 os resultados da combinação de verbetações não apresentaram melhoras em relação às verbetações originais.

O quarto e último experimento consistiu em repetir os dois últimos experimentos ante-

riores, porém empregando geração textual não-determinística top-K. Ao utilizar amostragem, é possível gerar múltiplas variações de verbetações para uma mesma entrada e foi estudado o impacto de concatenar as múltiplas variações de verbetações aos documentos do corpus de busca. No entanto, os resultados obtidos no melhor caso são equiparáveis às verbetações geradas com decodificação gulosa e não foram observados ganhos ao variar número de repetições e valores K.

Em virtude dos resultados apresentados, validamos a hipótese investigada. Assim, concluímos que é viável utilizar um *Transformer* (PTT5), treinado em um conjunto pequeno com menos de 100.000 exemplos, para gerar verbetações de forma automática. Uma contribuição relevante para a área esta no fato de que foram observados melhores resultados ao empregar o método mais simples de decodificação (a gulosa). Além disto, outra contribuição obtida deste trabalho foi o fato que a combinação de verbetações originais e geradas gerou ganhos significativos para o TF-IDF, mas não para o método BM25 e possíveis justificativas foram apresentadas.

No geral, os resultados obtidos foram satisfatórios, considerando a disponibilidade de dados e a complexidade da linguagem presente no meio jurídico. Mesmo que não haja interesse em automatizar completamente a escrita de verbetações, os modelos estudados ainda podem auxiliar na escrita de verbetações, fornecendo sugestões de verbetações para que um especialista faça correções conforme a necessidade.

Como produtos desta Dissertação de Mestrado, tivemos os seguintes artigos:

- (SAKIYAMA; ROMERO; NOGUEIRA, 2023) - aceito para publicação na conferência *International Joint Conference on Neural Networks* (IJCNN).
- (SAKIYAMA *et al.*, 2023) - aceito para publicação na conferência *Brazilian Conference on Intelligent Systems (BRACIS)*.

Finalizaremos as conclusões, apresentando exemplos de trabalhos futuros que podem ser originados desta Dissertação. Planejamos investigar a necessidade de prefixos nos modelos de codificador-decodificador (assim como os usados para os modelos somente decodificador) e experimentar realizar um pré-treinamento do modelo de linguagem PTT5 em documentos legais visando melhorar a geração de texto. Além disto, como o tokenizador do PTT5 distingue caracteres caixa-alta e caixa-baixa, também planejamos investigar a influência deste aspecto na geração das verbetações (que são predominantemente escritas em caixa-alta).

Também é possível melhorar a qualidade do treinamento investigando formas de obter mais documentos de diferentes fontes de todo Brasil. Conforme apontado na metodologia, optamos por não avaliar modelos (Exemplo: GPT-4) e métricas adicionais (Exemplo: *BERTScore*) por falta de tempo ou de recursos. Sendo assim, pretendemos investigar estas lacunas em trabalhos futuros. Por fim, desejamos realizar um estudo mais aprofundado a respeito da existência de alucinações nas verbetações geradas por todos os métodos de decodificação avaliados.

REFERÊNCIAS

ALTHAMMER, S.; ASKARI, A.; VERBERNE, S.; HANBURY, A. Dossier@ coliee 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. **arXiv preprint arXiv:2108.03937**, 2021. Disponível em: <https://arxiv.org/pdf/2108.03937.pdf>. Citado na página 53.

BA, J. L.; KIROS, J. R.; HINTON, G. E. Layer normalization. **arXiv preprint arXiv:1607.06450**, 2016. Disponível em: <https://arxiv.org/abs/1607.06450>. Citado na página 46.

BASSANI, E. ranx: A blazing-fast python library for ranking evaluation and comparison. In: **ECIR (2)**. Springer, 2022. (Lecture Notes in Computer Science, v. 13186), p. 259–264. Disponível em: <https://doi.org/10.1007/978-3-030-99739-7_30>. Citado na página 67.

BELZ, A.; REITER, E. Comparing automatic and human evaluation of nlg systems. In: **11th conference of the european chapter of the association for computational linguistics**. [s.n.], 2006. p. 313–320. Disponível em: <https://aclanthology.org/E06-1040.pdf>. Citado na página 38.

BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. **Transactions of the association for computational linguistics**, MIT Press, v. 5, p. 135–146, 2017. Disponível em: <https://doi.org/10.1162/tacl_a_00051>. Citado na página 54.

BROWN, T.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J. D.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. Language models are few-shot learners. In: LAROCHELLE, H.; RANZATO, M.; HADSELL, R.; BALCAN, M.; LIN, H. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2020. v. 33, p. 1877–1901. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf>. Citado na página 51.

BUBECK, S.; CHANDRASEKARAN, V.; ELDAN, R.; GEHRKE, J.; HORVITZ, E.; KAMAR, E.; LEE, P.; LEE, Y. T.; LI, Y.; LUNDBERG, S. *et al.* Sparks of artificial general intelligence: Early experiments with gpt-4. **arXiv preprint arXiv:2303.12712**, 2023. Disponível em: <https://arxiv.org/abs/2303.12712>. Citado na página 52.

CANEDO, E. D.; MARTINS, V. A.; RIBEIRO, V. C.; REIS, V. E. dos; CHAVES, L. A. C.; GRAVINA, R. M.; DIAS, F. A. M.; MENDONÇA, F. L. Lopes de; OROZCO, A. L. S.; BALANIUK, R. *et al.* Development and evaluation of an intelligence and learning system in jurisprudence text mining in the field of competition defense. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 11, n. 23, p. 11365, 2021. Disponível em: <https://doi.org/10.3390/app112311365>. Citado na página 21.

CARMO, D.; PIAU, M.; CAMPIOTTI, I.; NOGUEIRA, R.; LOTUFO, R. Ptt5: Pretraining and validating the t5 model on brazilian portuguese data. **arXiv preprint arXiv:2008.09144**, 2020. Disponível em: <https://arxiv.org/abs/2008.09144>. Citado nas páginas 22, 59, 61 e 70.

CELIKYILMAZ, A.; CLARK, E.; GAO, J. Evaluation of text generation: A survey. **arXiv preprint arXiv:2006.14799**, 2020. Disponível em: <https://arxiv.org/abs/2006.14799>. Citado na página 38.

CNJ, C. N. de J. **Conselho Nacional de Justiça — Justiça em Números**. 23. <https://www.cnj.jus.br/pesquisas-judiciarias/justica-em-numeros/>. (Accessed on 24/01/2023). Citado na página 21.

CRESTANI, F.; LALMAS, M.; RIJSBERGEN, C. J. V.; CAMPBELL, I. "is this document relevant?... probably" a survey of probabilistic models in information retrieval. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 30, n. 4, p. 528–552, 1998. Disponível em: <https://doi.org/10.1145/299917.299920>. Citado na página 66.

DAI, Z.; YANG, Z.; YANG, Y.; CARBONELL, J.; LE, Q. V.; SALAKHUTDINOV, R. Transformer-xl: Attentive language models beyond a fixed-length context. **arXiv preprint arXiv:1901.02860**, 2019. Disponível em: <https://arxiv.org/abs/1901.02860>. Citado na página 53.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. Disponível em: <https://arxiv.org/abs/1810.04805>. Citado nas páginas 42 e 54.

FAN, A.; LEWIS, M.; DAUPHIN, Y. Hierarchical neural story generation. **arXiv preprint arXiv:1805.04833**, 2018. Disponível em: <https://arxiv.org/abs/2202.06417>. Citado na página 64.

FAN, A.; LEWIS, M.; DAUPHIN, Y. N. Hierarchical neural story generation. **CoRR**, abs/1805.04833, 2018. Disponível em: <http://arxiv.org/abs/1805.04833>. Citado nas páginas 50 e 52.

FEIJO, D.; MOREIRA, V. Summarizing legal rulings: Comparative experiments. In: **proceedings of the international conference on recent advances in natural language processing (RANLP 2019)**. [s.n.], 2019. p. 313–322. Disponível em: <http://dx.doi.org/10.26615/978-954-452-056-4_036>. Citado na página 53.

FILHO, J. A. W.; WILKENS, R.; IDIART, M.; VILLAVICENCIO, A. The brwac corpus: A new open resource for brazilian portuguese. In: **Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)**. [s.n.], 2018. Disponível em: <https://aclanthology.org/L18-1686.pdf>. Citado nas páginas 47 e 59.

FLORIDI, L.; CHIRIATTI, M. Gpt-3: Its nature, scope, limits, and consequences. **Minds and Machines**, Springer, v. 30, n. 4, p. 681–694, 2020. Disponível em: <https://doi.org/10.1007/s11023-020-09548-1>. Citado nas páginas 37, 48 e 51.

GOMES, T.; LADEIRA, M. A new conceptual framework for enhancing legal information retrieval at the brazilian superior court of justice. In: **Proceedings of the 12th International Conference on Management of Digital EcoSystems**. [s.n.], 2020. p. 26–29. Disponível em: <https://doi.org/10.1145/3415958.3433087>. Citado na página 54.

GUIMARÃES, J. A. C.; SANTOS, J. C. G. A ementa jurisprudencial como resumo informativo em um domínio especializado: aspectos estruturais. **Brazilian Journal of Information Science: research trends**, v. 10, n. 3, 2016. Disponível em: <https://doi.org/10.36311/1981-1640.2016.v10n3.05.p32>. Citado na página 24.

HARMAN, D. Information retrieval evaluation. **Synthesis Lectures on Information Concepts, Retrieval, and Services**, Morgan & Claypool Publishers, v. 3, n. 2, p. 1–119, 2011. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/215206.215351>. Citado na página 28.

HOLTZMAN, A.; BUYS, J.; FORBES, M.; CHOI, Y. The curious case of neural text degeneration. **CoRR**, abs/1904.09751, 2019. Disponível em: <http://arxiv.org/abs/1904.09751>. Citado nas páginas 49, 50 e 52.

HUANG, Y.; SHEN, X.; LI, C.; GE, J.; LUO, B. Dependency learning for legal judgment prediction with a unified text-to-text transformer. **arXiv preprint arXiv:2112.06370**, 2021. Disponível em: <https://arxiv.org/abs/2112.06370>. Citado na página 53.

JI, Z.; LEE, N.; FRIESKE, R.; YU, T.; SU, D.; XU, Y.; ISHII, E.; BANG, Y.; MADOTTO, A.; FUNG, P. Survey of hallucination in natural language generation. **ACM Computing Surveys**, ACM New York, NY, 2022. Disponível em: <https://doi.org/10.1145/3571730>. Citado na página 52.

KUDO, T.; RICHARDSON, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. **arXiv preprint arXiv:1808.06226**, 2018. Disponível em: <https://arxiv.org/abs/1808.06226>. Citado na página 61.

LEWIS, M.; LIU, Y.; GOYAL, N.; GHAZVININEJAD, M.; MOHAMED, A.; LEVY, O.; STOYANOV, V.; ZETTLEMOYER, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. **arXiv preprint arXiv:1910.13461**, 2019. Disponível em: <https://arxiv.org/abs/1910.13461>. Citado na página 53.

LIMA, J. P.; COSTA, J. A.; ARAÚJO, D. C. Comparison of feature extraction methods for brazilian legal documents clustering. In: IEEE. **2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)**. 2021. p. 1–5. Disponível em: <https://doi.org/10.1109/LA-CCI48322.2021.9769839>. Citado nas páginas 54, 55 e 66.

LIN, C.-Y. Rouge: A package for automatic evaluation of summaries. In: **Text summarization branches out**. [s.n.], 2004. p. 74–81. Disponível em: <https://aclanthology.org/W04-1013.pdf>. Citado na página 40.

LIN, J.; NOGUEIRA, R.; YATES, A. Pretrained transformers for text ranking: Bert and beyond. **Synthesis Lectures on Human Language Technologies**, Morgan & Claypool Publishers, v. 14, n. 4, p. 1–325, 2021. Disponível em: <https://doi.org/10.1145/3437963.3441667>. Citado nas páginas 28, 30 e 53.

MAIA, M.; BEZERRA, C. A. Análise bibliométrica dos artigos científicos de jurimetria publicados no brasil. **RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação**, v. 18, p. e020018–e020018, 2020. Disponível em: <https://doi.org/10.20396/rdbci.v18i0.8658889>. Citado na página 22.

MANDAL, A.; GHOSH, K.; GHOSH, S.; MANDAL, S. Unsupervised approaches for measuring textual similarity between legal court case reports. **Artificial Intelligence and Law**, Springer,

v. 29, n. 3, p. 417–451, 2021. Disponível em: <https://doi.org/10.1007/s10506-020-09280-2>. Citado nas páginas 54 e 66.

MAYNEZ, J.; NARAYAN, S.; BOHNET, B.; MCDONALD, R. T. On faithfulness and factuality in abstractive summarization. **CoRR**, abs/2005.00661, 2020. Disponível em: <https://arxiv.org/abs/2005.00661>. Citado na página 52.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: BURGES, C.; BOTTOU, L.; WELLING, M.; GHAHRAMANI, Z.; WEINBERGER, K. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2013. v. 26. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf>. Citado na página 36.

NGUYEN, T.; ROSENBERG, M.; SONG, X.; GAO, J.; TIWARY, S.; MAJUMDER, R.; DENG, L. Ms marco: A human generated machine reading comprehension dataset. In: **CoCo@ NIPS**. [s.n.], 2016. Disponível em: <https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf>. Citado na página 31.

NOGUEIRA, R.; YANG, W.; LIN, J.; CHO, K. Document expansion by query prediction. **arXiv preprint arXiv:1904.08375**, 2019. Disponível em: <https://arxiv.org/abs/1904.08375>. Citado nas páginas 65, 74 e 75.

OLIVEIRA, R. A. N de; JUNIOR, M. C. Experimental analysis of stemming on jurisprudential documents retrieval. **Information**, Multidisciplinary Digital Publishing Institute, v. 9, n. 2, p. 28, 2018. Disponível em: <https://doi.org/10.3390/info9020028>. Citado na página 54.

OPENAI. Gpt-4 technical report. **arXiv**, 2023. Disponível em: <https://arxiv.org/abs/2303.08774>. Citado nas páginas 52 e 60.

OSTENDORFF, M.; ASH, E.; RUAS, T.; GIPP, B.; MORENO-SCHNEIDER, J.; REHM, G. Evaluating document representations for content-based legal literature recommendations. In: **Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law**. [s.n.], 2021. p. 109–118. Disponível em: <https://doi.org/10.1145/3462757.3466073>. Citado nas páginas 54 e 63.

OUNIS, I.; AMATI, G.; PLACHOURAS, V.; HE, B.; MACDONALD, C.; JOHNSON, D. Terrier information retrieval platform. In: SPRINGER. **European Conference on Information Retrieval**. 2005. p. 517–519. Disponível em: <https://doi.org/10.1007/978-3-540-31865-1_37>. Citado na página 35.

PAPINENI, K.; ROUKOS, S.; WARD, T.; ZHU, W.-J. Bleu: a method for automatic evaluation of machine translation. In: **Proceedings of the 40th annual meeting of the Association for Computational Linguistics**. [s.n.], 2002. p. 311–318. Disponível em: <https://aclanthology.org/P02-1040.pdf>. Citado nas páginas 38, 39 e 60.

PEDROSO, D. d. S. C.; LADEIRA, M.; FALEIROS, T. de P. Does semantic search performs better than lexical search in the task of assisting legal opinion writing? In: IEEE. **2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)**. 2019. p. 680–685. Disponível em: <https://doi.org/10.1109/ICMLA.2019.00123>. Citado nas páginas 54 e 55.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [s.n.], 2014. p. 1532–1543. Disponível em: <https://aclanthology.org/D14-1162.pdf>. Citado nas páginas 36 e 41.

PERIC, L.; MIJIC, S.; STAMMBACH, D.; ASH, E. Legal language modeling with transformers. In: CEUR-WS. **Proceedings of the Fourth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2020) held online in conjunction with te 33rd International Conference on Legal Knowledge and Information Systems (JURIX 2020) December 9, 2020**. 2020. v. 2764. Disponível em: <https://doi.org/10.3929/ethz-b-000456079>. Citado na página 53.

POST, M. A call for clarity in reporting bleu scores. **arXiv preprint arXiv:1804.08771**, 2018. Disponível em: <https://arxiv.org/abs/1804.08771>. Citado na página 60.

PRADEEP, R.; MA, X.; ZHANG, X.; CUI, H.; XU, R.; NOGUEIRA, R.; LIN, J. H2oloo at trec 2020: When all you got is a hammer... deep learning, health misinformation, and precision medicine. **Corpus**, v. 5, n. d3, p. d2, 2020. Disponível em: <https://trec.nist.gov/pubs/trec29/papers/h2oloo.DL.HM.PM.pdf>. Citado nas páginas 54 e 66.

RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I. *et al.* Language models are unsupervised multitask learners. **OpenAI blog**, v. 1, n. 8, p. 9, 2019. Disponível em: <https://life-extension.github.io/2020/05/27/GPT%E6%8A%80%E6%9C%AF%E5%88%9D%E6%8E%A2/language-models.pdf>. Citado nas páginas 42, 48, 51, 59, 60 e 70.

RAFFEL, C.; SHAZEER, N.; ROBERTS, A.; LEE, K.; NARANG, S.; MATENA, M.; ZHOU, Y.; LI, W.; LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. **J. Mach. Learn. Res.**, JMLR.org, v. 21, n. 1, jan 2020. ISSN 1532-4435. Disponível em: <https://dl.acm.org/doi/abs/10.5555/3455716.3455856>. Citado na página 52.

REI, R.; STEWART, C.; FARINHA, A. C.; LAVIE, A. COMET: A neural framework for MT evaluation. In: **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Online: Association for Computational Linguistics, 2020. p. 2685–2702. Disponível em: <https://aclanthology.org/2020.emnlp-main.213>. Citado na página 60.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. **arXiv preprint arXiv:1908.10084**, 2019. Disponível em: <https://arxiv.org/abs/1908.10084>. Citado na página 37.

ROBERTSON, S. E. The probability ranking principle in ir. **Journal of documentation**, MCB UP Ltd, 1977. Disponível em: <https://doi.org/10.1108/eb026647>. Citado na página 35.

ROBERTSON, S. E.; WALKER, S. Okapi/keenbow at trec-8. In: CITESEER. **TREC**. 1999. v. 8, p. 151–162. Disponível em: <https://trec.nist.gov/pubs/trec8/papers/okapi.pdf>. Citado nas páginas 35, 53 e 66.

ROSA, G. M.; BONIFACIO, L. H.; SOUZA, L. R. de; LOTUFO, R.; NOGUEIRA, R. A cost-benefit analysis of cross-lingual transfer methods. **arXiv preprint arXiv:2105.06813**, 2021. Disponível em: <https://arxiv.org/abs/2105.06813>. Citado na página 70.

ROSA, G. M.; RODRIGUES, R. C.; LOTUFO, R.; NOGUEIRA, R. Yes, bm25 is a strong baseline for legal case retrieval. **arXiv preprint arXiv:2105.05686**, 2021. Disponível em: <https://arxiv.org/abs/2105.05686>. Citado nas páginas 54 e 66.

ROTHE, S.; NARAYAN, S.; SEVERYN, A. Leveraging pre-trained checkpoints for sequence generation tasks. **Transactions of the Association for Computational Linguistics**, MIT Press, v. 8, p. 264–280, 2020. Disponível em: <https://doi.org/10.1162/tacl_a_00313>. Citado na página 53.

RUSSELL-ROSE, T.; CHAMBERLAIN, J.; AZZOPARDI, L. Information retrieval in the workplace: A comparison of professional search practices. **Information Processing & Management**, Elsevier, v. 54, n. 6, p. 1042–1057, 2018. Disponível em: <https://doi.org/10.1016/j.ipm.2018.07.003>. Citado na página 67.

SAKIYAMA, K.; MONTANARI, R.; JUNIOR, R.; NOGUEIRA, R.; ROMERO, R. Exploring text decoding methods for portuguese legal text generation. In: **Intelligent Systems: 12th Brazilian Conference, BRACIS 2023, Minas Gerais, Brazil, September 25–29, 2023**. [S.l.: s.n.], 2023. Citado na página 86.

SAKIYAMA, K. M.; ROMERO, R.; NOGUEIRA, R. F. Automatic keyphrase generation for brazilian legal information retrieval. In: **2023 International Joint Conference on Neural Networks (IJCNN) (IJCNN 2023)**. Queensland, Australia: [s.n.], 2023. p. 8. Citado na página 86.

SANDERSON, M. **Test collection based evaluation of information retrieval systems**. Now Publishers Inc, 2010. Disponível em: <http://dx.doi.org/10.1561/1500000009>. Citado na página 29.

SCAO, T. L.; FAN, A.; AKIKI, C.; PAVLICK, E.; ILIĆ, S.; HESSLOW, D.; CASTAGNÉ, R.; LUCCIONI, A. S.; YVON, F.; GALLÉ, M. *et al.* Bloom: A 176b-parameter open-access multilingual language model. **arXiv preprint arXiv:2211.05100**, 2022. Disponível em: <https://arxiv.org/abs/2211.05100>. Citado nas páginas 52 e 60.

SOUZA, E.; MORIYAMA, G.; VITÓRIO, D.; CARVALHO, A. C. de; FÉLIX, N.; ALBUQUERQUE, H. O.; OLIVEIRA, A. L. Assessing the impact of stemming algorithms applied to brazilian legislative documents retrieval. In: SBC. **Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana**. 2021. p. 227–236. Disponível em: <https://doi.org/10.5753/stil.2021.17802>. Citado na página 54.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Bertimbau: Pretrained bert models for brazilian portuguese. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. 2020. p. 403–417. Disponível em: <https://doi.org/10.1007/978-3-030-61377-8_28>. Citado nas páginas 22, 47 e 70.

STJ. **Supremo Tribunal de Justiça — Composição**. 23. <https://www.stj.jus.br/sites/portalp/Institucional/Composicao>. (Acessado em 18/02/2023). Citado na página 57.

SU, Y.; LAN, T.; WANG, Y.; YOGATAMA, D.; KONG, L.; COLLIER, N. A contrastive framework for neural text generation. **arXiv preprint arXiv:2202.06417**, 2022. Disponível em: <https://arxiv.org/abs/2202.06417>. Citado nas páginas 52 e 65.

TONON, A.; DEMARTINI, G.; CUDRÉ-MAUROUX, P. Pooling-based continuous evaluation of information retrieval systems. **Information Retrieval Journal**, Springer, v. 18, n. 5, p. 445–472, 2015. Disponível em: <https://doi.org/10.1007/s10791-015-9266-y>. Citado na página 30.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L. u.; POLOSUKHIN, I. Attention is all you need. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>. Citado nas páginas 11, 22, 41, 43 e 51.

XUE, L.; CONSTANT, N.; ROBERTS, A.; KALE, M.; AL-RFOU, R.; SIDDHANT, A.; BARUA, A.; RAFFEL, C. mt5: A massively multilingual pre-trained text-to-text transformer. **arXiv preprint arXiv:2010.11934**, 2020. Disponível em: <https://arxiv.org/abs/2010.11934>. Citado nas páginas 52, 59 e 61.

YANG, P.; FANG, H.; LIN, J. Anserini: Reproducible ranking baselines using lucene. **Journal of Data and Information Quality (JDIQ)**, ACM New York, NY, USA, v. 10, n. 4, p. 1–20, 2018. Disponível em: <https://doi.org/10.1145/3239571>. Citado na página 35.

YILMAZ, Z. A.; WANG, S.; YANG, W.; ZHANG, H.; LIN, J. Applying bert to document retrieval with birch. In: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations**. [s.n.], 2019. p. 19–24. Disponível em: <http://dx.doi.org/10.18653/v1/D19-3004>. Citado na página 28.

YOON, J.; JUNAID, M.; ALI, S.; LEE, J. Abstractive summarization of korean legal cases using pre-trained language models. In: IEEE. **2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)**. 2022. p. 1–7. Disponível em: <https://doi.org/10.1109/IMCOM53663.2022.9721808>. Citado na página 53.

ZHANG, S.; ROLLER, S.; GOYAL, N.; ARTETXE, M.; CHEN, M.; CHEN, S.; DEWAN, C.; DIAB, M.; LI, X.; LIN, X. V. *et al.* Opt: Open pre-trained transformer language models. **arXiv preprint arXiv:2205.01068**, 2022. Disponível em: <https://arxiv.org/abs/2205.01068>. Citado nas páginas 51 e 60.

ZHANG, T.; KISHORE, V.; WU, F.; WEINBERGER, K. Q.; ARTZI, Y. Bertscore: Evaluating text generation with BERT. **CoRR**, abs/1904.09675, 2019. Disponível em: <http://arxiv.org/abs/1904.09675>. Citado na página 60.

ZHUANG, F.; QI, Z.; DUAN, K.; XI, D.; ZHU, Y.; ZHU, H.; XIONG, H.; HE, Q. A comprehensive survey on transfer learning. **Proceedings of the IEEE**, IEEE, v. 109, n. 1, p. 43–76, 2020. Disponível em: <https://doi.org/10.1109/JPROC.2020.3004555>. Citado na página 46.

GLOSSÁRIO

BERT: Bidirectional Encoder Representations from Transformers.

BERTimbau: Modelo de linguagem baseado no *BERT*, ajustado para textos em português.

ColBERTimbau: Modelo de ranqueamento baseado no *ColBERT*, porém utilizando o modelo de linguagem *BERTimbau*.

monoBERTimbau: Modelo de ranqueamento baseado no *monoBERT*, porém utilizando o modelo de linguagem *BERTimbau*.

monoPTT5: Modelo de ranqueamento baseado no *monoT5*, porém utilizando o modelo de linguagem *PTT5*.

QRELS: Anotações de relevância para pares consulta e documento.

APÊNDICE

A

EXEMPLOS DE VERBETAÇÕES GERADAS

Neste Apêndice serão apresentados exemplos de verbetações geradas. Foram amostrados exemplos aleatoriamente entre as verbetações geradas para o conjunto de teste para gerar as visualizações.

A.1 Verbetações Geradas com Decodificação Gulosa

Original: PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO NO RECURSO ESPECIAL. ALEGAÇÃO DE OMISSÃO. RESSALVAS DE FUNDAMENTAÇÃO CONTIDAS NO VOTO-VISTA. OMISSÃO NÃO CARCTERIZADA.
Gerada: PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO. OMISSÃO. VÍCIO INEXISTENTE. REJEIÇÃO.

Original: PROCESSUAL CIVIL. AGRAVO REGIMENTAL NOS EMBARGOS DE DIVERGÊNCIA. ACÓRDÃO EMBARGADO QUE NÃO CONHECEU DA QUESTÃO OBJETO DA DIVERGÊNCIA.
Gerada: AGRAVO REGIMENTAL NOS EMBARGOS DE DIVERGÊNCIA EM AGRAVO RECURSO ESPECIAL. DISSÍDIO NÃO CONFIGURADO. AUSÊNCIA DE SIMILITUDE FÁTICA ENTRE OS ACÓRDÃOS CONFRONTADOS. AGRAVO REGIMENTAL NÃO PROVIDO.

Original: AGRAVO REGIMENTAL. HABEAS CORPUS. TRÁFICO DE DROGAS MAJORADO. SENTENÇA. PRETENSÃO DE ABSOLVIÇÃO. ALEGAÇÃO DE NULIDADE. IMPROCEDÊNCIA. SENTENÇA BASEADA EM ELEMENTOS DE PROVA PRODUZIDOS EM JUÍZO. WRIT SUBSTITUTIVO DE RECURSO ESPECIAL. FALTA DE CABIMENTO. CONSTRANGIMENTO ILEGAL MANIFESTO. AUSÊNCIA. MANUTENÇÃO DA DECISÃO MONOCRÁTICA QUE SE IMPÕE.
Gerada: AGRAVO REGIMENTAL HABEAS CORPUS. TRÁFICO DE DROGAS. WRIT IMPETRADO CONTRA DECISÃO MONOCRÁTICA DO RELATOR, QUE INDEFERIU LIMINARMENTE O WRIT. SUBSTITUIÇÃO AO RECURSO CABÍVEL. AUSÊNCIA DE CONSTRANGIMENTO ILEGAL MANIFESTO. MANUTENÇÃO DA DECISÃO QUE SE IMPÕE.

Original: AGRAVO REGIMENTAL NO AGRAVO EM RECURSO ESPECIAL. ORGANIZAÇÃO CRIMINOSA. FURTOS E ROUBOS DE VEÍCULOS. ADULTERAÇÃO DE PLACAS DE CHASSIS. FALSIFICAÇÃO DE DOCUMENTOS PÚBLICOS. PENA-BASE FIXADA ACIMA DO MÍNIMO LEGAL. PROPORCIONALIDADE. DESCONSTITUIÇÃO. REEXAME DE PROVAS. IMPOSSIBILIDADE. INCIDÊNCIA DA SÚMULA N. 7 DO STJ. AGRAVO REGIMENTAL NÃO PROVIDO.
Gerada: AGRAVO REGIMENTAL NO AGRAVO EM RECURSO ESPECIAL. ORGANIZAÇÃO CRIMINOSA. DOSIMETRIA. PENA-BASE. CULPABILIDADE, CONDUTA SOCIAL, PERSONALIDADE, MOTIVOS, CIRCUNSTÂNCIAS E CONSEQUÊNCIAS DO CRIME. FUNDAMENTAÇÃO CONCRETA. DISCRICIONARIEDADE DO JULGADOR. SÚMULA N. 7 DO STJ. AGRAVO REGIMENTAL NÃO PROVIDO.

Figura 21 – Exemplos de verbetações geradas com decodificação gulosa.

Original: AGRAVO REGIMENTAL NO RECURSO ORDINÁRIO EM HABEAS CORPUS. IMPETRAÇÃO SUBSTITUTIVA DE REVISÃO CRIMINAL. NULIDADE. OITIVA DE TESTEMUNHAS. AUSÊNCIA DE INTIMAÇÃO PELO JUÍZO DEPRECADO. DESNECESSIDADE. PREJUÍZO NÃO COMPROVADO. NO MAIS, NÃO ENFRENTAMENTO DOS FUNDAMENTOS DA DECISÃO AGRAVADA. SÚMULA 182/STJ. AGRAVO DESPROVIDO.
Gerada: AGRAVO REGIMENTAL NO RECURSO ORDINÁRIO EM HABEAS CORPUS. CRIME DE ESTUPRO DE VULNERÁVEL. NULIDADE. INTIMAÇÃO DA DEFESA DA EXPEDIÇÃO DA CARTA PRECATÓRIA. DESNECESSIDADE. NO MAIS, NÃO ENFRENTAMENTO DOS FUNDAMENTOS DA DECISÃO AGRAVADA. SÚMULA 182/STJ. AGRAVO DESPROVIDO.

Original: DIREITO TRIBUTÁRIO. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. APURAÇÃO DO VALOR VENAL DE IMÓVEL NÃO PREVISTO ORIGINALMENTE NA PLANTA GENÉRICA DE VALORES. REPERCUSSÃO GERAL RECONHECIDA. DEVOLUÇÃO À ORIGEM PELO TEMA 1.084. PRECEDENTES DO SUPREMO TRIBUNAL FEDERAL.
Gerada: PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. AÇÃO DE COBRANÇA. IPTU. IMÓVEL NÃO PREVISTO NA PLANTA VALOR VENAL. REPERCUSSÃO GERAL RECONHECIDA. TEMA 1.084 COM REPERCUSSÃO GERAL RECONHECIDA. DEVOLUÇÃO À ORIGEM.

Original: AGRAVO REGIMENTAL NO HABEAS CORPUS. TRÁFICO DE DROGAS. PRISÃO PREVENTIVA. PERICULUM LIBERTATIS. CRISE MUNDIAL DE COVID-19. NÃO DEMONSTRADA A EXISTÊNCIA DE RISCO À INTEGRIDADE FÍSICA DO ENCARCERADO. AGRAVO REGIMENTAL NÃO PROVIDO.
Gerada: AGRAVO REGIMENTAL NO HABEAS CORPUS. TRÁFICO DE DROGAS. PRISÃO PREVENTIVA. PERICULUM LIBERTATIS. AGRAVO REGIMENTAL NÃO PROVIDO.

Original: AGRAVO INTERNO. AGRAVO EM RECURSO ESPECIAL. RESPONSABILIDADE CIVIL. ACIDENTE ENVOLVENDO CAMINHÃO DE PROPRIEDADE DA AGRAVANTE QUE TOMBOU AO REALIZAR CURVA EM RODOVIA ATINGINDO RESIDÊNCIAS PRÓXIMAS QUE A MARGEAVAM, INCENDIANDO-SE EM SEGUIDA. ALEGAÇÃO DE OFENSA AO ART. 1.022 DO CÓDIGO DE PROCESSO CIVIL DE 2015. INEXISTÊNCIA. DANO E NEXO CAUSAL COMPROVADOS, ENSEJANDO O DEVER DE INDENIZAR. REEXAME FÁTICO DOS AUTOS. SÚMULA N. 7/STJ. DANO MORAL.VALOR RAZOÁVEL.
Gerada: AGRAVO INTERNO. AGRAVO EM RECURSO ESPECIAL. AÇÃO DE INDENIZAÇÃO. ACIDENTE DE TRÂNSITO. VIOLAÇÃO DOS ARTS. 489 E 1.022 DO CÓDIGO DE PROCESSO CIVIL DE 2015. NÃO OCORRÊNCIA. DANO MORAL. VALOR. REEXAME DE PROVAS. SÚMULA N. 7/STJ. NÃO PROVIMENTO.

Original: HABEAS CORPUS SUBSTITUTIVO DE RECURSO PRÓPRIO. TRÁFICO DE DROGAS E ASSOCIAÇÃO PARA O TRÁFICO. PRISÃO PREVENTIVA. FUNDAMENTAÇÃO. PERICULOSIDADE. PAPEL DE RELEVÂNCIA DO ESQUEMA CRIMINOSO. PACIENTE MÃE DE CRIANÇA MENOR DE 12 ANOS. PRISÃO DOMICILIAR. RISCO DE REITERAÇÃO. BENEFÍCIO CONCEDIDO EM DATA RECENTE. DESCUMPRIMENTO E APREENSÃO POSTERIOR DE DROGA NA RESIDÊNCIA (50g DE CRACK). SITUAÇÃO EXCEPCIONALÍSSIMA. AUSÊNCIA DE CONSTRANGIMENTO ILEGAL. HABEAS CORPUS NÃO CONHECIDO.
Gerada: HABEAS CORPUS SUBSTITUTIVO DE RECURSO ORDINÁRIO. TRÁFICO DE DROGAS E ASSOCIAÇÃO PARA O TRÁFICO. PRISÃO PREVENTIVA. FUNDAMENTAÇÃO. PERICULOSIDADE. GRAVIDADE CONCRETA. ENVOLVIMENTO COM ORGANIZAÇÃO CRIMINOSA. PRISÃO DOMICILIAR. PACIENTE MÃE DE CRIANÇA MENOR DE 12 ANOS. SITUAÇÃO EXCEPCIONALÍSSIMA. HABEAS CORPUS NÃO CONHECIDO.

Figura 22 – Exemplos de verbetações geradas com decodificação gulosa.

A.2 Verbetações Geradas com Decodificação com Amostragem

A.2.1 Top-15

A.2.2 Top-50

A.2.3 Top-100

Original: EMBARGOS DE DECLARAÇÃO NOS EMBARGOS DE DIVERGÊNCIA NO RECURSO ESPECIAL. RESÍDUO DE 3,17%. MP N. 2.225-45/2001. LIMITAÇÃO TEMPORAL. IMPOSSIBILIDADE. REESTRUTURAÇÃO DA CARREIRA ANTERIOR AO TRÂNSITO EM JULGADO. AUSÊNCIA DE PREVISÃO NO TÍTULO EXECUTIVO. OMISSÃO QUANTO À FIXAÇÃO DE HONORÁRIOS ADVOCATÍCIOS SUPRIDA. ALEGAÇÃO DE APLICAÇÃO DA SÚMULA 7. NÃO OCORRÊNCIA.
Gerada: EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO RECURSO ESPECIAL. PROCESSUAL CIVIL. AÇÃO DE COBRANÇA. SEGURO HABITACIONAL. COISA JULGADA. OMISSÃO. INEXISTÊNCIA. EMBARGOS DE DECLARAÇÃO DA UNIVERSIDADE FEDERAL DE SANTA MARIA REJEITADOS.

Original: AGRAVO REGIMENTAL NO AGRAVO REGIMENTAL NO HABEAS CORPUS. CRIME AMBIENTAL. ARTIGOS 40 E 63 DA LEI N. 9.605/1998. PRESCRIÇÃO DA PRETENSÃO PUNITIVA. CRIMES COMETIDOS ANTES DE 2010. TRANSCURSO DO LAPSO TEMPORAL ENTRE A DATA DOS FATOS E O RECEBIMENTO DA DENÚNCIA. ARTIGO 48 DA LEI AMBIENTAL. PRESCRIÇÃO RETROATIVA. OCORRÊNCIA. AGRAVO DESPROVIDO.
Gerada: AGRAVO REGIMENTAL NO RECURSO ESPECIAL. CRIME AMBIENTAL. PRESCRIÇÃO RETROATIVA. INAPLICABILIDADE DA REDAÇÃO ANTERIOR À LEI N. 12.234/2010. PRESCRIÇÃO DA PRETENSÃO PUNITIVA. OCORRÊNCIA. AGRAVO REGIMENTAL DESPROVIDO.

Original: PROCESSO PENAL. AGRAVO REGIMENTAL NO HABEAS CORPUS. TRÁFICO E ASSOCIAÇÃO PARA O TRÁFICO. PRISÃO PREVENTIVA. GRAVIDADE CONCRETA. MODUS OPERANDI. ORGANIZAÇÃO CRIMINOSA. MEDIDAS CAUTELARES DIVERSAS DA PRISÃO. IMPOSSIBILIDADE. PRISÃO DOMICILIAR. FILHOS MENORES. ENQUADRAMENTO NOS CASOS EXCEPCIONAIS QUE IMPEDEM O BENEFÍCIO. AGRAVO DESPROVIDO.
Gerada: AGRAVO REGIMENTAL NO HABEAS CORPUS. TRÁFICO DE DROGAS. PRISÃO PREVENTIVA. GRAVIDADE CONCRETA. REITERAÇÃO DELITIVA. MEDIDAS CAUTELARES DIVERSAS DA PRISÃO. IMPOSSIBILIDADE. PRISÃO DOMICILIAR. IMPOSSIBILIDADE. AGRAVO DESPROVIDO.

Original: AGRAVO INTERNO NO AGRAVO RECURSO ESPECIAL. INVIABILIDADE DE ANÁLISE DE OFENSA A RESOLUÇÕES. PLANO DE SAÚDE. TRATAMENTO MÉDICO DOMICILIAR (HOME CARE). RECUSA INDEVIDA À COBERTURA. AGRAVO NÃO PROVIDO.
Gerada: AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. PLANO DE SAÚDE. NEGATIVA DE COBERTURA. RESOLUÇÃO E CIRCULAR. ATOS NORMATIVOS QUE NÃO SE ENQUADRAM NO CONCEITO DE LEI FEDERAL. NORMA QUE NÃO SE ENQUADRA NO CONCEITO DE LEI FEDERAL. AGRAVO INTERNO NÃO PROVIDO.

Original: PROCESSO PENAL. AGRAVO REGIMENTAL DA DECISÃO QUE NEGOU PROVIMENTO AO RECURSO ORDINÁRIO EM HABEAS CORPUS. PEDIDO DE SUSTENTAÇÃO ORAL. IMPOSSIBILIDADE. AUSÊNCIA DE PREVISÃO REGIMENTAL. INÉPCIA DA DENÚNCIA. AUSÊNCIA DE JUSTA CAUSA PARA A AÇÃO PENAL. REGULARIDADE FORMAL DA PEÇA ACUSATÓRIA. LASTRO PROBATÓRIO IDÔNEO. TESES DE VIOLAÇÃO DA COISA JULGADA E DE LITISPENDÊNCIA. IMPROCEDÊNCIA. REVOLVIMENTO DE FATOS E PROVAS. IMPOSSIBILIDADE. AGRAVO REGIMENTAL DESPROVIDO.
Gerada: PROCESSO PENAL. AGRAVO REGIMENTAL DA DECISÃO QUE NEGOU PROVIMENTO AO RECURSO ORDINÁRIO. SUSTENTAÇÃO ORAL. IMPOSSIBILIDADE. AUSÊNCIA DE PREVISÃO REGIMENTAL. CORRUPÇÃO PASSIVA E LAVAGEM DE CAPITAIS. TRANCAMENTO DA AÇÃO PENAL. AUSÊNCIA DE JUSTA CAUSA. NÃO OCORRÊNCIA. ART. 41 DO CPP. PRINCÍPIOS CONSTITUCIONAIS. AMPLA DEFESA, DO CONTRADITÓRIO, DA INDIVIDUALIZAÇÃO DAS PENAS. DENÚNCIA. AUSÊNCIA DE FUNDAMENTAÇÃO. AGRAVO REGIMENTAL DESPROVIDO.

Figura 23 – Exemplos de verbetações geradas com decodificação gulosa.

Original: PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL RECURSO MANEJADO SOB A ÉGIDE DO NCPC. AÇÃO DE INDENIZAÇÃO. AUSÊNCIA DE FUNDAMENTOS CLAROS E CONCATENADOS. ILAÇÕES GENÉRICAS. DEFICIÊNCIA NA FUNDAMENTAÇÃO DO RECURSO. INCIDÊNCIA DA SÚMULA Nº 284 DO STF. DANO MORAL CONFIGURADO. REFORMA DO ENTENDIMENTO. INVIABILIDADE. INCIDÊNCIA DA SÚMULA Nº 7 DO STJ. QUANTUM INDENIZATÓRIO. RAZOABILIDADE. REEXAME DO ACERVO FÁTICO-PROBATÓRIO. IMPOSSIBILIDADE. APLICAÇÃO DA SÚMULA Nº 7 DO STJ. RECURSO PROTELATÓRIO. INCIDÊNCIA DA MULTA DO ART. 1.021, § 4º, DO NCPC. AGRAVO INTERNO NÃO PROVIDO, COM IMPOSIÇÃO DE MULTA.

1) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL RECURSO MANEJADO SOB A ÉGIDE DO NCPC. AÇÃO DE INDENIZAÇÃO POR DANOS MORAIS. FUNDAMENTAÇÃO DEFICIENTE. INCIDÊNCIA DA SÚMULA No 284 DO STF. REVISÃO. PRETENSÃO RECURSAL QUE ENVOLVE O REEXAME DE PROVAS. INCIDÊNCIA DA SÚMULA No 7 DO STJ AGRAVO INTERNO NÃO PROVIDO, COM IMPOSIÇÃO DE MULTA.

2) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL RECURSO MANEJADO SOB A ÉGIDE DO NCPC. RECURSO INTEMPESTIVO. AÇÃO INDENIZATÓRIA. FALHA NA PRESTAÇÃO DE SERVIÇOS. MATÉRIA JORNALÍSTICA. DEFICIÊNCIA NA FUNDAMENTAÇÃO DO RECURSO. INCIDÊNCIA DA SÚMULA No 284 DO STF. DANO MORAL. REVISÃO. PRETENSÃO RECURSAL QUE ENVOLVE O REEXAME DE PROVAS. INCIDÊNCIA DA SÚMULA No 7 DO STJ. QUANTUM INDENIZATÓRIO. RAZOABILIDADE. ERRO MATERIAL CONFIGURADO. EXORBITÂNCIA NÃO CIVIL.

3) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO RECURSO ESPECIAL. RECURSO MANEJADO SOB A ÉGIDE DO NCPC. AÇÃO DE INDENIZAÇÃO. DANO MORAL. VIOLAÇÃO DOS ARTS. 171, II E 161, § 1o, 489, II, E 1.022, I, DO NCPC. DEFICIÊNCIA NA FUNDAMENTAÇÃO. INCIDÊNCIA DA SÚMULA No 284 DO STF. ARBITRAMENTO. REFORMA DO ENTENDIMENTO. APLICAÇÃO DA SÚMULA No 7 DO STJ. QUANTUM INDENIZATÓRIO. PRETENSÃO RECURSAL QUE ENVOLVE O REEXAME DE PROVAS. INCIDÊNCIA DA SÚMULA No 7 DO STJ.

4) REC PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL RECURSO MANEJADO SOB A ÉGIDE DO NCPC. AÇÃO INDENIZATÓRIA. TRANSPORTE ORDINÁRIO. FRATURA MENSAL. DEFICIÊNCIA NA FUNDAMENTAÇÃO DO RECURSO. INCIDÊNCIA DA SÚMULA No 284 DO STF, POR ANALOGIA. ATOS ILÍCITOS. AUSÊNCIA DE DEMONSTRAÇÃO DEMONSTRAÇÃO DE OFENSA LITERAL E DE DANO MORAL. REFORMA DA MATÉRIA. INCIDÊNCIA DA SÚMULA No 7 DO STJ. DANO MORAL CONFIGURADO. RAZOABILIDADE E PROPORCIONALIDADE. REVISÃO.

5) DESC PROCESSO CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL RECURSO MANEJADO SOB A ÉGIDE DO NCPC. AÇÃO INDENIZATÓRIA. IMÓVEL RESIDENCIAL. DANOS MORAIS. EXISTÊNCIA DE ORDEM PÚBLICA. FUNDAMENTAÇÃO DEFICIENTE. ILEGITIMIDADE PASSIVA. INCIDÊNCIA DA SÚMULA No 284 DO STF. CERCEAMENTO DE DEFESA. INCIDÊNCIA DA SÚMULA No 7 DO STJ. QUANTUM INDENIZATÓRIO. REVISÃO. NECESSIDADE DE REEXAME DO ACERVO FÁTICO-PROBATÓRIO. INCIDÊNCIA DA SÚMULA No 7 DO STJ. DECISÃO MANTIDA.

Figura 24 – Visualização de cinco verbetações geradas utilizando amostragem top-15.

Original: PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO AGRAVO RECURSO ESPECIAL. OBSCURIDADE PARCIALMENTE CONFIGURADA. DEMAIS VÍCIOS APONTADOS. AUSÊNCIA DE VÍCIOS NO ACÓRDÃO EMBARGADO. REDISCUSSÃO. PRETENSÃO DE REEXAME. NÃO CABIMENTO. EMBARGOS DE DECLARAÇÃO PARCIALMENTE ACOLHIDOS SEM EFEITOS INFRINGENTES.

1) PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO AGRAVO RECURSO ESPECIAL AUSÊNCIA DE VÍCIOS NO ACÓRDÃO EMBARGADO. REDISCUSSÃO. PRETENSÃO DE REEXAME. NÃO CABIMENTO. EMBARGOS DE DECLARAÇÃO PARCIALMENTE ACOLHIDOS SEM EFEITOS INFRINGENTES.

2) PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO RECURSO ESPECIAL. SERVIDOR PÚBLICO. ANULAÇÃO DE PROCESSO ADMINISTRATIVO DISCIPLINAR. AUSÊNCIA DE VÍCIOS NO ACÓRDÃO EMBARGADO. REDISCUSSÃO. PRETENSÃO DE REEXAME. NÃO CABIMENTO. EMBARGOS DE DECLARAÇÃO PARCIALMENTE ACOLHIDOS SEM EFEITOS INFRINGENTES.

3) PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. ENUNCIADO ADMINISTRATIVO N. 3/ STJ. AUSÊNCIA DE VÍCIOS NO ACÓRDÃO EMBARGADO. REDISCUSSÃO. PRETENSÃO DE REEXAME. NÃO CABIMENTO. EMBARGOS DE DECLARAÇÃO PARCIALMENTE ACOLHIDOS.

4) PROCESSUAL CIVIL. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO AGRAVO RECURSO ESPECIAL ENUNCIADO ADMINISTRATIVO N. 3/ STJ. AUSÊNCIA DE VÍCIOS NO ACÓRDÃO EMBARGADO. REDISCUSSÃO. PRETENSÃO DE REEXAME. NÃO CABIMENTO. EMBARGOS DE DECLARAÇÃO PARCIALMENTE ACOLHIDOS SEM EFEITOS INFRINGENTES.

5) PROCESSUAL CIVIL. ENUNCIADO ADMINISTRATIVO No 03/STJ. EMBARGOS DE DECLARAÇÃO NO AGRAVO INTERNO NO CONFLITO DE COMPETÊNCIA. SERVIDOR PÚBLICO FEDERAL. AUSÊNCIA DE VÍCIOS NO ACÓRDÃO EMBARGADO. REDISCUSSÃO. PRETENSÃO DE REEXAME. NÃO CABIMENTO. EMBARGOS DE DECLARAÇÃO PARCIALMENTE ACOLHIDOS, SEM EFEITOS INFRINGENTES.

Figura 25 – Visualização de cinco verbetações geradas utilizando amostragem top-15.

Original: PENAL. AGRAVO REGIMENTAL NO RECURSO ESPECIAL. FURTO. NULIDADE DO RECONHECIMENTO FOTOGRÁFICO. PROVA CONFIRMADA EM JUÍZO. AFIRMAÇÃO DA VÍTIMA QUE O ACUSADO ERA SEU CONHECIDO. CONDENAÇÃO BASEADA EM OUTRAS PROVAS. AGRAVO REGIMENTAL NÃO PROVIDO.

1) AGRAVO REGIMENTAL NO HABEAS CORPUS SUBSTITUTIVO DE RECURSO EM SENTIDO ESTRITO SEM PRÉVIO RECONHECIMENTO FOTOGRÁFICO E PESSOAL. RECONHECIMENTO REALIZADO EM SEDE POLICIAL. INOBSERVÂNCIA DO PROCEDIMENTO PREVISTO NO ART. 226 DO CPP. NULIDADE. AUTORIA ESTABELECIDA COM BASE OUTROS ELEMENTOS PROBATÓRIOS. AGRAVO DESPROVIDO.

2) AGRAVO REGIMENTAL NO HABEAS CORPUS. ROUBO MAJORADO. RECONHECIMENTO PESSOAL E FOTOGRÁFICO. INOBSERVÂNCIA DO PROCEDIMENTO PREVISTO NO ART. 226 DO CPP. INOBSERVÂNCIA. AUTORIA ESTABELECIDA COM BASE OUTROS ELEMENTOS PROBATÓRIOS. INEXISTÊNCIA DE ILEGALIDADE. AUTORIA INDICADA NAS INSTÂNCIAS ORDINÁRIAS. DISTINGUISHING. NÃO OCORRÊNCIA. AGRAVO NÃO PROVIDO.

3) AGRAVO REGIMENTAL NO HABEAS CORPUS. ROUBO MAJORADO. RECONHECIMENTO PESSOAL REALIZADO EM SEDE POLICIAL. INOBSERVÂNCIA DO PROCEDIMENTO PREVISTO NO ART. 226 DO CPP. INVALIDADE DA PROVA CONFIGURADA COM BASE OUTROS ELEMENTOS PROBATÓRIOS PRODUZIDOS NA FASE JUDICIAL. AUTORIA ESTABELECIDA DE OFÍCIO. AGRAVO REGIMENTAL NÃO PROVIDO.

4) AGRAVO REGIMENTAL NO HABEAS CORPUS. ROUBO. RECONHECIMENTO FOTOGRÁFICO. INOBSERVÂNCIA DO PROCEDIMENTO PREVISTO NO ART. 226 DO CPP. AUTORIA DELITIVA. INOBSERVÂNCIA DOS PROCEDIMENTOS LEGAIS. MATÉRIA FÁTICO-PROBATÓRIA. AUTORIA ESTABELECIDA COM BASE OUTRAS PROVAS JUDICIAIS. AGRAVO NÃO PROVIDO.

5) AGRAVO REGIMENTAL NO HABEAS CORPUS. ROUBO MAJORADO. RECONHECIMENTO PESSOAL E FOTOGRÁFICO. INOBSERVÂNCIA DO PROCEDIMENTO PREVISTO NO ART. 226 DO CPP. AUTORIA ESTABELECIDA COM BASE OUTROS ELEMENTOS PROBATÓRIOS. AGRAVO NÃO PROVIDO.

Figura 26 – Visualização de cinco verbetações geradas utilizando amostragem top-15.

Original: AGRAVO REGIMENTAL NO HABEAS CORPUS. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. INDEFERIMENTO LIMINAR DO WRIT, PELA INCIDÊNCIA DA SÚMULA N. 691/STF. AUSÊNCIA DE FLAGRANTE ILEGALIDADE OU TERATOLOGIA. AGRAVO DESPROVIDO.

1) DIREITO PROCESSUAL PENAL. AGRAVO REGIMENTAL NO HABEAS CORPUS. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. INDEFERIMENTO LIMINAR DO WRIT, PELA INCIDÊNCIA DA SÚMULA N. 691/STF. AUSÊNCIA DE FLAGRANTE ILEGALIDADE OU TERATOLOGIA. SUSTENTAÇÃO ORAL. IMPOSSIBILIDADE. ART. 159 RISTJ. AGRAVO DESPROVIDO.

2) PROCESSUAL PENAL. AGRAVO REGIMENTAL NO HABEAS CORPUS. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. AGRAVO DESPROVIDO.

3) AGRAVO REGIMENTAL NO HABEAS CORPUS. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. INDEFERIMENTO LIMINAR DO WRIT, PELA INCIDÊNCIA DA SÚMULA N. 691/STF. AUSÊNCIA DE FLAGRANTE ILEGALIDADE OU TERATOLOGIA. SUSTENTAÇÃO ORAL. IMPOSSIBILIDADE. REGIMENTAL DESPROVIDO.

4) AGRAVO REGIMENTAL NO HABEAS CORPUS. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. AGRAVO DESPROVIDO.

5) AGRAVO REGIMENTAL NO HABEAS CORPUS. INEXISTÊNCIA DE NOVOS ARGUMENTOS APTOS A DESCONSTITUIR A DECISÃO AGRAVADA. INDEFERIMENTO LIMINAR DO WRIT, PELA INCIDÊNCIA DA SÚMULA N. 691/STF. AUSÊNCIA DE FLAGRANTE ILEGALIDADE OU TERATOLOGIA. SUSTENTAÇÃO ORAL. IMPOSSIBILIDADE. ART. 159, IV, RISTJ. DECISÃO MANTIDA. AGRAVO DESPROVIDO.

Figura 27 – Visualização de cinco verbetações geradas utilizando amostragem top-50.

Original: PROCESSUAL CIVIL E TRIBUTÁRIO. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. OFENSA AO ART. 1.022 DO CÓDIGO FUX NÃO CARACTERIZADA. DIVERGÊNCIA ACERCA DO SENTIDO, DA SUFICIÊNCIA E DA EFICÁCIA DA PROVA ENCARTADA AOS AUTOS. ERROR IN PROCEDENDO NÃO CONFIGURADO. AGRAVO INTERNO DA EMPRESA NÃO PROVIDO.

1) PROCESSUAL CIVIL E TRIBUTÁRIO. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. EXECUÇÃO FISCAL. IPTU. ICMS. TRIBUTO CONCEDIDO PELO TRIBUNAL DE ORIGEM, DIANTE DA AUSÊNCIA DOS REQUISITOS DA TUTELA PROVISÓRIA. INVIABILIDADE DE EXAME O PREJUÍZO QUANTO A RECOLHIMENTO DE ICMS DE BOA-FÉ. DISSÍDIO QUANTO À REALIDADE DA PROVA ERROR IN JUDICANDO E NÃO ERROR IN PROCEDENDO. AGRAVO INTERNO DA EMPRESA NÃO PROVIDO.

2) PROCESSO CIVIL. AGRAVO INTERNO NO PEDIDO DE TUTELA PROVISÓRIA. NEGATIVA DE PRESTAÇÃO JURISDICIONAL. NÃO OCORRÊNCIA. ALEGAÇÃO DE IRREGULARIDADE DO PRODUTO. ERROR IN JUDICIANDO NÃO CARACTERIZADO. VERIFICAÇÃO QUE É POSSÍVEL O ERROR IN PROCEDENDO. TUTELA DE URGÊNCIA. NÃO OCORRÊNCIA. AGRAVO INTERNO DA EMPRESA NÃO PROVIDO.

3) PROCESSUAL CIVIL E TRIBUTÁRIO. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. ENUNCIADO ADMINISTRATIVO No 3 DO STJ. VIOLAÇÃO AO ART. 1.022 DO CÓDIGO FUX. INOCORRÊNCIA. ACÓRDÃO RECORRIDO QUE, EM FACE DAS PREMISSAS FÁTICAS DOS AUTOS, CONCLUIU PELA COMPROVAÇÃO DE QUE AS ATIVIDADES REALIZADAS NÃO SÃO PROVEDORES À INCIDÊNCIA DE ICMS, DADOS, PELA CONTRIBUINTE, NÃO PODEM SER CONSIDERADAS IRREDUTIÁRIAS, ENQUANTO NÃO COMPROVADOS OS REQUISITOS NECESSÁRIOS PARA A CONCESSÃO DA PROCESSO CIVIL.

4) AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. NEGATIVA DE PRESTAÇÃO JURISDICIONAL. NÃO OCORRÊNCIA. TUTELA PROVISÓRIA JULGADA IMPROCEDENTE PELO TRIBUNAL A QUO. EXCLUSÃO DO ICMS DAS OPERAÇÕES DE VERBA SALARIAL E DE INADIMPLEMENTO. INEXISTÊNCIA DOS REQUISITOS PARA A CONCESSÃO DA TUTELA PROVISÓRIA. AUSÊNCIA DE DEMONSTRAÇÃO DA NECESSIDADE DE CÔMPUTO DA PROVA. POSSIBILIDADE DE CONFIGURAÇÃO (ERROR IN JUDICIANDO). AGRAVO INTERNO DA EMPRESA NÃO PROVIDO.

5) TRIBUTÁRIO E PROCESSUAL CIVIL. AGRAVO INTERNO NO RECURSO ESPECIAL. NEGATIVA DE PRESTAÇÃO JURISDICIONAL. NÃO OCORRÊNCIA. INDEFERIMENTO DE PEDIDO TUTELA PROVISÓRIA. IDENTIDADE FÁTICA QUE, NOS AUTOS DENOMINADA NULIDADE DA PROVA EXTRAJUDICIAL DO FATO CONSTITUTIVO, PODE CONFERIR ERROR IN JUDICANDO E NÃO VERIFICAR, AINDA, NÃO ESTAR CONDENADO A DESPEITO DE DESCONSTITUIR ACÓRDÃO QUE, APESAR DOS ELEMENTOS FÁTICOS E PROBATÓRIOS DOS AUTOS, CONCLUIU QUE A LIDE TRIBUTADA NA INTERPRETAÇÃO.

Figura 28 – Visualização de cinco verbetações geradas utilizando amostragem top-50.

Original: PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. AGRAVO EM RECURSO ESPECIAL INTERPOSTO CONTRA DECISÃO PUBLICADA NA VIGÊNCIA DO CPC/2015. RECURSO INTERPOSTO VIA E-MAIL. INADMISSIBILIDADE. NÃO EQUIPARAÇÃO AO FAC-SIMILE. INTEMPESTIVIDADE. PRECEDENTES DO SUPERIOR TRIBUNAL DE JUSTIÇA. AGRAVO INTERNO NÃO PROVIDO.

1) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL SUBMISSÃO À REGRA PREVISTA NO ENUNCIADO ADMINISTRATIVO 03/STJ. RECURSO INTERPOSTO VIA CORREIO ELETRÔNICO. INTEMPESTIVIDADE. PRECEDENTES DO STJ. AGRAVO INTERNO IMPROVIDO.

2) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. RECURSO ESPECIAL INTERPOSTO VIA CORREIO ELETRÔNICO. INADMISSIBILIDADE. NÃO VINCULAÇÃO À ASSINATURA ELETRÔNICA. PRESCRIÇÃO DA PRETENSÃO EXECUTÓRIA. NÃO CONHECIMENTO.

3) AGRAVO INTERNO IMPROVIDO. TRIBUTÁRIO E PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. EXPEDIENTE AVULSO. INTEMPESTIVIDADE. PRAZO RECURSAL DE 15 DIAS ÚTEIS CORRIDOS. INAPLICABILIDADE EM CONTACORRENTE. PRECEDENTES DO STJ. AGRAVO INTERNO IMPROVIDO.

4) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. RECURSO FAC-SÍMILE. INTEMPESTIVIDADE. CONTRATO DE ADESÃO. ASSINATURA ELETRÔNICA. INADMISSIBILIDADE. PRECEDENTES DO STJ. AGRAVO INTERNO IMPROVIDO.

5) ADMINISTRATIVO E PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. RECURSO INTERPOSTO VIA CORREIO ELETRÔNICO. NÃO CABIMENTO. PRECEDENTES DO STJ. AGRAVO INTERNO IMPROVIDO.

Figura 29 – Visualização de cinco verbetações geradas utilizando amostragem top-50.

Original: PENAL E PROCESSUAL PENAL. AGRAVO REGIMENTAL NO AGRAVO EM RECURSO ESPECIAL. HOMICÍDIO. ART. 478, I, DO CPP. ROL TAXATIVO. EXPOSIÇÃO DOS ANTECEDENTES CRIMINAIS DO RÉU. POSSIBILIDADE. AGRAVO REGIMENTAL DESPROVIDO.

1) AGRAVO REGIMENTAL NO AGRAVO EM RECURSO ESPECIAL. ART. 478, I, DO CPP. ROL TAXATIVO. IMPOSSIBILIDADE DE AMPLIAÇÃO. EXPOSIÇÃO DOS ANTECEDENTES. NULIDADE. INOCORRÊNCIA. AGRAVO DESPROVIDO.

2) AGRAVO REGIMENTAL NO RECURSO ESPECIAL. HOMICÍDIO E LESÃO CORPORAL. 478, I, DO CPP. ROL TAXATIVO. MÉRITO. ANTECEDENTES. ARGUIÇÃO PARA O PLENÁRIO. AUSÊNCIA DE NULIDADE. EXPOSIÇÃO À PLENÁRIO DO RÉU. POSSIBILIDADE. MANUTENÇÃO DA DECISÃO AGRAVADA. AGRAVO DESPROVIDO.

3) AGRAVO REGIMENTAL NO RECURSO ESPECIAL. HOMICÍDIO CULPOSO. SENTENÇA CONDENATÓRIA. PLEITO DE RECONHECIMENTO DA MATERIALIDADE E AUTORIA. VEDAÇÃO INDEVIDA. ART. 478, I, DO CPP. ROL TAXATIVO. AGRAVO REGIMENTAL

4) AGRAVO REGIMENTAL NO RECURSO ESPECIAL. ART. 478, I, DO CPP. ROL TAXATIVO. IMPOSSIBILIDADE DE INTERPRETAÇÃO AMPLIATIVA. VEDAÇÃO LEGAL. EXPOSIÇÃO DOS ANTECEDENTES NA PLENÁRIO JÚRI. NULIDADE. NÃO OCORRÊNCIA. AGRAVO REGIMENTAL DESPROVIDO.

5) PENAL. AGRAVO REGIMENTAL NO RECURSO ESPECIAL. TRIBUNAL DO JÚRI. INCIDÊNCIA DA SÚMULA 443 DO STJ. ART. 478, I, DO CPP. ROL TAXATIVO. IMPOSSIBILIDADE DE INTERPRETAÇÃO AMPLIATIVA. ANTECEDENTES. EXPOSIÇÃO DOS ANTECEDENTES SOB O PLENÁRIO. NULIDADE. NÃO OCORRÊNCIA. AGRAVO DESPROVIDO.

Figura 30 – Visualização de cinco verbetações geradas utilizando amostragem top-100.

Original: HABEAS CORPUS SUBSTITUTIVO DE RECURSO PRÓPRIO. NÃO CABIMENTO. EXECUÇÃO PENAL. UNIFICAÇÃO DE PENAS. CONTINUIDADE DELITIVA. NÃO CONFIGURAÇÃO DOS REQUISITOS DO ART. 71 DO CP. AUSÊNCIA DE IDENTIDADE DE DESÍGNIOS. REANÁLISE FÁTICO-PROBATÓRIA. CONSTRANGIMENTO ILEGAL NÃO CONFIGURADO. HABEAS CORPUS NÃO CONHECIDO.

1) PENAL. HABEAS CORPUS SUBSTITUTIVO DE RECURSO PRÓPRIO. TRÁFICO DE ENTORPECENTES. DOSIMETRIA. CONTINUIDADE DELITIVA RECONHECIDA PELAS INSTÂNCIAS ORDINÁRIAS. CRIME ÚNICO. AUSÊNCIA DE CONSTRANGIMENTO ILEGAL. AMPLO REVOLVIMENTO FÁTICO-PROBATÓRIO INCOMPATÍVEL COM A VIA ESTREITA DO WRIT. HABEAS CORPUS NÃO CONHECIDO.

2) PENAL. HABEAS CORPUS SUBSTITUTIVO DE RECURSO PRÓPRIO. NÃO CABIMENTO. EXECUÇÃO PENAL. AMEAÇA. IMPOSSIBILIDADE DE INVERSÃO DAS CONDIÇÕES LEGAIS CUJAS ATIVIDADES FORMAS IGUALMENTE DELITIVAS. RECONHECIMENTO DA CONTINUIDADE DELITIVA. IMPOSSIBILIDADE. REQUISITOS NÃO ATENDIDOS. REVOLVIMENTO DE MATÉRIA FÁTICO-PROBATÓRIA. HABEAS CORPUS NÃO CONHECIDO.

3) PENAL E PROCESSUAL PENAL. HABEAS CORPUS. SUBSTITUTIVO DE RECURSO PRÓPRIO. NÃO CABIMENTO. EXECUÇÃO PENAL. CONTINUIDADE DELITIVA. CRIME ÚNICO. REQUISITOS NÃO EXIGIDOS. REVOLVIMENTO FÁTICO-PROBATÓRIO INCOMPATÍVEL COM A VIA ESTREITA DO HABEAS CORPUS. HABEAS CORPUS NÃO CONHECIDO.

4)PENAL. HABEAS CORPUS SUBSTITUTIVO DE RECURSO PRÓPRIO. NÃO CABIMENTO. EXECUÇÃO PENAL. CRIME ÚNICO OU FIGURA DA CONTINUIDADE DELITIVA. DELITOS AUTÔNOMOS NÃO RECONHECIDOS. NECESSIDADE DE AMPLO REEXAME DE PROVAS. HABEAS CORPUS NÃO CONHECIDO.

5) PENAL. HABEAS CORPUS SUBSTITUTIVO DE RECURSO ORDINÁRIO. NÃO CABIMENTO. ROUBO MAJORADO. CRIME ÚNICO OU FIGURA DA CONTINUIDADE DELITIVA. AUSÊNCIA DE REQUISITOS. HABITUALIDADE CRIMINOSA. REVOLVIMENTO FÁTICO-PROBATÓRIO INCOMPATÍVEL COM A VIA ESTREITA DO HABEAS CORPUS. CONSTRANGIMENTO ILEGAL NÃO CONSTATADO. HABEAS CORPUS NÃO CONHECIDO.

Figura 31 – Visualização de cinco verbetações geradas utilizando amostragem top-100.

Original: PROCESSUAL CIVIL E ADMINISTRATIVO. AGRAVO INTERNO NO RECURSO ESPECIAL. VIOLAÇÃO DOS ARTIGOS 489 E 1.022 DO CPC/2015 NÃO CONFIGURADA. JULGAMENTO EXTRA PETITA. NÃO OCORRÊNCIA. PRECEDENTES. COISA JULGADA. LIMITES DO TÍTULO EXECUTIVO. REEXAME DE FATOS E PROVAS. IMPOSSIBILIDADE. SÚMULA 7/STJ. ILEGITIMIDADE DOS SUBSTITUÍDOS EM FACE DA CONDENAÇÃO HONORÁRIOS ADVOCATÍCIOS. AUSÊNCIA DE COMANDO NORMATIVO NOS DISPOSITIVOS INDICADOS COMO MALFERIDOS. DEFICIÊNCIA NA ARGUMENTAÇÃO RECURSAL. SÚMULA 284/STF.

1) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. VIOLAÇÃO DOS ARTS. 489 E 1.022 DO CPC/2015. NÃO OCORRÊNCIA. JULGAMENTO EXTRA PETITA. CERCEAMENTO DE DEFESA. NÃO OCORRÊNCIA. ALCANCE DO TÍTULO EXECUTIVO. AFASTAMENTO. SÚMULA 7 DO STJ. DEFICIÊNCIA DE ARGUMENTAÇÃO RECURSAL. SÚMULA 284 DO STF.

2) PROCESSUAL CIVIL E ADMINISTRATIVO. AGRAVO INTERNO NO RECURSO ESPECIAL. VIOLAÇÃO DOS ARTIGOS 489 E 1.022 DO CPC/2015 NÃO CONFIGURADA. JULGAMENTO EXTRA PETITA. NÃO OCORRÊNCIA. CONTROVÉRSIA SOBRE O ALCANCE DO TÍTULO EXECUTIVO. REVISÃO. IMPOSSIBILIDADE. SÚMULA 7/STJ. ARGUMENTAÇÃO DEFICIENTE. SÚMULA 284/STF.

3) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL. VIOLAÇÃO DOS ARTIGOS 489 E 1.022 DO CPC/2015. NÃO OCORRÊNCIA. JULGAMENTO EXTRA PETITA. NÃO CONFIGURAÇÃO. ALCANCE DO TÍTULO EXECUTIVO. ACÓRDÃO RECORRIDO. REVISÃO. IMPOSSIBILIDADE. REEXAME DE PROVAS. SÚMULA 7/STJ. AUSÊNCIA DE COMANDO NORMATIVO NO DISPOSITIVO INDICADO. SÚMULA 284/STF. AGRAVO INTERNO NÃO PROVIDO.

4) PROCESSUAL CIVIL. AGRAVO INTERNO NO AGRAVO EM RECURSO ESPECIAL AUSÊNCIA DE VIOLAÇÃO DOS ARTIGOS 489 E 1.022 DO CPC/2015. JULGAMENTO ULTRA PETITA. NÃO CONFIGURADA. CANDIDATO APROVADO EM HOSPITAL. TÍTULO EXECUTIVO. ALCANCE. TÍTULO ABRANGIDO. REVISÃO. IMPOSSIBILIDADE. SÚMULA 7/STJ. AUSÊNCIA DE COMANDO NORMATIVO NOS DISPOSITIVOS INDICADOS. SÚMULA 284/STF.

5) PROCESSUAL CIVIL. AGRAVO INTERNO PETIÇÃO NO AGRAVO EM RECURSO ESPECIAL. VIOLAÇÃO DOS ARTS. 489 E 1.022 DO CPC/2015. NÃO OCORRÊNCIA. JULGAMENTO EXTRA PETITA AFASTADO. GRUPAMENTO ECONÔMICO. REQUISITOS. TÍTULO EXECUTIVO. ALCANCE. TÍTULO EXEQUENDO. NECESSIDADE DE REEXAME DE FATOS E PROVAS. SÚMULA 7/STJ. DEFICIÊNCIA NA ARGUMENTAÇÃO RECURSAL. SÚMULA 284/STF.

Figura 32 – Visualização de cinco verbetações geradas utilizando amostragem top-100.


ICMC
USP
SÃO CARLOS