







Enhancing Brazilian Legal Information Retrieval: An Automated Keyphrase Generation

Breno O. Funicheli   [University of São Paulo, São Carlos, Brazil | bfunicheli@usp.br]
Kenzo Sakiyama  [University of São Paulo, São Carlos, Brazil | mkenzosakiyama@usp.br]
Rodrigo Nogueira  [University of Campinas, Campinas, Brazil | rodrigo@neuromind.com]
Roseli A. F. Romero  [University of São Paulo, São Carlos, Brazil | rafrance@icmc.usp.br]

 Institute of Science, Math and Computing, São Paulo University, SP, 24210-590, Brazil.

Received: 28 March 2025 • Accepted: 17 July 2025 • Published: 24 October 2025

Abstract The volume of legal proceedings in Brazil has grown significantly in recent years, highlighting the potential for leveraging advances in Natural Language Processing (NLP) to automate tasks in the legal domain. This study explores text decoding methods for automating keyphrase generation—sequences of key terms that summarize legal documents. A sequence-to-sequence Transformer-based framework generates keyphrases using three decoding techniques: greedy, top-K, and top-p sampling. To evaluate the effectiveness of the generated keyphrases, we integrate them into legal document retrieval tasks using traditional Information Retrieval (IR) methods, such as TF-IDF and BM25. Our results, validated through IR metrics, demonstrate that greedy decoding produces high-quality keyphrases that closely align with those written by human specialists, achieving statistically significant improvements in retrieval performance. As part of this study, we introduce a new data set of Brazilian legal documents, including dates and pre-processed keyphrases, which allows reproducibility and supports further research on keyphrase generation and legal document retrieval tasks.

Keywords: Text Generation, Information Retrieval and Legal Texts.

1 Introduction

Brazilian judicial system has three instances of jurisdiction, guaranteeing the decision review Perlingeiro and Schmidt [2023]. The first instance is where most cases begin, with judges analyzing the evidence and issuing the initial sentence. The second instance, comprised of state courts of justice, reviews decisions rendered by the first instance courts. Meanwhile, the third instance is represented by the Superior Court of Justice, which standardizes case law and analyzes federal legal questions. The Supreme Federal Court, as the highest court in the country, adjudicates special appeals and constitutional matters Pacheco [2008].

In this context, the third instance is central as a pillar for the support of precedents. By issuing definitive decisions, for binding precedents that guide the application of the law throughout the national territory, its normative function is fundamental for the unification of jurisprudence, because these decisions are used to base other cases Tinarrage *et al.* [2024].

STJ is organized into three specialized sections, each with distinct areas of jurisdiction. The First Section handles public Law, including administrative, social security, and tax law, dealing primarily with state and public administration issues. The second Section is focused on private Law, covering civil, commercial, and family law, and addresses disputes between private parties and issues concerning contracts, obligations, and civil liability. The Third Section specializes in Criminal Law, overseeing criminal cases such as habeas corpus and criminal review Perlingeiro and Schmidt [2023].

Figure 1 shows the stages of legal proceedings at STJ.

The initial processing takes place with the receipt and filing of the case, those originating from other courts or tribunals. Next, screening is a step in which one analyzes if the process will be sent to judgment or immediately rejected. When the process is accepted it is distributed and the judgment process starts. The judgment step can produce interlocutory decisions, preliminary injunctions, and sentences. These decisions are published and stay subject to appeal, which is a step of the reply's first decision. Finally, a single judge or a collegiate can decide the final judgment.

Understanding the different types of documents generated in a legal proceeding is essential for finding precedents, as some decisions and documents carry less weight than others. For example, motions for clarification typically do not have a binding effect but rather serve to guide the specific case in which they were issued, as they are meant to clarify a previous ruling Pacheco [2008].

However, based on data from the National Council of Justice - *Conselho Nacional de Justiça* (CNJ) Conselho Nacional de Justiça - CNJ [23], there were 84 million cases in transit in the Brazilian judiciary at the end of 2024, indicating a 10% increase from the previous year. Besides this, Supremo Tribunal de Justiça (STJ) registered a 5 % growth in received processes in the first semester of 2024 Superior Tribunal de Justiça [2024].

Because of the number of cases, finding relevant proceedings and judgment patterns is hard and time-consuming work for lawyers. In this context, retrieval information systems have a main role, in providing relevant proceedings. The important part of the text to provide relevant proceedings is the dockets, which consist of a special part of a decision

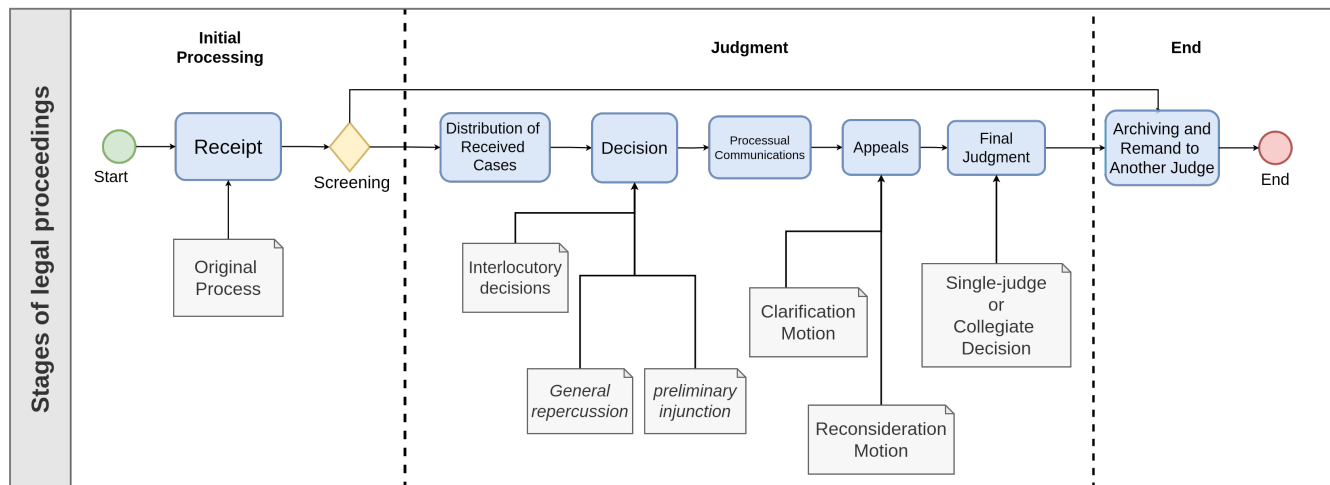


Figure 1. Stages of Legal Proceedings on STJ.

document, that aims to summarize a legal case. They are used in courts around Brazil and are designed to provide a summarised representation of judicial decisions. Figure 2 presents an example of a docket.

PROCESSUAL CIVIL. AGRAVO EM RECURSO ESPECIAL. ACÓRDÃO RECORRIDO. FUNDAMENTAÇÃO CONSTITUCIONAL. REVISÃO. IMPOSSIBILIDADE.
 1. À luz do art. 105, III, da Constituição Federal, o recurso especial não serve à revisão da fundamentação constitucional.
 2. Na hipótese dos autos, o acórdão a quo foi proferido com fundamento constitucional, uma vez que aponta na direção da violação pela lei estadual da proibição do efeito confiscatório da multa tributária e da proporcionalidade da penalidade (art. 150, IV, da CF).
 3. Agravo interno desprovido.

Figure 2. Example of a docket. Keyphrases are highlighted in bold text.

The dockets usually follow a pre-defined structure composed of two components: keyphrases and enumerated paragraphs. The keyphrases consist of a header present at the beginning of the docket. They are composed of sequences of capitalised key terms that highlight the key subjects present in the decision document. Note that each keyphrase contains one or more sentences, listing the key subjects of the docket. This header is created to improve the search and retrieval of jurisprudences (precedents) Guimarães and Santos [2016]. The enumerated paragraphs discuss the themes (or topics) present in the document.

Besides, certain types of judicial decisions and procedural documents often do not include dockets, as this synthesizes the key aspects of decisions of greater legal relevance. Examples include administrative orders, simple interlocutory decisions, requests for case review, and internal communications, which address procedural or administrative matters without directly impacting the case’s merits. Similarly, extrajudicial settlements and legal opinions typically do not contain a summary, as their content focuses on the parties’ agreement or the legal analysis provided, thus not requiring a formal synthesis of the points discussed.

Then, by analyzing the form and linguistic style found in the keyphrases, it is possible to note similarities between the writing of keyphrases and two Natural Language Processing (NLP) tasks: summarization and key terms extraction. How-

ever, keyphrases are not written fluidly and naturally such as summaries. In addition, most of the terms appearing in their text are not present in the remainder of the docket which originated the keyphrase, which makes it difficult to treat its writing as an extractive task.

Given the predictable structure and availability of dockets, it would be possible to prepare input-output pairs to generate keyphrases using the enumerated paragraphs as inputs, by employing a supervised approach. Transformers, such as GPT Radford *et al.* [2019], were already proven effective in various text-to-text generative Natural Language Processing (NLP) tasks Floridi and Chiriatti [2020] (such as translation, question answering and summarization). Also, the availability of pre-trained language models Carmo *et al.* [2020]; Scao *et al.* [2022]; Zhang *et al.* [2022] presents many opportunities to automate NLP tasks.

In our previous work Sakiyama *et al.* [2023a], we investigated the usage of state-of-the-art generative Transformers to automate the writing of keyphrases. Specifically, we analyzed text decoding methods to generate keyphrases that aid retrieval in the legal domain. This study was unprecedented in Brazil and can be widely used to automate keyphrase generation in courts in the country.

In this work, we extend the previous work, presenting a qualitative analysis and discussion of the used data (dockets) and further discussing the experiments that exemplify the usage Transformers to automate the writing of keyphrases. In addition, we provide the dockets corpus used for our analysis as a public dataset, aiming to encourage further research.

The main contributions of this work are:

1. Investigation of a novel approach to generate keyphrases from Brazilian dockets, using a sequence-to-sequence Transformer;
2. Comparison of three different text decoding methods for the proposed task (greedy and sampling methods);
3. Provide the quantitative and qualitative analysis of the generated keyphrases;
4. We provide a dataset, enabling reproducibility and further analysis.

This paper is organized as follows. Section 2 presents related works. Section 3 discussed the methodology applied

for the keyphrase generation. Next, Section 4 presents the experiments performed and discusses the obtained results. Finally, Section 5, presents conclusions and future works.

2 Related Works

In this Section, we will present studies related to the main objectives of this proposal. The Transformer Vaswani *et al.* [2017] consists of a deep neural network architecture that achieved the state of the art in several NLP tasks. It consists of an encoder-decoder architecture used originally for translation. However, the context-aware representations generated by the model can be used in diverse tasks.

Following the success of the Generative Pretrained Transformers (GPT) models Radford *et al.* [2019]; Zhang *et al.* [2022]; Scao *et al.* [2022], there is a predominance of decoder-only models in NLP tasks that can be approached as text generation (such as question answering and summarization) Floridi and Chiriatti [2020]. In addition, recent studies showed the great potential in using such models in zero and few-shot scenarios Brown *et al.* [2020]. Other studies Raffel *et al.* [2020]; Xue *et al.* [2020] investigate text generation using the full Transformer architecture (encoder-decoder) for some NLP tasks. The T5 Raffel *et al.* [2020] Transformer proposes the unification of a series of NLP tasks in a single text-to-text framework and Xue *et al.* [2020] expanded the original work to add multilingual support.

Although the presented Transformer approaches for text generation are different in terms of the architecture and scale (number of parameters), they all deal with common issues concerning the quality of the artificially generated text. Generated texts are often simplistic, inconsistent, or end up being repetitive Holtzman *et al.* [2019]. There is also the possibility of hallucination, generating contradictory texts, meaningless and without foundation or evidence Ji *et al.* [2022].

In order to mitigate the challenges (repetitive and predictable texts), there were initiatives aimed at making text generation non-deterministic Holtzman *et al.* [2019]; Fan *et al.* [2018]. Such proposals arise as an alternative to simpler text generation methods (also called greedy decoding), arguing that choosing most probable words (or tokens) is one of the main causes of repetitive texts.

Another example of a study aimed at mitigating repetitive texts is the work from Suet *et al.* [2022]. Proposed in 2022, the contrastive-search consists of a modification in the choosing words (or tokens) predicted by a textual generator, which aims to increase the variability of the text while maintaining its coherence. For this purpose, the authors suggest penalizing, during decoding or the language unsupervised model training, the softmax function scores of the most likely tokens by their similarity to other tokens within the context. The importance given to the similarity is controlled by a parameter α .

At last, we will present examples of studies employing Transformers to generate text in the legal domain. Keyphrases such as the ones used in this work are exclusive to Brazil, and in the best of our knowledge, this is the first depth study of decoding methods for Brazilian keyphrase generation. Feijo and Moreira Feijo and Moreira [2019] and Yoon *et al.* Yoon

et al. [2022] applied Transformer models to summarise rulings from the Brazilian Supreme Court and Korean legal cases, respectively. Peric *et al.* Peric *et al.* [2020] proposed Transformer models to generate opinions about legal cases originating in the *U.S Circuit Court*, by employing an encoder-decoder architecture.

Huanget *et al.* Huang *et al.* [2021] proposed a solution to automate the Legal Judgment Prediction (PJL) subtasks using the T5 text-to-text framework. At last, Althammer *et al.* Althammer *et al.* [2021] investigated the use of summaries (generated by Transformer) as part of an information retrieval pipeline for the legal domain as part of the 2021 Competition on Legal Information Extraction/Entailment (COLIEE).

3 Methodology

The methodology used in this work is composed of: I) Data Collection and Preprocessing, II) Keyphrase Generator Training, III) Decoding Methods Evaluation, and IV) Qualitative Analysis. We discuss these components below. Figure 3 illustrates these key stages of our methodology.

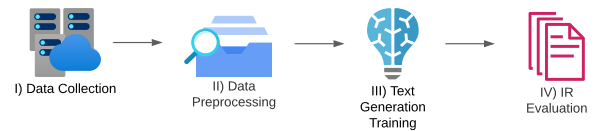


Figure 3. Overview of the keyphrase generation and evaluation pipeline.

3.1 Data Collection and Preprocessing

In 2022, the Brazilian *Supremo Tribunal de Justiça* (STJ) - Supreme Court of Justice made available the *Dados Abertos*¹ platform. The platform consists of a public website, sharing legal decisions from various courts in Brazil. The published documents comprise a large variety of topics in Brazil’s legal domain, such as crimes in general, commerce, taxes, etc. We collected a total of 712,161 documents from the platform in August 2022.

After the data collection, we extracted the dockets from the document’s metadata and preprocessed the text of the decisions. We removed duplicated examples and removed URLs from the text. 111,964 dockets remained after the preprocessing described. This amount is justified, because most of the decisions do not have summary statements, such as monocratic decisions, that is, those issued by a single judge. With the remaining examples, we extracted the keyphrases and enumerated paragraphs from the dockets, identifying and extracting capitalized sentences present in the header of the collected decisions. By extracting the inputs (enumerated paragraphs) and expected outputs (keyphrases), the original keyphrases (written by specialists) compose the reference set used for supervised training and evaluation.

To gain an initial qualitative insight into the lexical composition of the dataset, we generated a word cloud of the most frequent keyphrases, as shown in Figure 4. This visualization highlights recurring legal expressions such as *agravo*

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

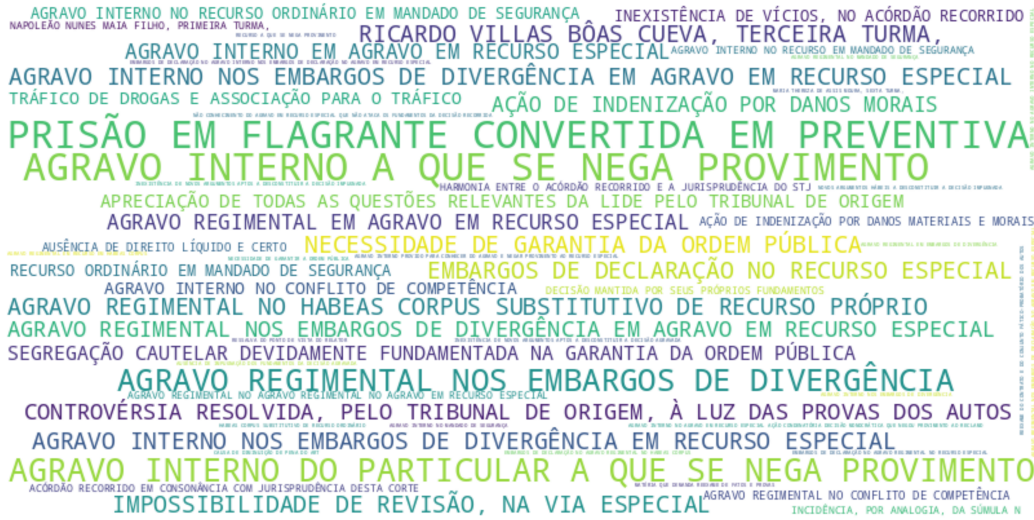


Figure 4. Word cloud illustrating the most frequent keyphrases extracted from the dataset. Larger font sizes indicate higher frequency.

interno, *embargos de declaração*, *prisão em flagrante*, and *recurso especial*, which are central to the procedural language of Brazilian legal documents. The prominence of these terms reflects their institutionalized role in summarizing court decisions and categorizing case types. These expressions are handy because they correspond to formal document categories and often appear in the docket headers, functioning as standardized descriptors that enhance legal context recognition and facilitate information retrieval.

As shown in Figure 5, the ranked frequencies of keyphrases follow a Zipfian distribution Zipf [1949], with a power-law exponent $\gamma \approx 2.22$, consistent with known patterns in natural language corpora Piantadosi [2014]; Li [2002]. The power-law exponent γ was estimated using Maximum Likelihood Estimation (MLE) over the tail of the distribution, considering only keyphrases with frequency greater than or equal to a minimum threshold $x_{\min} = 10$, following the method proposed by Clausen *et al.* [2009].

This statistical regularity suggests that the dataset exhibits the typical long-tailed behavior expected in linguistic resources, where a few terms dominate in frequency while many others occur rarely. Rather than the presence of highly frequent keyphrases, the use of standardized legal terminology that ensures consistency and clarity across judicial documents reflects the use of standardized terminology. This reinforces the observation that legal keyphrases are not only semantically standardized but also structurally concentrated, facilitating categorization and retrieval in legal information systems Saravanan *et al.* [2009a,b]; Schweighofer [2010].

Figure 6 (a) shows some descriptive analyses of the documents that remained after filtering. It is observed that most of the documents are from the years 2019 to 2024. In addition, we can see that the group decisions sections are balanced in these years. Figure 6 (c) presents a cumulative graph where we can see that approximately 60 % of documents are interlocutory decisions, and when we compare this information

with the graph (d), it is noted that interlocutory decisions have a mean length of all dockets. Another distribution visualization is seen in Figure (e), the boxplots display the distribution of docket lengths across types of appeals, highlighting differences in median, interquartile range, and outliers for each category. Special appeals and habeas corpus cases stand out with the highest median lengths and substantial variability, indicating that these cases often involve more intricate legal issues or require extensive deliberation, leading to prolonged case handling.

Finally, the last graph demonstrates that most decisions have less than 8 keyword phrases in a docket, besides all kinds of appeals showing outliers in the number of cases. These counts reveal that besides the length of the sentence, the number of phrase keywords is in a similar number. With these analyses, the results presented throughout this paper can be more clear to understand.

As a final preprocessing step, we divided the corpus (111,964 examples) into training (70%), validation (10%), and test (20%) splits. From the training set examples, we observed that enumerated paragraphs and keyphrases have a mean of 203.26 and 55.84 space-separated tokens, respectively. We used the splits to train and evaluate a supervised Deep Learning text generator. The dataset used in this work is publicly available on Hugging Face ².

documents comprise a large variety of topics in Brazil’s legal

3.2 Keyphrase Generation

This section describes the methodology employed for keyphrase training and generation.

²https://huggingface.co/datasets/bfunicheli/stj_docket_generation

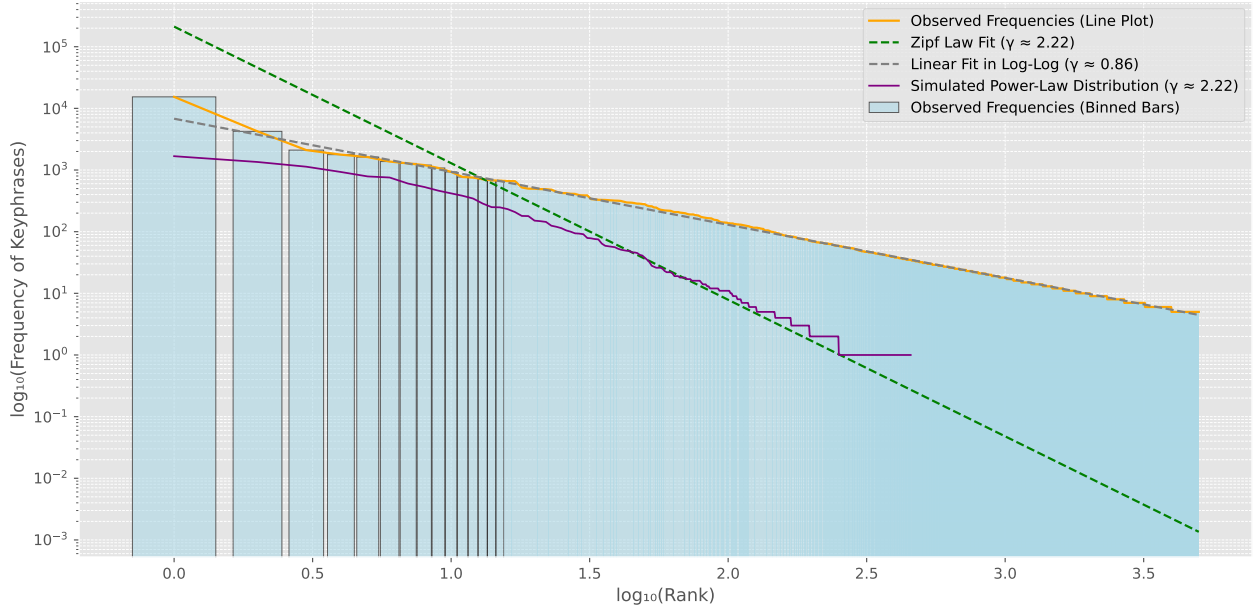


Figure 5. Log-log plot of the keyphrase frequency distribution in the dataset. The orange line represents the empirical frequency of the most common keyphrases, plotted against their rank. The shaded blue bars show the same data as a histogram, helping to visualize the relative density across ranks. The dashed green line corresponds to the Zipfian maximum likelihood estimation (MLE) with a power-law exponent $\gamma \approx 2.22$, indicating that the data follows a typical long-tailed distribution. The dashed gray line shows a linear regression in log-log space, which approximates the tail behavior but tends to underestimate the head. Finally, the purple line displays a simulated distribution generated from a power-law model with the same estimated exponent ($\gamma \approx 2.22$), validating the fit visually. Taken together, these elements suggest that keyphrase frequencies follow a Zipf-like distribution, which is characteristic of natural language corpora.

3.2.1 Transformers for text generation

Based on the dockets collected, we noted that most of the terms in the keyphrases are not directly present in the dockets. By further analyzing examples from the validation set, we noted that only $\sim 10\%$ of the terms present in the keyphrases are, in fact, present in the input text. Thus, we decided to approach writing keyphrases as generation rather than extraction of text. For this purpose, a sequence-to-sequence (or text-to-text) Transformer model was chosen.

We choose PTT5 Carmo *et al.* [2020] as our keyphrase generator. PTT5 was pretrained in a large Brazilian Portuguese corpus (*brWaC* Wagner Filho *et al.* [2018]) with 2.7 billion tokens and the *base* version of the model (220M parameters) was used in our experiments. In a previous work Sakiyama *et al.* [2023b], we experimented with other state-of-the-art multilingual generative Transformer models (mT5 Xue *et al.* [2020], BLOOM Scao *et al.* [2022] and OPT Zhang *et al.* [2022]), but the Portuguese model (PTT5) performed better. Previous works Carmo *et al.* [2020]; Souza *et al.* [2020]; Rosa *et al.* [2021a] observed that models pretrained for the task language tend to outperform multilingual models on the same tasks, and the same trend was observed in our experiments.

3.2.2 Training details

We fixed the input (enumerated paragraphs) and output (keyphrases) sizes to 512 and 256 sentence-piece tokens, respectively. We padded shorter sequences of tokens and truncated longer sequences (to the maximum length). We fine-tuned the method PTT5 using a fixed learning rate of 1×10^{-3} , batch size equal to 256 and 20 maximum training epochs.

For the sequence-to-sequence training, the cross-entropy loss function was adopted. The BLEU score Post [2018] metric was considered to evaluate the text generation quality. The metric was chosen because it serves the purpose of estimating the quality of the generated text and is highly correlated with human evaluations Papineni *et al.* [2002]. Note that ROUGE Lin [2004] could also be used for similar reasons. Furthermore, we chose not to evaluate metrics based on pre-trained models, such as *BERTScore* Zhang *et al.* [2019]. Due to time constraints, we considered it more appropriate (and practical) to adopt an evaluation metric that does not rely on a pre-trained language model.

We used early-stopping during training, monitoring the BLEU metric in the validation set. The training process is stopped after two epochs without improving the BLEU score. For evaluation, we repeated the training process with five different seeds (1000, 2000, 3000, 4000 and 5000) and obtained a 37.254 ± 0.783 BLEU score. The best model achieved 38.607 BLEU on the test set. The fine-tuning was done using the *HuggingFace*³ library, and a Tesla P100 GPU with 16 GB VRAM.

Figure 7 shows the train and validation losses for the best execution of the PTT5 model in addition to the validation BLEU scores. The model was trained for 17 epochs, totaling 5219 iterations.

3.3 Decoding Methods Evaluation

For the evaluation of the generated text and to compare the decoding methods, we have concatenated the generated keyphrases to their original document and used a real use case

³huggingface.co/

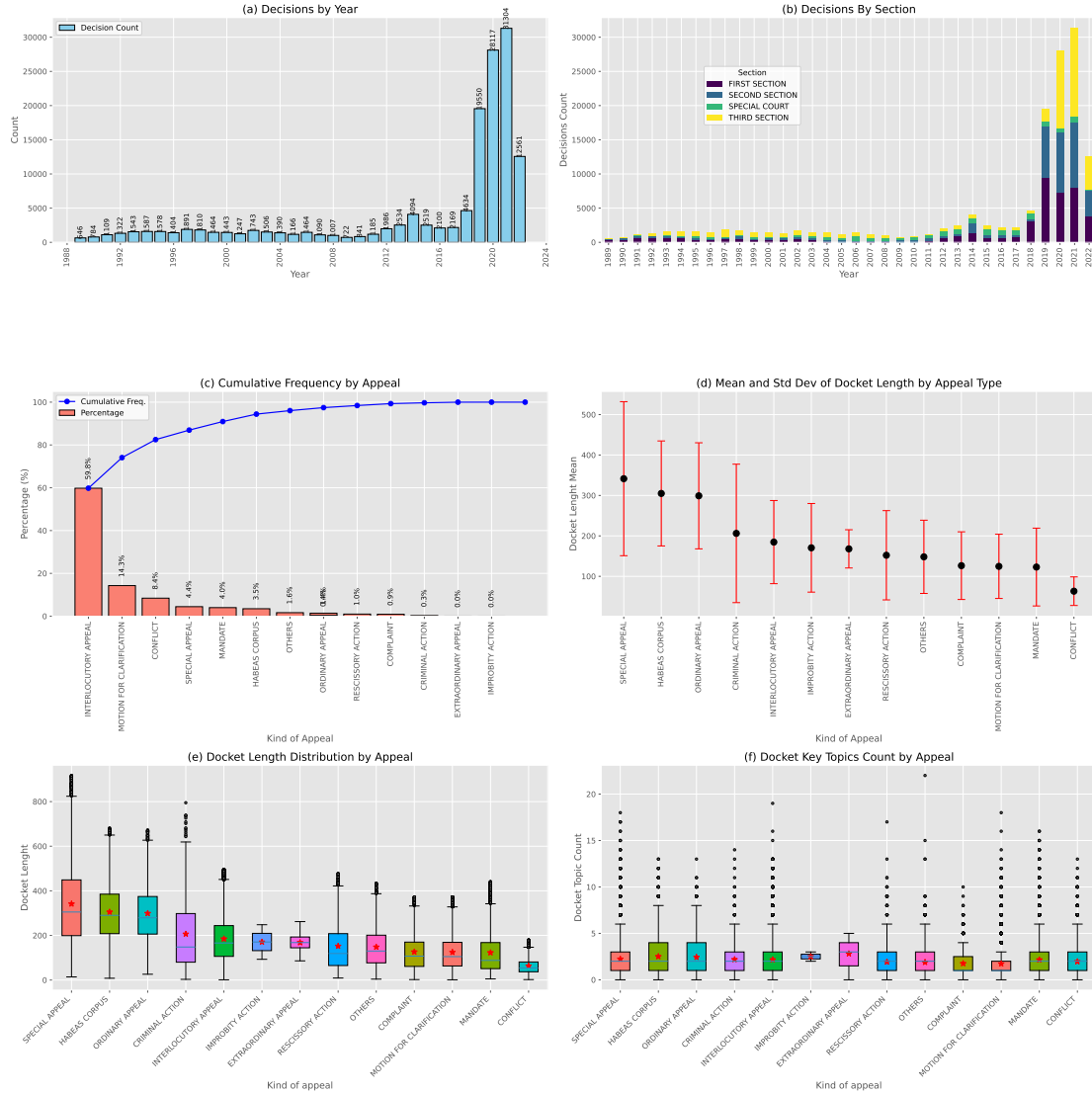


Figure 6. (a) shows the annual number of decisions. (b) illustrates the number of decisions made by sections of the court over the years. (c) shows the percentage distribution of appeal types, with a cumulative frequency line highlighting the contribution of each type to the total. (d) presents the mean and standard deviation of the summary length by appeal type. (e) illustrates the dispersion and identifies outliers in summary lengths. (f) details the distribution of the length of the main topics in the summary.

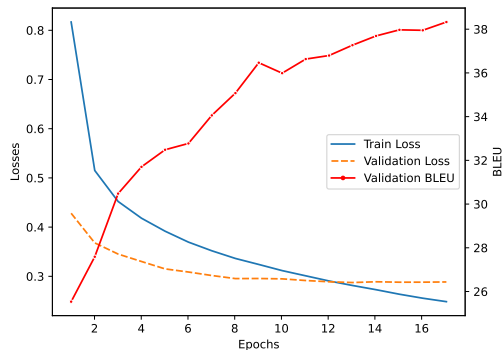


Figure 7. Train and validation losses obtained for the PTT5 model are shown in the left y-axis. The plot also shows the validation BLEU scores in the right y-axis.

of retrieval to extract IR metrics. We opted to use an IR task to evaluate the generated keyphrases (created using different decoding methods) to evaluate them in their intended use: improving retrieval tasks. The details of the evaluations will be presented as follows.

3.3.1 Decoding Methods

Decoding techniques are used to guide neural text generation, to generate meaningful and coherent text. The methods are used to generate human-readable text from the internal representations of language models. In this work, we evaluate three decoding methods: greedy, top-k Fan *et al.* [2018] and top-p Holtzman *et al.* [2019]. Top-k and top-p are sampling decoding methods, that, during text generation, sample tokens from finite sets. A brief description of the methods will be presented next.

- **Greedy**: greedy decoding always selects the most probable token (highest softmax score) during generation.
- **Top-K**: consists of filtering the most probable K tokens at a given instant, and redistributing their probabilities among them before sampling.
- **Top-p**: limits the set of selectable tokens to a set of more probable tokens whose summed probabilities are lower than the established threshold p . Note that the number

of tokens that can be chosen is not constant, since the probability distributions vary at each instant.

3.3.2 Task Formulation

We have used the themes (categorical information), present in the dockets' metadata, to simulate a retrieval task in which a specialist seeks to retrieve documents similar to a query document using a search engine. The themes are unique identifiers, mapped to common questions of law. Examples of themes are shown in Figure 8. Thus, by using the binary relevance definition: given a query document Q , the relevant documents R to Q must have the same theme as Q . Note that, in the real scenario, the documents consist in dockets containing both keyphrases and enumerated paragraphs.

The presented formulation is similar to that used by Ostendorff *et al.* [2021], in which the authors created relevance pairs (query document and relevant document) using decisions from the United States Supreme Court drawn from the same casebooks or categories. Note that there is no guarantee that the query – relevant document pairs are truly relevant to each other. However, given the effort and cost required to annotate such pairs Sanderson [2010], the annotations derived from the themes provide a reasonable proxy for relevance.

Theme 8: Monetary adjustment shall accrue from the date on which payment of each installment was due.

Theme 470: Statutory default interest arising from labor-related amounts recognized in a court decision is not subject to income tax.

Theme 1205: The mere immediate and full restitution of the stolen property is not, in itself, sufficient to warrant the application of the principle of insignificance.

Figure 8. Examples of themes listed by STJ.

From the collected decisions, only 801 have themes. These documents were removed from the training set and used to prepare query-relevant document pairs for IR evaluation. The query set consists of dockets whose themes occur at least twice. From those, we prepared 482 query - relevant document pairs (pairs of same theme documents).

To prepare the final retrieval corpus, we combined the test set presented in Section 3.1 with the dockets with themes and obtained a total of 23,194 documents. We increased the retrieval corpus to make the retrieval task more challenging. In the worst case, the documents without a theme may introduce false negatives (documents with the same theme of the query, but considered non-relevant), hindering the IR metrics.

3.3.3 Experimental setup

Two different experiments were performed during the IR evaluation and they are described in the following.

1. *Studying Sampling-based Decoding Methods:* this experiment aims to investigate the generation of multiple keyphrases from a single docket using sampling decoding. By concatenating multiple keyphrases to a single docket, we expect to see improvements in the IR metrics since we are adding more text variations.

We generated up to 10 keyphrases for each example in the search corpus, using top-K and top-p sampling, and concatenated them to the original input (enumerated paragraphs) to generate the IR metrics. We repeated the text generation five times with seeds of different random numbers (1,000, 2,000, 3,000, 4,000, and 5,000) and aggregated the results for comparisons. The effects of the K of the top-K, and the p of the top-p sampling methods were also evaluated in this experiment, varying the values of both K and p .

We choose K and p from the following sets of values: $K \in \{15, 50, 100\}$ and $p \in \{0.1, 0.5, 0.9\}$. Note that for this experiment, we are not interested in determining the best number of repetitions, nor the best value for K or p . The goal is to investigate the effect of the parameters on the proposed IR task, but the results may indicate the best parameter ranges.

2. *Decoding Methods Comparison:* to compare the decoding methods, we extracted IR metrics for dockets using keyphrases generated using top-K and top-p sampling. We used the generated ones in place of the originals in this experiment. For reference, we also evaluated IR metrics considering documents with and without the original keyphrases (for both query and corpus documents) and using simple greedy decoding.

We choose to use only one keyphrase generated by each method based on the results obtained from the previous experiment and to evaluate the decoding methods in similar scenarios. For this experiment, we used the following parameters for the sampling-based decoding methods: $K = 15$ and $p = 0.9$ (based on the performances obtained in the previous experiment).

The experiments with sampling decoding methods were inspired by the work *doc2query* Nogueira *et al.* [2019]. In this work, for each example in a corpus, the authors generated several queries related to the example's content using a sequence-to-sequence Transformer model. The authors used top-K sampling to create several queries per example. Then the queries were concatenated to the input documents for improving IR metrics. Considering both experiments with sampling methods, we used contrastive-search with $\alpha = 0.6$, based on the original paper Su *et al.* [2022]. We choose the K and p values based on previous works with top-K and top-p sampling Nogueira *et al.* [2019]; Holtzman *et al.* [2019].

3.3.4 Information Retrieval Methods and Metrics

To evaluate the proposed IR task, we choose two traditional methods: TF-IDF and BM25 Robertson and Walker [1999]. The methods were chosen due to their popularity in search engines (such as Lucene⁴) and competitive performance Rosa *et al.* [2021b]; Pradeep *et al.* [2020]. Previous works Lima *et al.* [2021]; Mandal *et al.* [2021]; Pedrosa *et al.* [2019] also discussed that sparse representation methods (such as the chosen ones) tend to perform better in similar tasks in the legal domain.

As an additional preprocessing for the IR methods, the documents were tokenized and Portuguese stop-words and

⁴<https://lucene.apache.org/>

punctuation were removed. For TF-IDF, we utilized a vocabulary size of 10,000 tokens (that appeared at least three times), and n-grams from 1 to 3. To sort documents during retrieval using TF-IDF, we used the cosine similarity between queries and documents. Considering BM25, the documents were sorted by the probability ranking principle, estimating the relevance of a document to the presented query. The additional preprocessing was done using spaCy⁵ and sklearn⁶. For BM25, we used the implementation and default parameters from *rank-bm25*⁷.

At last, we evaluated the performance in the proposed IR task using two traditional IR metrics: Mean Reciprocal Rank (MRR) and Recall. The metrics were chosen by their use in similar works in the legal domain Russell-Rose *et al.* [2018]; Souza *et al.* [2021]. We used a threshold of 10 documents (top-10 ranked documents) to compute the metrics. According to Russell *et al.* Russell-Rose *et al.* [2018], law professionals tend to analyze, for the most part, up to 50 documents in their searches. Therefore, we are evaluating an even more challenging scenario than the described by the authors.

3.4 Qualitative Analysis

As a final analysis, for all decoding methods evaluated (greedy, top-K, and top-p), we sampled examples generated using all methods and performed a qualitative analysis on them. For this analysis, we compared the generated keyphrases to the references and discussed the similarities between them and the effect of the sampling methods.

4 Results and discussions

The next sections discuss the results obtained for each experiment described in Section 3.3.3. In all experiments, we aim not to compare the retrieval methods (TF-IDF and BM25), but to use them to evaluate the quality of the generated keyphrases using different decoding methods.

4.1 Studying Sampling-based Decoding Methods

Tables 1 and 2 present the IR metrics obtained, varying as the number of repetitions as K e p parameters. Also, Figures 9 and 10 illustrate the effect of changing the aforementioned parameters. The metrics consist of the mean of five different executions (using five different seeds).

When carrying out this experiment, the expectation was to observe a logarithmic growth as more different keyphrases were concatenated to the dockets (similar to *doc2query* Nogueira *et al.* [2019]), since we are using more variations of keyphrases. However, this result was not observed in any of the evaluated metrics (see Tables 1 and 2). Contrary to expectations, in the worst cases, there was a decay in the metrics as new variations were added to the input texts for both top-K and top-p decoding methods. The decay is more noticeable for TF-IDF method, with reductions between 2%

Table 1. Top-k experiments evaluation metrics. N represents the number of samples included at the beginning of each docket.

N	k=15		k=50		k=100	
	MRR@10	R@10	MRR@10	R@10	MRR@10	R@10
1	0.826	0.804	0.824	0.811	0.824	0.809
2	0.820	0.796	0.818	0.798	0.817	0.799
3	0.815	0.793	0.812	0.789	0.812	0.790
4	0.811	0.784	0.812	0.785	0.812	0.783
5	0.809	0.779	0.810	0.780	0.810	0.780
6	0.808	0.776	0.810	0.782	0.810	0.783
7	0.810	0.781	0.806	0.783	0.805	0.782
8	0.807	0.781	0.809	0.784	0.809	0.784
9	0.806	0.777	0.810	0.784	0.808	0.783
10	0.804	0.778	0.810	0.784	0.809	0.787

(a) TF-IDF experiments.

N	k=15		k=50		k=100	
	MRR@10	R@10	MRR@10	R@10	MRR@10	R@10
1	0.861	0.882	0.857	0.880	0.856	0.881
2	0.857	0.879	0.857	0.876	0.856	0.875
3	0.858	0.875	0.856	0.874	0.856	0.873
4	0.859	0.878	0.856	0.875	0.856	0.874
5	0.858	0.876	0.857	0.872	0.854	0.872
6	0.858	0.875	0.854	0.873	0.852	0.873
7	0.859	0.874	0.855	0.872	0.855	0.872
8	0.860	0.874	0.855	0.870	0.853	0.870
9	0.860	0.873	0.855	0.871	0.854	0.869
10	0.861	0.871	0.857	0.871	0.856	0.871

(b) BM25 experiments.

Table 2. Top-p experiments evaluation metrics. N represents the number of samples included at the beginning of each docket.

N	p=0.1		p=0.5		p=0.9	
	MRR@10	R@10	MRR@10	R@10	MRR@10	R@10
1	0.821	0.815	0.823	0.813	0.826	0.807
2	0.787	0.768	0.804	0.790	0.820	0.797
3	0.761	0.724	0.788	0.774	0.810	0.790
4	0.743	0.703	0.781	0.763	0.809	0.785
5	0.727	0.684	0.776	0.748	0.805	0.779
6	0.717	0.672	0.772	0.740	0.806	0.781
7	0.713	0.661	0.768	0.734	0.803	0.780
8	0.707	0.649	0.764	0.733	0.804	0.777
9	0.701	0.640	0.760	0.724	0.805	0.777
10	0.697	0.637	0.758	0.719	0.801	0.774

(a) TF-IDF experiments.

N	p=0.1		p=0.5		p=0.9	
	MRR@10	R@10	MRR@10	R@10	MRR@10	R@10
1	0.854	0.877	0.855	0.880	0.859	0.883
2	0.842	0.866	0.852	0.880	0.858	0.880
3	0.842	0.862	0.853	0.875	0.860	0.881
4	0.832	0.853	0.852	0.872	0.858	0.878
5	0.828	0.845	0.852	0.866	0.855	0.876
6	0.827	0.836	0.851	0.866	0.857	0.876
7	0.823	0.826	0.851	0.865	0.858	0.874
8	0.821	0.823	0.853	0.864	0.859	0.876
9	0.819	0.822	0.853	0.862	0.859	0.876
10	0.816	0.818	0.853	0.861	0.861	0.873

(b) BM25 experiments.

(top-K) and 12% (top-p) in all observed metrics. However, as shown by the shaded regions in Figure 10, top-p have the lowest variance in the metrics. The mentioned behaviours were observed for all evaluated K and p values.

The aforementioned patterns can also be seen in Figures 9 and 10. In addition to the decline in metrics, the Figures also show the performance when using only the original keyphrases as dashed lines. As we can see, the values tend to move away from the performance obtained using the original

⁵<https://spacy.io/>

⁶<https://scikit-learn.org>

⁷<https://pypi.org/project/rank-bm25/>

keyphrases as generated keyphrases are added to the dockets. The results obtained from the original keyphrases will be further discussed in the following Section.

The worst performances were observed in increasing repetitions for $p = 0.1$ (see Table 10 and Figures 10a and 10b). The most probable explanation is that the low p value is too restrictive, reducing the set of selectable tokens. Hence, the top- p tends to generate similar keyphrases with low text variation (more similar or equal keyphrases). This way, the repetitive text hindered the performance of both IR methods evaluated. Also, when using $p = 0.1$, we observed almost no variation in the metrics shown in Figures 10a and 10b. This reinforces the low variability in the generated text over, considering the 5 repetitions of the experiment.

By increasing the K and p values, we increase the generated text variability, since the tokens to be predicted are chosen from a larger set. A positive effect on the metrics was also expected due to the possibility of adding more discriminative terms in the generated keyphrase, which is beneficial for the evaluated sparse methods. However, we observed a certain deterioration of the performance obtained, and at the best case, similar metrics by varying the K and p values. We suspect that even with the increase in variability, the generated keyphrases remained similar for each other, resulting in the addition of repetitive texts to the dockets.

The conclusion from these results is that there is no evidence that if we use more keyphrases (by using sampling decoding) would be beneficial for the evaluated task. Also, there is no benefit in using K values above 15, and p values lower than 0.9. We will discuss the results of the sampling methods further in Section 4.3.2.

4.2 Decoding Methods Comparison

In Table 3 is presented the results comparing decoding methods. For both the methods: TFIDF and BM25, a single keyphrase using greedy was generated for both top- K and top- p decoding. We adopted $K = 15$ and $p = 0.9$ for the top- K and top- p decoding, respectively, based on the results from the previous experiment. Table 3 also presents the results obtained performing the proposed retrieval task with and without the original (reference) keyphrases for comparisons.

We observe that the keyphrases can help in the retrieval tasks by comparing the metrics obtained by using documents with and without the keyphrases. For both metrics, we observed statistically significant differences. Since both sparse methods (TF-IDF and BM25) benefit from the existence of discriminative terms in the documents, these results were already expected. Note that the metrics obtained using the original keyphrases act as an upper bound to our experiments.

Considering the TF-IDF retrieval, we observed an increment in all metrics by using the generated texts (compared to not using any). The differences in all metrics are statistically significant (see Table 3a). For the BM25 method, we observe similar results (see Table 3b). However, no significant differences were observed when considering the R@10 metric. Note that by using generated keyphrases, there is the possibility of introducing false positives (false similar) and false negatives (false non-similar) in the search corpus, originated by noisy keyphrases. The IR metrics obtained by the BM25

Table 3. IR metrics obtained for each decoding method evaluated. Superscript characters denote statistically significant pairwises, according to a paired T-test (p -value < 0.05).

(a) TF-IDF experiments.

	MRR@10	R@10
a) Without	0.806	0.790
b) Generated greedy	0.822 ^a	0.815 ^a
c) Generated top-K	0.828 ^a	0.810 ^a
d) Generated top-p	0.825 ^a	0.811 ^a
Original	0.825 ^a	0.832 ^a

(b) BM25 experiments.

	MRR@10	R@10
a) Without	0.819	0.878
b) Generated greedy	0.854 ^a	0.877
c) Generated top-K	0.863 ^a	0.890
d) Generated top-p	0.859 ^a	0.880
Original	0.879 ^a	0.916 ^a

method suggest that the method was sensitive to these noisy examples.

By comparing the decoding methods, we note small increments for the sampling methods related to greedy decoding. However, considering a paired T-Test using a threshold of 5%, there is no significant difference between the decoding methods. Thus, there is not evidence enough to reject the null hypothesis (metrics have the same mean) by observing the comparisons between the metrics of all three decoding approaches. Therefore, there is no evidence to justify the choice to sample decoding methods over a simpler greedy decoding approach, considering the proposed task.

4.3 Qualitative Analysis

The following Sections will present a qualitative analysis in the generated keyphrases using the investigated decoding methods.

4.3.1 Greedy Decoding

Examples of keyphrases generated by PTT5, using greedy decoding, are presented in Figure 11. We can note that with BLEU scores close to 40%, although generated by a model trained in a modest training set (less than 100K examples), the generated keyphrases do not present spelling and lexical errors. They captured the writing style of the reference keyphrases and are very similar to the keyphrases written by humans.

A comparison between the number of tokens of the original and the generated keyphrases using greedy decoding is shown in Figure 12. It is possible to observe that, although the distributions presented by the two histograms are similar, the generated keyphrases have a higher concentration of examples below 60 tokens. The average in space-separated tokens of the generated keyphrases is lower than the average of the tokens presented by the references (42.34 compared to 48.28).

Therefore, we identified that the keyphrases generated with greedy decoding have a tendency for phrases of smaller length (in tokens) than the originals. We also observed the same pattern for the keyphrases generated using sampling decoding.

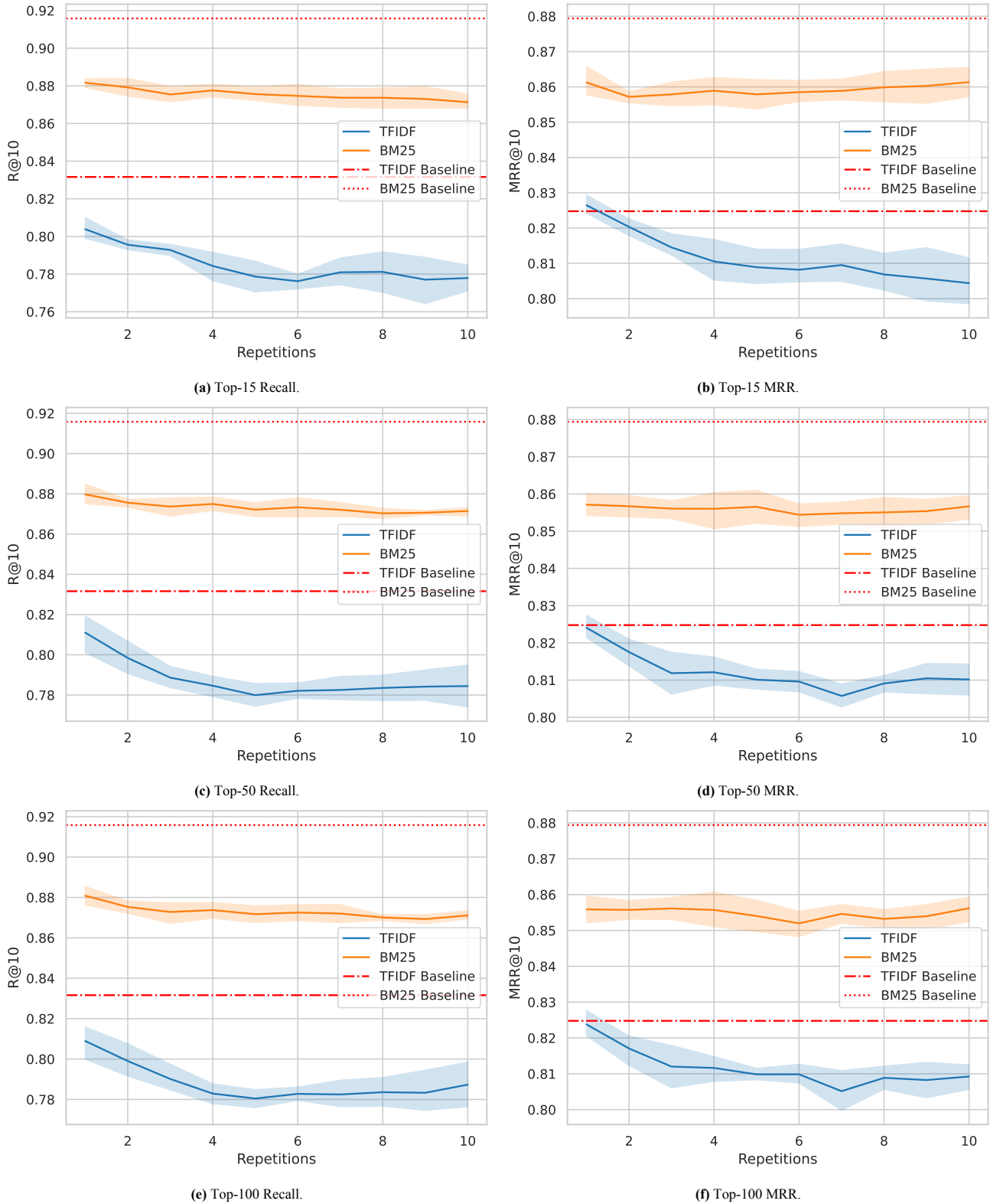


Figure 9. Metrics obtained generating keyphrases using top-K sampling. The horizontal axis shows the number of repetitions concatenated to the docket. The dashed lines indicate the values obtained, using the original keyphrases in the IR task. The shaded regions indicate the intervals with 95% confidence, considering 5 repetitions of the generation procedure.

4.3.2 Sampling Decoding

Figure 13 shows examples of keyphrases generated using top-K and top-p decoding. We used $K = 15$ and $p = 0.9$ based on the results from the previous analysis. From the examples, it is possible to note that the main effect of using sampling is

the generation of paraphrases of the original keyphrases. We also observe examples of reordering of the phrases present in the keyphrases. Hence, the generated keyphrases tend to be similar to each other.

Next, Figure 14 illustrates two interesting recurrent behaviours observed when using both sampling decoding tech-

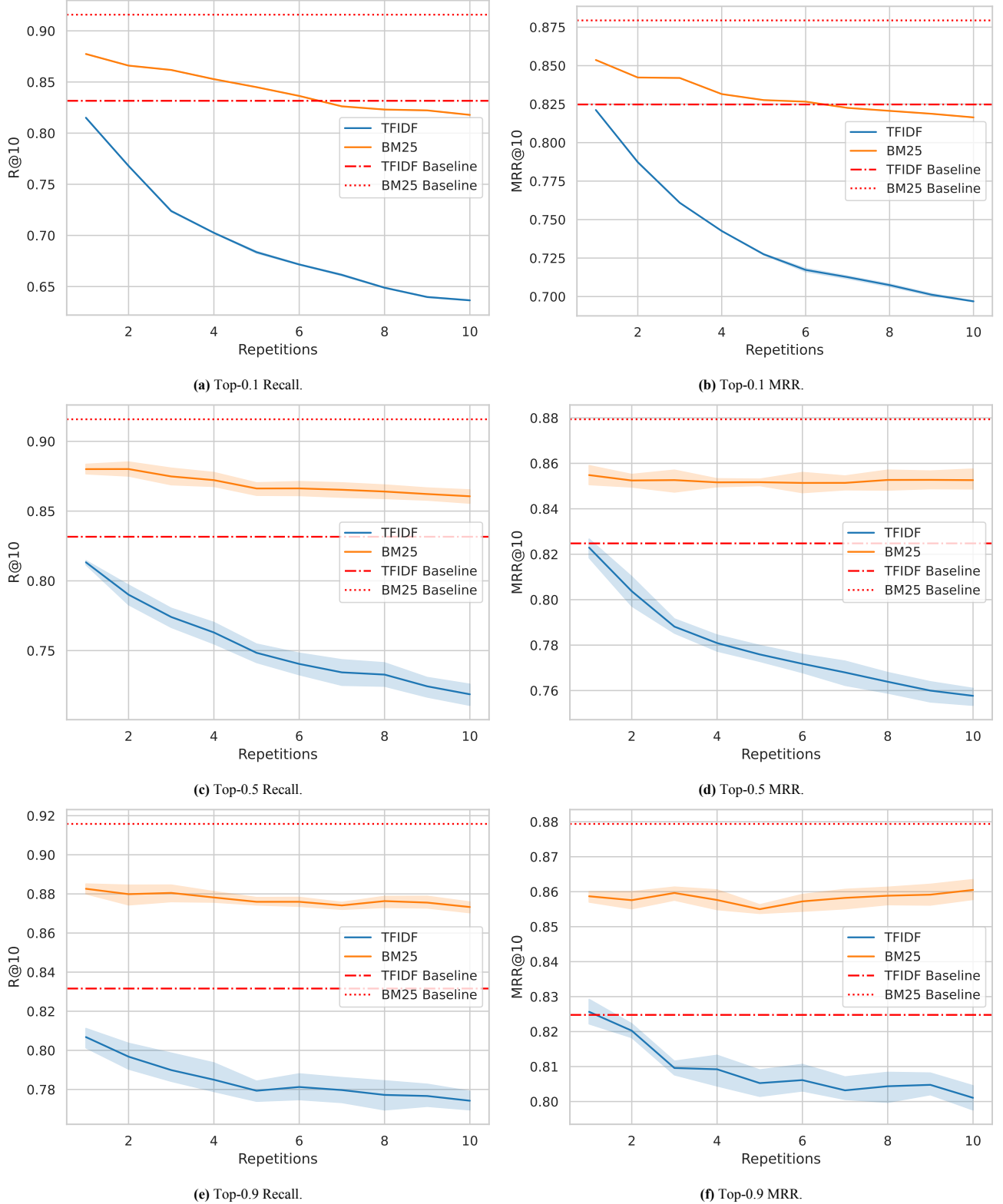


Figure 10. Similar to Figure 9, using keyphrases generated using Top-p sampling instead of Top-k.

niques (top-K and top-p). In addition to exact terms, we highlight in yellow terms that were changed by the Transformer. We observed that the model successfully switched ‘SERVIDORES PÚBLICOS’ (public servants) by an equivalent synonym ‘AGENTE PÚBLICO’ (public agent). In addition, the PTT5 model was also able for expanding the acronyms such as ‘CF/1998’ to ‘CONSTITUIÇÃO FEDERAL DE 1998’ and ‘EC’ to ‘EMENDA CONSTITUCIONAL’.

The described behaviours are justified by the working of language models based on Transformers since, during text generation, they tend to generate tokens that appear in similar contexts. These behaviours are also present in the greedy decoding generated texts, but were more visible when using sampling decoding methods, given the larger next token possibilities.

By using sampling-based methods, we observed an increase

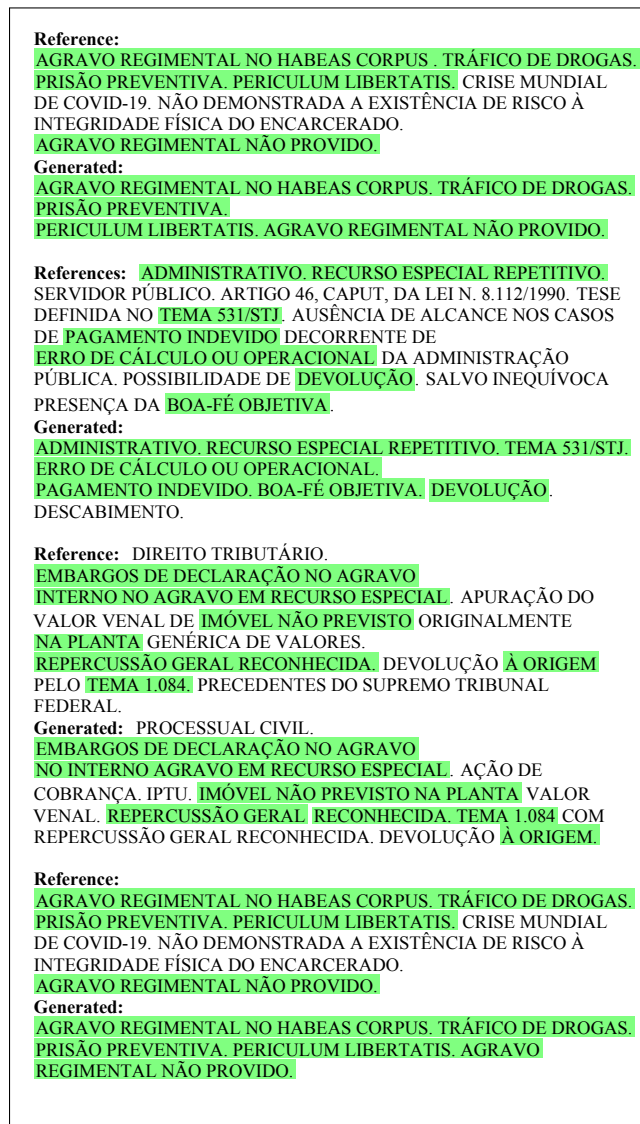


Figure 11. Examples generated using greedy decoding. Exact matches are highlighted in green.

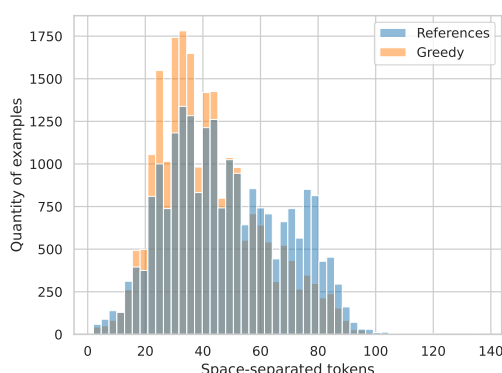


Figure 12. Histogram comparing the number of space-separated tokens between the reference keyphrases and the ones generated using greedy decoding.

in text variability. However, the possibility of the model generating text not related to the input (hallucinations) also increases, which may have harmed the IR methods. In Figure 15 we show examples of this behaviour by highlighting the potential hallucinated terms (terms that aren't present in the original keyphrase).

Original: PROCESSUAL CIVIL E TRIBUTÁRIO. AGRADO INTERNO NO RECURSO ESPECIAL. EXECUÇÃO FISCAL. NOMEAÇÃO DE BENS À PENHORA. RECUSA INJUSTIFICADA DA FAZENDA. REVISÃO DAS CONCLUSÕES ADOTADAS NA ORIGEM QUE DEMANDA REEXAME DE PROVA. SÚMULA 7/STJ. AGRADO INTERNO DA FAZENDA DO ESTADO DE SÃO PAULO DESPROVIDO.

1) Top-K: PROCESSUAL CIVIL E TRIBUTÁRIO. AGRADO INTERNO NO RECURSO ESPECIAL. EXECUÇÃO FISCAL. PENHORA. ORDEM LEGAL. NOMEAÇÃO. RECUSA JUSTIFICADA. REVISÃO. IMPOSSIBILIDADE. SÚMULA 7/STJ. AGRADO INTERNO DA FAZENDA DO ESTADO DE SÃO PAULO A QUE SE NEGA PROVIMENTO.

2) Top-K: TRIBUTÁRIO. AGRADO INTERNO NO RECURSO ESPECIAL. EXECUÇÃO FISCAL. PENHORA. NOMEAÇÃO. RECUSA JUSTIFICADA. POSSIBILIDADE. REVISÃO. INCIDÊNCIA DA SÚMULA 7/STJ. AGRADO INTERNO DA FAZENDA DO ESTADO DE SÃO PAULO DESPROVIDO.

3) Top-p: TRIBUTÁRIO. AGRADO INTERNO NO AGRADO EM RECURSO ESPECIAL. EXECUÇÃO FISCAL. PENHORA DE BEM NOMEADO NA PRIMEIRA OPORTUNIDADE. RECUSA PELO FISCO EXEQUENTE NÃO JUSTIFICADA. POSSIBILIDADE. REVISÃO. SÚMULA 7/STJ. AGRADO INTERNO DA FAZENDA DO ESTADO DE SÃO PAULO DESPROVIDO.

4) Top-p: TRIBUTÁRIO E PROCESSUAL CIVIL. AGRADO INTERNO NO AGRADO EM RECURSO ESPECIAL. EXECUÇÃO FISCAL. PENHORA. NOMEAÇÃO DE PRECATÓRIOS. ORDEM LEGAL. NEGATIVA DE PRESTAÇÃO JURISDICIONAL. NÃO OCORRÊNCIA. REEXAME DE FATOS E PROVAS. INVIABILIDADE. AGRADO INTERNO DA FAZENDA DO ESTADO DE SÃO PAULO DESPROVIDO.

Figure 13. Examples of keyphrases generated by top-K ($K = 15$) and top-p ($p = 0.9$) decoding.

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.

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

Figure 14. Additional examples of generated keyphrases. Exact matches are highlighted in green and changed terms are highlighted in yellow.

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.

Generated: TRIBUTÁRIO E PROCESSUAL CIVIL. RECURSO ESPECIAL, POR FAC-SIMILE. INTERPOSIÇÃO, NA ORIGEM, DE AGRADO 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. AGRADO INTERNO IMPROVIDO.

Figure 15. Example of keyphrase that introduces terms not present in the original keyphrase. The terms are highlighted in red.

Without domain-specific knowledge, it is not possible to ensure that such 'new' terms are associated with the input text. We can only assume that they occur in similar contexts. Therefore, there is a risk of adding factually incorrect texts

to the generated text, due to the generative process. In addition, when concatenating multiple variations of keyphrases similar to each other, we added many repeated terms to the documents, which may influence negatively the sparse IR methods evaluated.

In addition to the justifications presented, the amount of training data may also have affected the sampling methods. Although the results for greedy generation were better, the lack of variability in the training examples (due to their small size), may have harmed the decoding using top-K and top-p sampling.

5 Conclusion and Future works

In this paper, we successfully trained a sequence-to-sequence Transformer to generate keyphrases and investigated three different text decoding methods. The results showed us that the keyphrases can bring significative increments to Information Retrieval tasks as used in conjunction with the dockets. This result was observed for all the keyphrases evaluated: the references and the generated ones (using greedy, top-K, and top-p decoding). Although we have evaluated different parameters and concatenated multiple variations of keyphrases generated using sampling decoding (top-K and top-p), the simpler greedy decoding performed similarly to these methods. We presented and discussed possible justifications for such behavior, and the results suggest that greedy decoding is enough for keyphrase generation considering legal dockets.

In future works, research will focus on pre-training transformer models on extensive legal-specific corpora to capture better the intricacies and nuances of legal language, which are often distinct from general text. By applying fine-tuning in the parameters of the models on a comprehensive collection of legal documents, we aim to improve the contextual relevance and accuracy of generated keyphrases. Additionally, efforts will be directed toward expanding the dataset by incorporating dockets from a broader range of courts and legal institutions across Brazil. This will enhance the model's ability to generalize across diverse legal contexts and procedural variations, ensuring robust performance across various types of legal documents and jurisdictions.

Declarations

Acknowledgements

This study was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001. We thank CEMEAI for granting access to the Euler cluster for the experiments. Also, this work is partially funded by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), grant 2022/01640-2. We thank also INCT (CAPES Concessão 88887.136349/2017-00, CNPQ 465755/2014-3 and FAPESP 2014/50851-0) for the support.

Authors' Contributions

All authors contributed equally to the research and writing of this article.

Competing interests

The authors declare that they have no competing interests.

Funding

This research received no external funding.

Availability of data and materials

The processed dataset used in this study, derived from Brazilian legal documents, is publicly available on Hugging Face at https://huggingface.co/datasets/bfunicheli/stj_docket_generation

References

- Althammer, S., Askari, A., Verberne, S., and Hanbury, A. (2021). Dossier@ coliee 2021: Leveraging dense retrieval and summarization-based re-ranking for case law retrieval. *arXiv preprint arXiv:2108.03937*. DOI: 10.48550/arXiv.2108.03937.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901. DOI: 10.48550/arxiv.2005.14165.
- Carmo, D., Piau, M., Campiotti, I., Nogueira, R., and Lotufo, R. (2020). Ptt5: Pretraining and validating the t5 model on brazilian portuguese data. *arXiv preprint arXiv:2008.09144*. DOI: 10.48550/arxiv.2008.09144.
- Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703. DOI: 10.1137/070710111.
- Conselho Nacional de Justiça - CNJ (23). Conselho nacional de justiça — justiça em números. Available at: <https://www.cnj.jus.br/pesquisas-judiciarias/justica-em-numeros/>. Accessed on 08/05/2023.
- Fan, A., Lewis, M., and Dauphin, Y. N. (2018). Hierarchical neural story generation. *CoRR*, abs/1805.04833. DOI: 10.18653/v1/p18-1082.
- Feijo, D. and Moreira, V. (2019). Summarizing legal rulings: Comparative experiments. In *proceedings of the international conference on recent advances in natural language processing (RANLP 2019)*, pages 313–322. DOI: 10.26615/978-954-452-056-4_36.
- Floridi, L. and Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694. DOI: 10.1007/s11023-020-09548-1.
- Guimarães, J. A. C. and Santos, J. C. G. (2016). A ementa jurisprudencial como resumo informativo em um domínio especializado: aspectos estruturais. *Brazilian Journal of Information Science: research trends*, 10(3). Available at: <https://dialnet.unirioja.es/servlet/articulo?codigo=5754542>.
- Holtzman, A., Buys, J., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *CoRR*, abs/1904.09751. DOI: 10.48550/arxiv.1904.09751.

- Huang, Y., Shen, X., Li, C., Ge, J., and Luo, B. (2021). Dependency learning for legal judgment prediction with a unified text-to-text transformer. *arXiv preprint arXiv:2112.06370*. DOI: 10.48550/arxiv.2112.06370.
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Madotto, A., and Fung, P. (2022). Survey of hallucination in natural language generation. *ACM Computing Surveys*. DOI: 10.1145/3571730.
- Li, W. (2002). Zipf's law everywhere. *Glottometrics*, 5:14–21. Available at: https://www.researchgate.net/profile/Wentian-Li/publication/253290454_Zipf's_Law_Everywhere/links/5cf59ef9299bf1fb185617ff/Zipfs-Law-Everywhere.pdf.
- Lima, J. P., Costa, J. A., and Araújo, D. C. (2021). Comparison of feature extraction methods for brazilian legal documents clustering. In *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–5. IEEE. DOI: 10.1109/la-cci48322.2021.9769839.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81. Available at: <https://aclanthology.org/W04-1013.pdf>.
- Mandal, A., Ghosh, K., Ghosh, S., and Mandal, S. (2021). Unsupervised approaches for measuring textual similarity between legal court case reports. *Artificial Intelligence and Law*, 29(3):417–451. DOI: 10.1007/s10506-020-09280-2.
- Nogueira, R., Yang, W., Lin, J., and Cho, K. (2019). Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*. DOI: 10.48550/arxiv.1904.08375.
- Ostendorff, M., Ash, E., Ruas, T., Gipp, B., Moreno-Schneider, J., and Rehm, G. (2021). Evaluating document representations for content-based legal literature recommendations. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 109–118. DOI: 10.1145/3462757.3466073.
- Pacheco, C. C. (2008). Os estudos sobre judiciário e política no brasil pós 1988: uma revisão da literatura. *Pensar-Revista de Ciências Jurídicas*, 13(1):75–86. DOI: 10.5020/23172150.2012.75-86.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318. Available at: <https://aclanthology.org/P02-1040.Pdf>.
- Pedroso, D. d. S. C., Ladeira, M., and de Paulo Faleiros, T. (2019). Does semantic search performs better than lexical search in the task of assisting legal opinion writing? In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 680–685. IEEE. DOI: 10.1109/icmla.2019.00123.
- Peric, L., Mijic, S., Stambach, D., and Ash, E. (2020). Legal language modeling with transformers. In *Proceedings of the Fourth Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2020) held online in conjunction with the 33rd International Conference on Legal Knowledge and Information Systems (JURIX 2020) December 9, 2020*, volume 2764. CEUR-WS. Available at: <https://ceur-ws.org/Vol-2764/paper2.pdf>.
- Perlingeiro, R. and Schmidt, L. S. (2023). An overview of environmental justice in brazil. *British Journal of American Legal Studies*, 12(1):27–50. DOI: 10.2478/bjals-2023-0003.
- Piantadosi, S. T. (2014). Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130. DOI: 10.3758/s13423-014-0585-6.
- Post, M. (2018). A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*. DOI: 10.18653/v1/w18-6319.
- Pradeep, R., Ma, X., Zhang, X., Cui, H., Xu, R., Nogueira, R., and Lin, J. (2020). H2oloo at trec 2020: When all you got is a hammer... deep learning, health misinformation, and precision medicine. *Corpus*, 5(d3):d2. DOI: 10.6028/nist.sp.1266.misinfo-h2oloo.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9. Available at: <https://storage.prod.researchhub.com/uploads/papers/2020/06/01/language-models.pdf>.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67. Available at: <https://www.jmlr.org/papers/v21/20-074.html>.
- Robertson, S. E. and Walker, S. (1999). Okapi/keenbow at trec-8. In *TREC*, volume 8, pages 151–162. Citeseer. DOI: 10.6028/nist.sp.500-246.adhoc-microsoft.
- Rosa, G. M., Bonifacio, L. H., de Souza, L. R., Lotufo, R., and Nogueira, R. (2021a). A cost-benefit analysis of cross-lingual transfer methods. *arXiv preprint arXiv:2105.06813*. DOI: 10.48550/arxiv.2105.06813.
- Rosa, G. M., Rodrigues, R. C., Lotufo, R., and Nogueira, R. (2021b). Yes, bm25 is a strong baseline for legal case retrieval. *arXiv preprint arXiv:2105.05686*. DOI: 10.48550/arXiv.2105.05686.
- Russell-Rose, T., Chamberlain, J., and Azzopardi, L. (2018). Information retrieval in the workplace: A comparison of professional search practices. *Information Processing & Management*, 54(6):1042–1057. DOI: 10.1016/j.ipm.2018.07.003.
- Sakiyama, K., Montanari, R., Malaquias Junior, R., Nogueira, R., and Romero, R. A. (2023a). Exploring text decoding methods for portuguese legal text generation. In *Brazilian Conference on Intelligent Systems*, pages 63–77. Springer. DOI: 10.1007/978-3-031-45368-7_5.
- Sakiyama, K., Nogueira, R., and Romero, R. A. F. (2023b). Automated keyphrase generation for brazilian legal information retrieval. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. DOI: 10.1109/IJCNN54540.2023.10191598.
- Sanderson, M. (2010). *Test collection based evaluation of information retrieval systems*. Now Publishers Inc. DOI: 10.1561/1500000009.
- Saravanan, M., Ravindran, B., and Raman, S. (2009a).

- Automatic identification of rhetorical roles using conditional random fields for legal document summarization. In *International Conference on Artificial Intelligence and Law*, pages 231–232. Available at: <https://aclanthology.org/I08-1063.pdf>.
- Saravanan, M., Ravindran, B., and Raman, S. (2009b). Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law*, 17(2):101–124. DOI: 10.1007/s10506-009-9075-y.
- Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*. DOI: 10.48550/arxiv.2211.05100.
- Schweighofer, E. (2010). Semantic indexing of legal documents. In *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, pages 157–169. Springer. DOI: 10.1007/978-3-642-12837-0.
- Souza, E., Moriyama, G., Vitorio, D., de Carvalho, A. C., Félix, N., Albuquerque, H. O., and Oliveira, A. L. (2021). Assessing the impact of stemming algorithms applied to brazilian legislative documents retrieval. In *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 227–236. SBC. DOI: 10.5753/stil.2021.17802.
- Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: pretrained bert models for brazilian portuguese. In *Brazilian conference on intelligent systems*, pages 403–417. Springer. DOI: 10.1007/978-3-030-61377-8_8.
- Su, Y., Lan, T., Wang, Y., Yogatama, D., Kong, L., and Collier, N. (2022). A contrastive framework for neural text generation. *arXiv preprint arXiv:2202.06417*. DOI: 10.48550/arxiv.2202.06417.
- Superior Tribunal de Justiça (2024). Stj registra crescimento de 5% no número de processos recebidos. Available at: <https://www.stj.jus.br/sites/portalp/Paginas/Comunicacao/Noticias/2024/01072024-STJ-registra-crescimento-de-5--no-numero-de-processos-recebidos.aspx>. Acessado em: 5 nov. 2024.
- Tinarrage, R., Ennes, H., Resck, L. E., Gomes, L. T., Ponciano, J. R., and Poco, J. (2024). Empirical analysis of biding precedent efficiency in the brazilian supreme court via similar case retrieval. *arXiv preprint arXiv:2407.07004*. DOI: 10.48550/arxiv.2407.07004.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30. DOI: 10.48550/arxiv.1706.03762.
- Wagner Filho, J. A., Wilkens, R., Idiat, M., and Villavicencio, A. (2018). The brwac corpus: a new open resource for brazilian portuguese. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*. Available at: <https://aclanthology.org/L18-1686.pdf>.
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2020). mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*. DOI: 10.18653/v1/2021.naacl-main.41.
- Yoon, J., Junaid, M., Ali, S., and Lee, J. (2022). Abstractive summarization of korean legal cases using pre-trained language models. In *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pages 1–7. IEEE. DOI: 10.1109/imcom53663.2022.9721808.
- Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*. DOI: 10.48550/arxiv.2205.01068.
- Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675. DOI: 10.48550/arxiv.1904.09675.
- Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley. Book.