information

MDPI

Article

Evaluation of Automatic Legal Text Summarization Techniques for Greek Case Law

Marios Koniaris [1,*], Dimitris Galanis [1], Eugenia Giannini [2] and Panayiotis Tsanakas [1]

[1] Division of Computer Science, School of Electrical and Computer Engineering, National Technical University of Athens, Iroon Polytechniou 9, Zographou Campus, 15780 Athens, Greece
[2] Department of Humanities Social Sciences and Law, School of Applied Mathematical and Physical Sciences, National Technical University of Athens, Iroon Polytechniou 9, Zographou Campus, 15780 Athens, Greece
* Correspondence: mkoniari@central.ntua.gr

Abstract: The increasing amount of legal information available online is overwhelming for both citizens and legal professionals, making it difficult and time-consuming to find relevant information and keep up with the latest legal developments. Automatic text summarization techniques can be highly beneficial as they save time, reduce costs, and lessen the cognitive load of legal professionals. However, applying these techniques to legal documents poses several challenges due to the complexity of legal documents and the lack of needed resources, especially in linguistically under-resourced languages, such as the Greek language. In this paper, we address automatic summarization of Greek legal documents. A major challenge in this area is the lack of suitable datasets in the Greek language. In response, we developed a new metadata-rich dataset consisting of selected judgments from the Supreme Civil and Criminal Court of Greece, alongside their reference summaries and category tags, tailored for the purpose of automated legal document summarization. We also adopted several state-of-the-art methods for abstractive and extractive summarization and conducted a comprehensive evaluation of the methods using both human and automatic metrics. Our results: (i) revealed that, while extractive methods exhibit average performance, abstractive methods generate moderately fluent and coherent text, but they tend to receive low scores in relevance and consistency metrics; (ii) indicated the need for metrics that capture better a legal document summary's coherence, relevance, and consistency; (iii) demonstrated that fine-tuning BERT models on a specific upstream task can significantly improve the model's performance.

Keywords: automatic text summarization; case law summarization; legal information; summarization evaluation

Citation: Koniaris, M.; Galanis, D.; Giannini, E.; Tsanakas, P. Evaluation of Automatic Legal Text Summarization Techniques for Greek Case Law. *Information* **2023**, *14*, 250. https://doi.org/10.3390/info14040250

Academic Editor: Kostas Stefanidis

Received: 20 March 2023
Revised: 12 April 2023
Accepted: 17 April 2023
Published: 21 April 2023

1. Introduction

The amount of legal information, available online in digital format, is vast and constantly growing. The excessive quantity of legal documents, manifested in various forms as statutes, regulations, judicial decisions, legal opinions, scholarly articles, and other legal and para-legal documents, can be overwhelming for legal professionals, making it difficult and time-consuming to find relevant information and keep up with the latest legal developments.

With the continuous expansion of legal information available online, attention towards techniques that have the potential to save time and reduce the cognitive load of legal professionals will progressively increase. Consider, for example, a lawyer preparing his/her arguments for a given case. He/she has to iteratively browse an enormous number of judgments selecting, through knowledge and experience, relevant passages, in order to acquire the needed in-depth context understanding. Browsing through condensed versions of the judgments is intuitively easier and less time-consuming, allowing them to focus on the main ideas and, thus, acquire a better understanding. Acknowledging the aforementioned problem, some Courts and commercial/proprietary legal information

retrieval systems (e.g., LexisNexis, Westlaw, Bloomberg Law) offer summaries of judicial decisions, "hand crafted" by a specialized team of experts. Provisioning of tools to fully or partial automate the process can save time, reduce costs, and allow highly experienced legal professionals to focus on higher-level tasks utilizing their unique skills and expertise.

There has been extensive work on automatic text summarization [1], where the key idea is to produce a shorter (summary) version of a document that represents the most-important or -relevant information within the original content. Extractive and abstractive text summarization are two common techniques used to generate summaries of documents. The former involves selecting and extracting the most-important sentences or phrases from the original text and using them to create a summary. On the contrary, the latter involves creating a summary that uses new wording and phrasing to convey the main ideas of the original text.

Automatic text summarization methods can be highly beneficial for legal information processing [2]. However, applying text summarization techniques to legal documents poses several challenges due to the complexity of legal documents and the unique characteristics of the legal language they convey. Legal documents are not only precise in their language and meaning, to avoid ambiguity and misinterpretation, but also are written in a formal style, using specialized vocabulary specific to the legal field [3]. Furthermore, legal documents often contain extensive amounts of text, follow a specific structure (e.g., headings, sections), and refer to other authoritative legal documents [4]. Additionally, legal documents typically carry a certain authority and create a legally binding obligation, often within specific time frames. As errors can have significant consequences, it is apparent that distinct features of legal documents should be properly incorporated into text summarization techniques to provide accurate and effective summaries of legal documents.

While the most-reliable way to evaluate the effectiveness of an automatic summarization method is to have humans read the original text and the summary and judge its quality, manual assessment is expensive, subjective, and not applicable to large collections [5]. Typically, text summarization techniques are evaluated by automatically applying a set of predefined metrics (e.g., ROUGE [6] and its variants [7], BLEU [8], BertScore [9]). This one-size-fits-all strategy poses additional challenges when it comes to evaluating the effectiveness of automatic summarization methods for legal documents: Which metric is most appropriate for a given task?

A key component for evaluating the effectiveness of an automatic summarization method is reference summaries, which are summaries of a document that have been created by humans and are considered to be high-quality summaries. While a few datasets provide reference summaries for evaluation purposes, in many cases, especially in linguistically under-resourced languages, such as the Greek language, reference summaries may not be available, making it impossible to automatically compare the performance of different legal automatic summarization methods.

In this paper, we address automatic summarization methods for Greek legal documents. There exists no dataset in the Greek language that is tailored for automated legal document summarization. To overcome this crucial obstacle, this paper introduces a dataset of Greek Court judgments that has been developed specifically for this purpose. The paper elaborates on the process of generating the dataset, outlines its characteristics, and compares it with other text summarization datasets (Section 3).

Additionally, we employed and adapted to the specific context well-known extractive summarization algorithms, LexRank and Biased LexRank. We also modeled abstractive summarization as a sequence-generation task, by training and utilizing an encoder–decoder deep learning model based on the BERT architecture (Section 4). We evaluated both of our approaches, extractive and abstractive, by conducting a human evaluation study, as well as utilizing automatic evaluation metrics, providing a detailed comparison of different variations of our extractive and abstractive summarization methods (Section 5).

In this work, to the best of our knowledge, we employed the first study on evaluating state-of-the-art methods for summarizing Greek legal judgments. We pre-trained and

evaluated our models with promising results on a dataset of Greek Court judgments we correspondingly developed for this task. Our automated evaluation revealed that fine-tuning BERT models on specific upstream tasks can significantly improve the models' performances; incorporating the case category tags into the extractive models offered no noticeable improvements, whereas for the abstractive ones, this resulted in greatly increased performance. According to the evaluations by legal experts, extractive methods exhibit average performance, while abstractive methods produce text that is moderately fluent and coherent. However, abstractive methods receive low scores in terms of relevance and consistency metrics. This may suggest that legal professionals favor methods that are factually aligned with the judgment text, methods that accurately represent the facts and information presented in the original text. Finally, we noticed the need for standardized practices in manual summary writing and better-curated datasets.

The remainder of this paper is organized as follows: Section 2 reviews previous work, while it stresses the differentiation and contribution of this work. Section 3 introduces our dataset and compares it with other text summarization datasets. Section 4 presents our summarization methods, while Section 5 describes our experimental results and discusses their significance. Finally, we draw our conclusions and present future work in Section 6.

2. Related Work

Several lines of work are related to the present paper. In this section, we first present related work on datasets for legal document summarization, afterwards on extractive and abstractive methods for legal document summarization, followed by methodologies used in evaluating text summarization systems.

2.1. Datasets for Legal Document Summarization

While a plethora of datasets are available for general-purpose text summarization, only a few datasets exists for legal document summarization. The Rechtspraak dataset consists of 403,585 legal judgments written in Dutch, from the Rechtspraak Court (https://www.rechtspraak.nl/, accessed on 16 April 2023). Each judgment text comes with the Court's summary, the category label corresponding to the case, and the Court's verdict on the case. The BillSum [10] dataset contains 22,218 U.S. Congressional and 1237 Californian bills, both with their corresponding reference summaries. The authors in [11] collected 17,347 judgments by the Supreme Court of India, spanning the years 1990–2018, along with their summaries created by *Westlaw* India (https://www.westlawasia.com/, accessed on 16 April 2023). The Multi-LexSum [12] dataset consists of 40,000 federal U.S. large-scale civil rights lawsuit documents. The dataset also contains abstractive summaries written by experts for 9000 of the documents, with the summaries coming in different granularities: from tiny (25 words) to long (650 words). The dataset introduced in [13] contains 28,733 legal cases that took place in Canadian Courts, along with their corresponding human-generated summaries. The dataset introduced in [14] contains 35,000 veteran's appeal cases. For a subset of the dataset, the authors provided extractive summaries, additional abstractive summaries, and thematic classification data. Reference [15] presented a dataset for the summarization of legal texts in Portuguese, containing 10,623 decisions from the Brazilian Federal Supreme Court.

Differences in legal and judicial systems across countries result in dataset variations, even for datasets in the same language. The structures and writing styles of Court judgments can vary significantly, based on the Court that issued them, making thematic segmentation techniques difficult to adapt across datasets.The issue of dataset biases in text summarization research [16] has not been systematically studied in the legal domain, possibly due to the absence of standardized datasets and differences between legal text summarization datasets. Therefore, it is unknown the extent to which biases recorded in the text summarization literature, such as layout bias [16] and extractive bias [17], apply to the legal summarization domain. In terms of annotator agreement, work on thematic segmentation for Court judgments found it to be moderate, suggesting that the task is not

under-constrained [18]. While few studies [12,19] investigated the creation of detailed annotation guidelines, the legal text summarization literature lacks a systematic examination of position bias.

While there is a plethora of methods, tools, and resources for processing text in high-density languages (e.g., English), this is not the case for other languages, lower-density languages, such as Greek [20]. Our dataset consists of Greek Court decisions originating from the Greek Supreme Civil and Criminal Court. Each case comes with corresponding metadata, thus enabling easier searching and filtering through the corpus and facilitating research on a low-resource language such as Greek.

2.2. Extractive and Abstractive Methods for Legal Document Summarization

Resent literature on automatic text summarization was reviewed in [21], while summarization of legal documents was reviewed in [2]. Research on extractive methods for legal document summarization was initially based on feature-based approaches, such as LetSum [22] and CaseSummarizer [23], which use domain-specific linguistic features and relative position information to score and select sentences. Graph-based methods, such as the one proposed in [24], construct sentence graphs and select key sentences based on their keyword strength and complementary sentences that explain facts, proofs, or rules. The performance of unsupervised extractive algorithms such as TextRank for plain English summarization of contracts was explored in [25], where it was shown that extractive algorithms under-perform because of the linguistic and abstractiveness differences between the reference summaries and the input legal text. In this work, we also utilized legal domain features and graph-based methods as a way to make the content of the legal documents, notably cases, more easily accessible by evaluating their performance in a linguistically underdeveloped domain.

The authors of [14] trained a CNN classifier to predict appeal outcomes in U.S. Board of Veterans cases, using sentences that are highly predictive of the outcome. They concluded that the predictive quality of a sentence concerning the case's output is not necessarily correlated with its informativeness in a summary. In [18], an extractive summarization labeled dataset was created using sentences similar to the case's abstractive reference summary.

Apart from extractive neural models, the performance of deep neural networks in abstractive summarization of legal Court cases was explored in [26]. It was shown that the abstractive summaries generated by attentive LSTMs and Seq2Seq Transformer networks are similar in fluency to human-generated summaries. However, in some cases, they may be completely irrelevant to the input text and mention nonfactual information.

2.3. Evaluation of Automatic Text Summarization

To assess Automatic Text Summarization (ATS) models, automatic and/or human evaluation methods can be employed. Automatic methods utilize metrics that compare system summaries with, often human-generated, reference summaries and are generally considered faster and less expensive than human evaluation methods. The most-commonly used automatic metric for ATS evaluation is the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [6]. ROUGE represents a family of similar metrics, such as ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S, and ROUGE-SU, that measure the overlap of n-grams (contiguous sequences of words) between the system and reference summaries. Automatic methods can be useful for measuring the informativeness of a summary, but can be limited in their ability to measure the quality of a summary. They may not capture important aspects such as the coherence, fluency, and overall readability of the summary [16].

Therefore, it is not uncommon to have human annotators score the quality of an automatically generated summary. The human evaluation data can be used to improve the metrics used to evaluate summarization tasks [27] and provide a more robust framework of evaluating summaries. Recent literature that employs human annotators assesses the quality of the summary based on different criteria such as relevance, consistency, fluency, and coherence [16]. Other researchers [28] explored the factuality and faithfulness of

the generated summaries. Abstractive summaries tend to be unfaithful to the source text, and there is active research interest in quantifying the percentage of valid summary "hallucinations" [28].

There is currently a lack of analysis regarding the performance of different families of summarization models (such as extractive vs. abstractive) when applied to legal case documents. We believe that a combination of automatic and human evaluation approaches can yield a more comprehensive and dependable evaluation of the quality of summarization, given that both methods have their own advantages and limitations. To this end, in this work, we utilized both approaches: automatic evaluation metrics allow for fast evaluation of a large number of summaries, without any human supervision, providing with best-performing methods that were further explored, in terms of a human evaluation study, to measure legal professionals' perceived quality of the summaries they generate.

3. A Summarization Dataset for Greek Case Law

A dataset plays a significant role in the development and improvement of legal document summarization models. It allows researchers and developers to compare the performance of various models and identify areas where enhancements can be made. At the same time, the quality of a text summarization dataset is critical to the performance of the machine learning models trained on it. Having a diverse and representative dataset, with a variety of document types, topics, and writing styles is important because it ensures that the models can generalize well to new documents and are effective across different types of documents and domains.

Generating a dataset demands a substantial investment of time and resources, particularly when ensuring that the dataset is of superior quality and faithfully reflects the domain. To ensure the accuracy and reliability of the dataset, we narrowed our search space to reputable sources offering publicly accurate and reliable information i.e., the highest-ranking Courts' official websites. Another criterion we focused our effort on was to ensure that the dataset is diverse and covers different legal domains, such as criminal law, civil law, and administrative law. Last but not least, as text summarization techniques are more useful for complex and lengthy documents, we considered selecting documents that are at least a few pages long and contain formal language and legal terminology that may be challenging for non-experts to understand.

In this section, we firstly briefly describe the role of case law in Greece, then we describe the legal corpus we collected, proceed to provide information regarding token-level statistics of the dataset, and finally, compare it with other text summarization datasets.

We published our dataset (https://huggingface.co/datasets/DominusTea/GreekLegalSum, accessed on 16 April 2023) with clear documentation and instructions on how it can be used, hoping that other researchers and practitioners may find it useful for text summarization tasks regarding Greek legal documents. Furthermore, all code, materials, and the baseline and best-performing models for this paper are openly available on GitHub (https://github.com/DominusTea/LegalSumPaper, accessed on 16 April 2023).

3.1. Case Law in Greece

The two main families of judicial systems in the world are the Common Law system, derived from the English legal system, and the Civil Law system, derived from Roman law. In the former, legal decisions made by judges in previous cases are considered as binding precedent for future cases, while, in the latter, laws are primarily codified and are based on statutes, rather than on case law.

The sources of Greek law are (a) legislation, that is statutes enacted by the State, generally accepted rules of international law, and European Union law, and (b) customs, whose importance though is extremely limited currently. Judicial rulings do not qualify as a source of law; by contrast, they are only binding as to the specific case under judicial review [29].

Justice in Greece is one of the three functions of the State. According to the principle of the separation of powers, the judiciary is independent of the legislative and executive authorities. Courts in Greece are divided into the following main categories: (a) Administrative Courts, (b) Civil Courts and (c) Criminal Courts. In terms of Court hierarchy, the Hellenic Council of State (http://www.adjustice.gr/, accessed on 16 April 2023) is the Supreme Administrative Court of Greece, and the Supreme Civil and Criminal Court of Greece (AreiosPagos) (http://www.areiospagos.gr/, accessed on 16 April 2023) is the Supreme Court of Greece for civil and criminal law; the decisions of AreiosPagos are final and can be appealed to the European Court of Human Rights.

3.2. Dataset Creation

Upon surveying the Greek Courts' websites, we found two Supreme Courts to have digitized part of their rulings, making them through an anonymization process publicly available, applicable for our task: the Supreme Civil and Criminal Court of Greece (AreiosPagos) and the Hellenic Council of State. Upon further investigation, the Hellenic Council of States focuses on administrative law and does not provide summaries for the cases published. On the contrary, AreiosPagos covers both civil and criminal law areas and, at the same time, provides annotations such as summaries, keyphrases, and topics for part of the contained cases. Since having reference summaries was a crucial aspect, for both training the methods that use neural networks (Section 4.2) and evaluating our methods (Section 5), we directed our effort towards the AreiosPagos Court. We developed a web crawler, using Python's SCRAPY (https://scrapy.org/, accessed on 16 April 2023) framework, and collected the Court's decision main text, summary, and corresponding category and classification tags. Decisions' metadata, such as the date, type of Court, and category tags, were inferred from the main text of the decision. Our corpus contains 8395 Court decisions from the Supreme Civil and Criminal Court of Greece along with their summaries, 6370 of which are classified with one or more tags.

Choosing the training, validation, and test split ratio is an important step in building a machine learning model. The choice of the split ratio depends on several factors, including the size and complexity of the dataset, the type of problem you are trying to solve, and the available computational resources. To train and evaluate our model, we divided our dataset into three portions: 70% for training, 15% for validation, and 15% for testing. The impact of different training data sizes and training–test split ratios on the performance of BERT models for text summarization tasks was studied in [30]. Table 1 provides the basic statistics for the AreiosPagos dataset and the sizes of the models we trained.

Table 1. Basic statistics for AreiosPagos dataset and trained models.

AreiosPagos Dataset		
Property	**# of Dataset Entries**	**# of Models (Train/Val/Test)**
Cases with Summary	8395	5888/1269/1238
Cases with Classification	6370	4458/956/956

While the structure of judgments may vary depending on the Court and the type of case, generally in Greece, they follow a format similar to our cases' format:

- Heading: This includes the name of the Court and its division, the case number, and the year of the judgment.
- Summary (optional): The summary comprises the key subjects that were deliberated in each case, along with the judges' verdicts. It is written by legal experts.
- Introduction: This section provides a brief overview of the case and the issues at stake. Relevant facts of the case as they were established during the trial are also noted.
- Legal analysis: This is the main part of the judgment, where the Court applies the relevant laws to the facts of the case and offers its legal reasoning for the decision.

- Decision: This section contains the Court's final ruling, which may include an order for the losing party to pay damages, comply with certain obligations, or face criminal penalties.
- Signature: The judgment is typically signed by the presiding judge or judges.

Along with the main text and summary, for each decision, we captured additional metadata that correspond to the order number and type of Court case, the year the decision was issued, and the division of the Court that issued it. The metadata are presented in Table 2.

Table 2. Metadata of the Supreme Civil and Criminal Court judgments' (AreiosPagos) dataset. Metadata that were automatically inferred, using the judgments' main text, are labeled with a ✓ in the *Inferred* column.

AreiosPagos Metadata			
Metadata	**Data Type**	**Inferred**	**Description**
Case category	String		The general category that each Court case is classified into by the Court legal editors. Each case belongs to one category.
Case tags	String		The category tags that correspond to each Court case, as classified to by the Court legal editors. Each case may have multiple tags.
Court division	String		The specific division and its type (e.g., penal, civil, etc.) of the Court that issued the decision.
Issue Year	Integer	✓	The year of the decision.
Court's case identifier	String	✓	The identifier given by the Court for the particular case. It is unique among cases judged by the same Court division, but not across them.
Source URL	String	✓	The URL to the original HTML web page of the Court from which the text, summary, and metadata were sourced.

3.3. Dataset Characteristics

Our dataset spans more than a quarter of a century, the years 1990–2018, and is organized into 504 unique Court case categories. The data are over-dispersed over the categories' labels as the average category frequency is 0.198% with the standard deviation equal to 0.5549%. Furthermore, the category labels correspond to quite different other Court cases, indicating high diversity in our dataset. Figure 1 highlights the 10 most-frequent case category labels.

We further explored the lexical properties of our dataset by calculating statistics concerning the average length in tokens, the average number of sentences, and the average token in every sentence. A comparison of our dataset with other widely accepted text summarization datasets is shown in Table 3. Our Court decisions' dataset contains longer documents and summaries, both in terms of tokens and in terms of sentences. Furthermore, sentences in the main texts are also significantly longer than sentences in news domain datasets.

In order to analyze the similarities between each Court decision text and its corresponding summary, we reflected the extractive fragment analysis found in [17] and compared our dataset with news domain datasets. Each reference summary was divided into segments such that each segment corresponds to the longest-possible segment of consecutive words found both in the reference summary and the main text. Let T, S be a text, summary pair and $\mathcal{F}(A, S)$ be the set of the corresponding extractive fragments. The extractive fragment coverage measures the percentage of words in the summary that are also extractive fragments; that is, they can also be found in the main text.

$$\mathrm{C} = \frac{1}{|S|} \sum_{f \in \mathcal{F}(A,S)} |f| \tag{1}$$

In order to measure how extractable a summary from a text is, the extractive fragment density metric is defined, which attributes higher density scores to texts that have longer extractive fragments:

$$\mathrm{D} = \frac{1}{|S|} \sum_{f \in \mathcal{F}(A,S)} |f|^2 \tag{2}$$

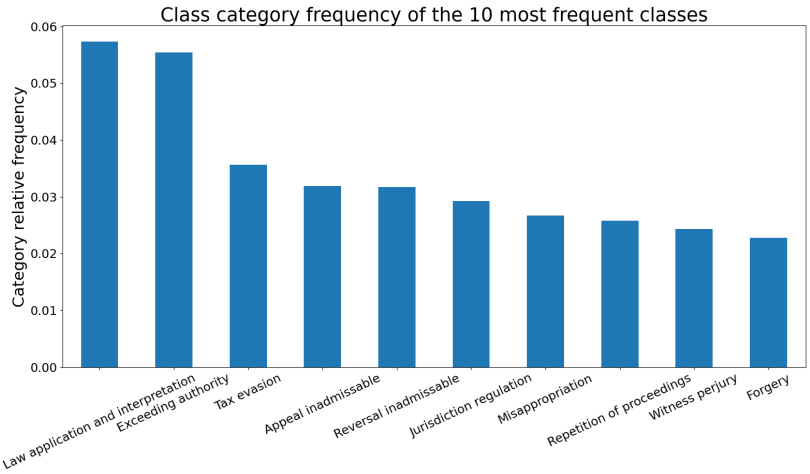


Figure 1. The 10 most-frequent categories in the AreiosPagos dataset. The x-axis corresponds to the category labels. The y-axis corresponds to the absolute frequency of each category. Original Greek labels: Νόμου Εφαρμογή και ερμηνεία, Υπέρβαση εξουσίας, Φοροδιαφυγή, Εφέσεως απαράδεκτο, Αναιρέσεως απαράδεκτο, Κανονισμός αρμοδιότητας, Υπεξαίρεση, Επανάληψη διαδικασίας, Ψευδορκία μάρτυρα, Πλαστογραφία.

Table 3. Statistical properties of the text summarization dataset using the average ratios of token-level lengths for documents and summaries and their sentences. AreiosPagos and Rechtspraak are legal Court case text datasets, while Newsroom and CNN-DailyMail are news domain summarization datasets. The results on the Newsroom dataset are reported from [17]. The results on the CNN-DailyMail and Rechtspraak datasets are reported from [31]. The upper part of the table presents statistics on the judgments' main texts, while the lower part presents the statistics on the judgment summaries.

Dataset Comparison on Token-Level Length Statistics				
Statistical Property	**AreiosPagos**	**Rechtspraak**	**Newsroom**	**CNN-DailyMail**
# of Documents	8395	403,585	1,321,995	311,672
Avg.tokens/doc	3169.06	2341.5	658.6	766.0
Avg.sent/doc	77.00	140.6	-	29.74
Avg.tokens/sent	40.6	16.6	-	25.75
# of Summaries	6370	403,585	1,321,995	311,672
Avg.tokens/sum	84.6	62.1	26.7	25.75
Avg.sent/sum	5.12	3.41	-	3.72
Avg.tokens/sent	18.2	-	-	-

Our dataset's results compared to other datasets are illustrated in Figure 2. Our dataset is rather diverse in terms of coverage, as it consists of texts whose summary may or may not contain many words found in the text. In terms of density, our dataset is most similar to the DUC 2003–2004 dataset, having a relatively low density score, which means that the reference summaries can be modeled by more sentence extractions than the extractions needed for other datasets. We also found moderate variance in the density axis, indicating that some judgments may be summarized by less sentence extractions than others. Overall, the provided lexical overlap is a good indicator of a candidate summary's quality; the aforementioned remarks imply that extractive summarization methods may

generate useful summaries for our dataset, but may struggle in summarizing judgments whose summary is rather abstractive in nature.

Furthermore, our dataset exhibits high compression scores, as the average length ratio of text to summary is 37, which is three-times bigger than the compression score of news domain datasets. This indicates that summarizing Court judgment texts can be more computationally expensive compared to news domain datasets and that pre-trained neural network methods that have a fixed input size constraint may need to be retrained with a bigger input size cap or used as they are, but with their inputs truncated or condensed.

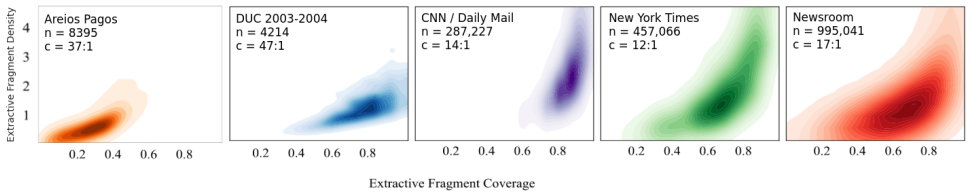


Figure 2. A comparison of the extractive fragment coverage - extractive fragment density relationship for the AreiosPagos dataset compared to other text summarization datasets. Data observations are plotted using a kernel density estimate method. *n* denotes the number of documents in the dataset, and *c* refers to the compression ratio of the main text's length over the summary's length. Leftmost: the AreiosPagos dataset. Right: news domain datasets as reported in [17]. The AreiosPagos dataset is homogeneous in terms of coverage, as a moderate percentage of words in the summary appear also in the main text. In terms of extractive fragment density, the AreiosPagos dataset shows less variance than the other datasets, while having generally low scores.

4. Method

In this section, we describe the models we propose for tackling the problem of summarizing Greek judicial decisions. We first describe our extractive summarization methods and, afterwards, the abstractive ones.

4.1. Extractive Summarization Methods

Extractive summarization methods generate a summary by extracting the most-important sentences from the text. From here on, when we refer to sentences in this section, we do not necessarily mean grammatical sentences, but sequences of tokens. Similarly with words, we generally refer to tokensṪhe process of extractive summarization typically involves the following steps:

1. Text pre-processing: Input text is pre-processed to remove any unwanted characters, convert the text to lowercase, and remove any stopwords, to reduce the computational complexity of the summarization process and improve the quality of the summary. Stemming, lemmatization, and tokenization are other processing techniques that may be applied to the text during the pre-processing step.
2. Text processing:
   (a) *Sentence scoring*: Various techniques and heuristics, such as frequency-based, graph-based, or machine learning scoring, can be utilized to assign a relevance score to each sentence in the text.
   (b) Sentence selection: After the sentences have been scored, the next step is to select the most-important sentences based on their scores. This involves setting a threshold for the sentence score and selecting the sentences that meet or exceed the threshold criteria.
3. Text post-processing: Finally, selected sentences are combined to create a summary of the text. The length of the summary can be controlled by setting a limit on the number of sentences or words in the summary. It may be necessary to re-arrange the selected sentences after they have been identified as important, to improve the coherence of the summary.

Figure 3 provides a visual overview of the basic steps of extractive text summarization: text pre-processing, processing, and post-processing.

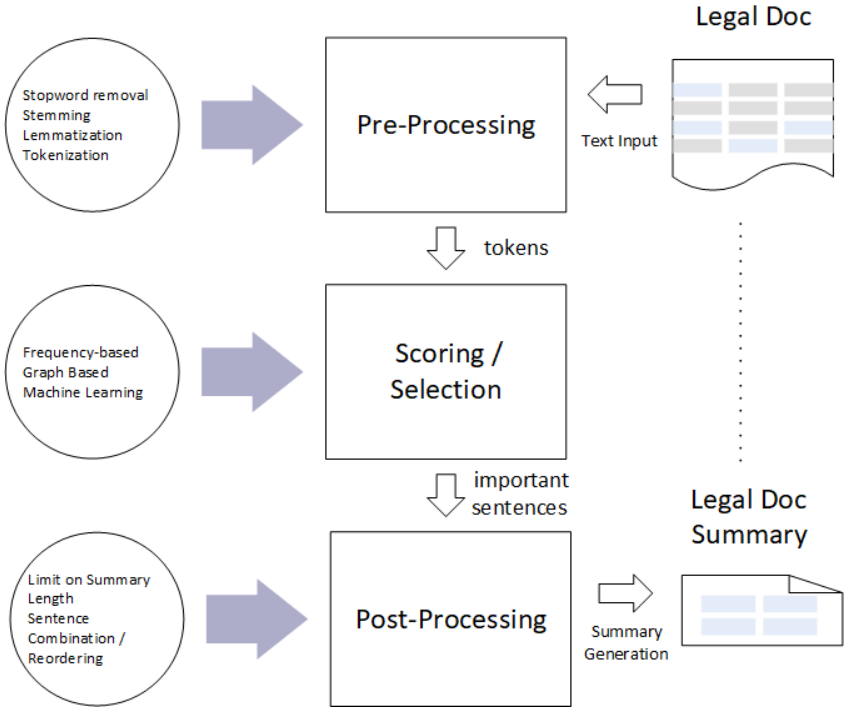


Figure 3. A general workflow of an extractive text summarization system. A combination of techniques/heuristics may be applied to the text at each step, to identify the most-important information from the original text and present it in a concise and readable format.

4.1.1. Pre-Processing Pipeline

The extractive summarization models we utilized require each input text to be segmented into its sentences and each sentence to be tokenized into separate words. Additionally, due to the Greek language's complexity (i.e., word declension), further pre-processing steps, such as stopword removal and word lemmatization, are performed to avoid a rapid increase of the vocabulary size. Specifically, we implemented the following pre-processing pipeline:

- Sentences were separated from each other, paying special attention to domain-specific acronyms that could potentially lead a punctuation sentence segmenter to end a sentence prematurely.
- Each sentence was tokenized into separate words using spacy's dependency parser.
- Stopwords were removed, and tokens were lemmatized.
- If token vectors are required by the algorithm, the pipeline returns the vectors in the language module utilized.

4.1.2. Processing Pipeline

In this work, we employed two widely used graph-based summarization methods, LexRank and Biased LexRank. Graph-based approaches utilize a graph model of the sentences where each node denotes the similarity between each sentence. Typically, sentences whose similarity score is below a certain threshold can be thought of as non-adjacent by defining a threshold so that only significantly similar sentences are connected to each other. In this work, we chose 3% as a threshold value.

LexRank: LexRank [32] is a stochastic graph-based method for computing the relative importance of textual units. A document is represented as a network of inter-related sentences, and a connectivity matrix based on intra-sentence similarity is used as the adjacency matrix of the graph representation of sentences.

The basic intuition behind LexRank is to apply the PageRank [33] algorithm over the sentence graph:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in \text{adj}[u]} \frac{\text{sim}(u,v)}{\sum_{z \in \text{adj[v]}} sim(z,v)} p(v) \tag{3}$$

Each sentence's score measures its centrality, that is its importance in the cluster of sentences that are similar to it. This enables having multiple sentence clusters with important sentences that are different between each cluster, whereas previous centroid-based algorithms had just a centroid sentence as a template. The scores are updated iteratively until the algorithm converges. The convergence is guaranteed by the properties of stochastic matrices in a Markov chain. More specifically, a vectorized version of Equation (3) is

$$p = \Big(dU + (1-d)S\Big)^T p = A^T p \tag{4}$$

where p is the centrality scores vector, S the cross-sentence similarity matrix, and U a matrix, every element of which is equal to $1/N$.

A corresponds to a transition matrix of a Markovian chain. For p to converge into a stationary distribution, it suffices that A is such that the Markovian chain is irreducible ($\forall i, j \; \exists n : A^n(i,j) \neq 0$. This means that every state is reachable by another state. The inclusion of the term dU guarantees that) and aperiodic ($\text{gcd}\{n : A^n(i,i) > 0\} = 1, \forall i$). By inserting the dumping factor d, the convergence is guaranteed.

More specifically, we utilized different similarity functions to measure cross-sentence similarity. Let s_1, s_2 be two sentences. We evaluated the following LexRank (Equation (3)) variations, by changing the similarity function and sentence representation combinations, as follows:

- $\text{LexRank}_{\text{tf-idf}}$ (tf-idf BoW with cosine sentence similarity): This is the similarity metric used in the LexRank [32] paper, in which the cross-sentence similarity is given by the cosine distance of the tf-idf BoW sentence vectors. Formally,

$$sim_{cos\_tfidf}(s_1, s_2) = 1 - \frac{Tf - Idf(s_1) \cdot Tf - Idf(s_2)}{\|Tf - Idf(s_1)\| \times \|Tf - Idf(s_2)\|} \tag{5}$$

  where the tf-Idf vectors are calculated in a BoW fashion, by the sum of the one-hot vectors of each word in the sentence weighted by the word's idf score:

$$\text{Tf-Idf}(s) = \sum_{w \in s} 1_{index(w)} idf(w) \tag{6}$$

  where index(x) $\in [1, N_{vocab}]$ and $1_i \in [0,1]^{N_{vocab}}$ denotes the indicator function and is non-zero only at the position of the index i.
- $\text{LexRank}_{\text{com}}$ (BoW with common words sentence similarity): Cross-sentence similarity is defined as the number of words found in both sentences normalized by the sum of both sentences' lengths. Formally:

$$sim_{cw}(s_1, s_2) = \frac{|s_1 \cap s_2|}{log(|s_1|) + log(|s_2|)} \tag{7}$$

  This essentially implements the similarity function used in the TextRank algorithm [34], aside from some minor changes in the parameterization of the power method used to converge to the LexRank sentence scores.

Biased LexRank: There are many variations of the PageRank algorithm. One particularly important in query-based summarization is the Biased LexRank algorithm [35]. It

modifies Equation (3) by increasing the dumping factor for sentences that are most relevant to the query and similarly decreases it for non-relevant sentences:

$$p(u) = d\frac{rel(u,q)}{\sum_{z\in \text{Corpus}} rel(z,q)} + (1-d)\sum_{v\in \text{adj}[u]} \frac{\text{sim}(u,v)}{\sum_{z\in \text{adj[v]}} sim(z,v)} p(v) \quad (8)$$

Relevance functions that can be used include a word frequency function:

$$rel(u,q) = \sum_{w\in q} log\big(tf(w,u)+1\big)\cdot log\big(tf(w,q)+1\big)\cdot idf(w) \quad (9)$$

or in the case of vector embeddings, vector similarity functions, such as the cosine distance function:

$$rel(u,q) = \frac{u\cdot q}{||u||\cdot||q||} \quad (10)$$

The Biased LexRank algorithm changes the way the damping factor is distributed to each sentence, from attributing it uniformly to biasing it according to a prior belief on the importance of each sentence.

We implemented the Biased LexRank algorithm, by extending our default LexRank implementation and utilizing the semantic similarity of each sentence with the judgment's tags/categories, as described in Section 3.2. The semantic similarity of each sentence with the judgment tags is calculated using the common words sentence similarity function.

4.1.3. Post-Processing Pipeline

Sentences extracted from the text are concatenated, in the order they appear in the input text, to form the generated summary. An exact match between our generated extractive summaries and the reference summaries is not possible, since the latter are abstractive in nature. To ensure a fair comparison, we constrained our generated extractive summaries to be at three-times the length of our reference summaries.

4.2. Abstractive Summarization Models

Unlike extractive summarization models, which select important sentences or phrases from the original text and combine them into a summary, abstractive summarization models generate new sentences that capture the essential meaning of the source text. The basic steps of abstractive text summarization: text pre-processing, processing, and post-processing, resemble the steps of the extractive summarization models.

4.2.1. Pre-Processing Pipeline

We implemented the following pre-processing pipeline:

- Split the input, both judgment and judgment summary, into tokens. When a word is not found in the tokenizer's vocabulary, then it is split into known subwords.
- The input is truncated and/or padded to accommodate the model's hidden size constraint of 512 tokens.
- Each token is encoded into the corresponding token id.
- The input is moved to the GPU (if it is available).

4.2.2. Processing Pipeline

In this work, we employed Bidirectional Encoder Representations from Transformers (BERT) [36], a popular language representation model, applied to various NLP tasks, including text summarization. BERT is trained on massive amounts of unannotated text data, allowing it to learn rich language features and achieve state-of-the-art performance on various natural language processing tasks, such as text classification, question answering, and text generation. The Transformer model is a type of neural network that uses self-attention to capture the dependencies between words in a text. It consists of an encoder and

a decoder, each of which is composed of multiple layers of self-attention and feedforward neural networks.

In [37], a standard encoder–decoder framework for abstractive summarization was presented, where both the encoder and the decoder are multi-layer bidirectional Transformer models. In the context of text summarization, the encoder processes the input text and encodes it into a fixed-length representation, which is then used by the decoder to generate the summary. During training, the model is fed pairs of input–output sequences, and the model is trained to generate a summary that captures the most-important information in the input text.

In our materialization, both the encoder and the decoder are multi-layer bidirectional Transformer models. The Greek BERT's weights [38] were used to initialize both the encoder and the decoder, as our legal judgments domain shares similarities with the Greek language used in the European Parliament Proceedings Parallel Corpus, which was part of the training data for the Greek BERT.

Figure 4 shows the architecture of the model we utilized. In this architecture, the input text is first transformed into contextualized representations using a BERT encoder. The contextualized representations are created by considering the surrounding context of each word in the input text. This allows the model to capture the meaning of the text more accurately. Once the input text has been transformed into contextualized representations, these representations are passed into a BERT decoder. The decoder then generates a summary of the input text by sequentially predicting each output summary token, while also attending to the output tokens generated so far.

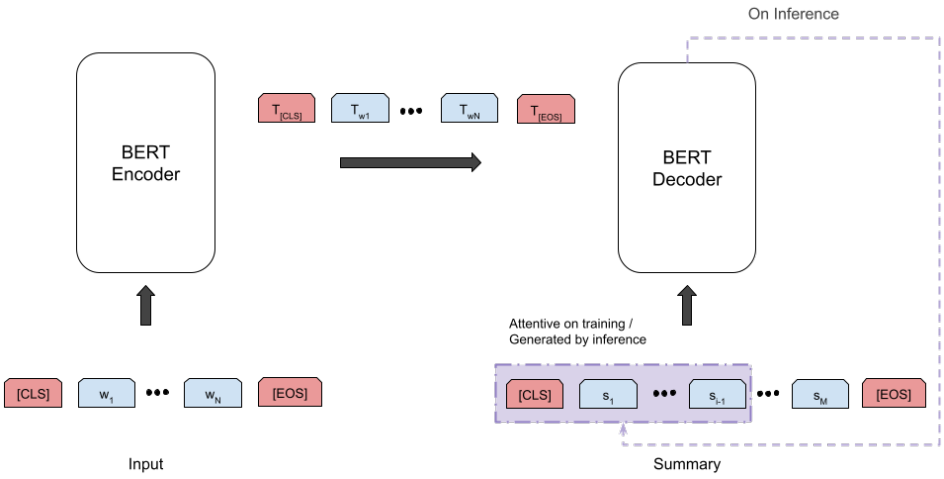


Figure 4. The architecture of the BERT encoder–decoder summarization model we utilized, a two-step process, where the input text is first transformed into contextualized representations and then passed through a decoder to generate a summary. The BERT encoder generates contextualized representations of the input sequence, which are passed to the BERT decoder for generating a summary. Tokens w_i (in blue) correspond to words in the input text, while [CLS] and [EOS] (in red) correspond to special tokens. During inference, the decoder takes as the input the output generated so far. During training, the decoder attends to the reference summary up to the token corresponding to each time step.

We trained several model variations, on the dataset we collected, using a learning rate of 4×10^{-5} and a batch size of 2 for 3 epochs, using sequence lengths of 512 tokens. The variations used are listed below and make use of the following abbreviations: *RE*: text reordering, *RM*: generic text removal, *C*: class information inclusion, *LR*: LexRank text reduction:

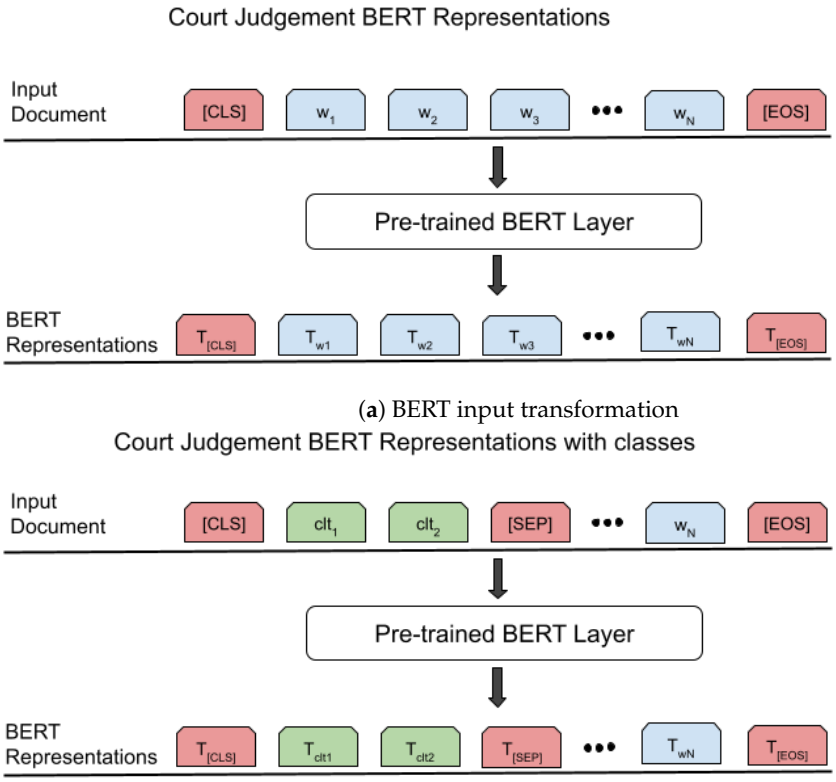
1. BERT: An encoder standard framework, where both the encoder and the decoder follow the BERT-BASE architecture of [36] and are initialized using the Greek BERT's [38] weights. The model's maximum input size was limited to 512 sub-word tokens, potentially resulting in the truncation of important information from the model's input.

This limitation served as a baseline model for the evaluation of other model variations that aim to maximize the inclusion of relevant information in the input, with the goal of improving performance, such as the Court's final ruling or classification metadata.

2. BERT (RE): In our case document format, the section that contains the Court's final ruling (decision) is near the end of the text (Section 3.2). Since the decision is one of the most-important parts in a case and should not be truncated, we moved the last part of the text (which contains the Court's decision) to the beginning of the text, ensuring that the Court's decision will never be truncated due to the model's maximum input size.
3. BERT (RE + RM): We further experimented with removing text from the beginning and the ending of each text, which correspond to general information about the date of the trial, the location it took place, and the names of the judges, the appellants, and the lawyers. The aim was to fit as much important data as possible to the max-tokens input.
4. BERT (RE + RM + C): We conducted additional experiments including the category classes, which correspond to each Court case, at the start of every input. The main text was separated by the class tokens using a special separator token ("SEP").
5. BERT (RE + RM + LR): We tested reducing the input document to half of its original size, using our $LexRank_{tf-idf}$ method. The purpose of reducing the input document size was to fit more important data into the input sequence of the summarization model. By selecting only the most-relevant and -informative sentences, the model can focus on the most-important information and generate a more concise summary, at the risk of altering the original document's coherence, as some important context or background information may be lost. The scope of this model is to test the trade-off between including more important data in the input sequence versus maintaining the coherence and completeness of the original document.
6. BERT (RE + RM + LR + C): This variation is a combination of the two previously discussed variations. Its purpose is to assess the impact of including category class information when the input is halved using our $LexRank_{tf-idf}$ method on the model's performance.

The input transformation during the BERT (+C) model variation is presented in juxtaposition with the other variations in Figure 5.

In the BERT model architecture, input text is usually transformed into contextualized representations using a BERT encoder. This input transformation process involves representing each word in the text as a contextualized vector and also adding special tokens to the text. In the BERT (+C) variations, which include class information, tokens (green color), which correspond to the class or category that the text belongs to, are added to the input text. To separate the class information from the rest of the decision's text, the [SEP] special token is used (red color).



(**a**) BERT input transformation

(**b**) BERT input transformation with class information

Figure 5. (**a**) BERT input transformation; (**b**) BERT input transformation with class information. The BERT input transformation with class information involves adding special tokens to indicate the class of the input text and separating the class information from the original text using the [SEP] token. The tokens in blue correspond to words in the input text, those in red to special tokens, and those in green to class tokens.

4.2.3. Post-Processing Pipeline

In the case of abstractive summarization, the model outputs token ids that are decoded into words using the model's tokenizer. Furthermore, special tokens, which the tokenizer adds by default, are omitted from the final summary.

5. Evaluation Setup

In this section, we outline the methodology and metrics used for the evaluation assessment, followed by a presentation of the results along with a brief discussion. Since both automatic and human evaluation have their own strengths and weaknesses, we believe that a combination of the two approaches can provide a more comprehensive and reliable assessment of the quality of the summaries generated. Therefore, in this study, we utilized both methods: automatic evaluation metrics enabled us to quickly evaluate a large number of summaries without human intervention, identify the best-performing methods, and then, further examine the best-performing methods, through a human evaluation study, to assess the quality of the summaries they produce.

For the purposes of disseminating the data collected for our Court judgment dataset and conducting a human evaluation study on automatic summarization methods for Court judgments, we developed a web application. The details of the application are presented in Appendix A.

5.1. Automatic Evaluation

Automatic evaluation of summarization is a process of using computational methods to assess the quality of a generated summary. Automatic evaluation is more efficient, as it can process a large number of summaries in a short amount of time, and more objective, than human evaluation, as it relies on a set of predefined metrics.

5.1.1. Automatic Evaluation Metrics

We utilized the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [6] lexical overlap metrics, as the most-prominent automatic evaluation metric in machine text summarization [39,40]. In particular:

- ROUGE-N measures the *N-gram* overlap between the candidate summary and a set of reference summaries, namely if we define RS to be the set of reference summaries and $GRAM(N, C)$ as the set of all *N-grams* in a candidate summary C,

$$ROUGE - N(RS, C) = \frac{\sum_{R \in RS} \sum_{g \in GRAM(N,R)} count_{matching}(g, C)}{\sum_{R \in RS} \sum_{g \in GRAM(N,R)} count(g)} \tag{11}$$

  which favors candidate summaries with common *N-grams* across multiple reference summaries, as the denominator normalizes the nominator's sum over all possible reference summaries *N-grams*. In the case of a single reference summary, the ROUGE-N definition is simplified to:

$$ROUGE - N(R, C) = \frac{\sum_{g \in GRAM(N,R)} count_{matching}(g, C)}{\sum_{g \in GRAM(N,R)} count(g)} \tag{12}$$

  The metric is recall-oriented because the percentage of overlapping N-grams is calculated over the *N-grams* found in the reference summaries.
- ROUGE-L measures the length of the Longest Common Subsequence (LCS) of words found in both the generated and reference summary. A subsequence of words is defined as a sequence of words that can be found in the original sequence in the exact relative order they appear in the subsequence. The LCS score is normalized by the candidate summary's length, when measuring recall or the reference summary's length when measuring precision accordingly. More formally, for single sentences r, c found in the reference and candidate summary, respectively, the LCS recall and precision scores are as follows:

$$LCS_R(r, c) = \frac{LCS(r, c)}{|c|} \tag{13}$$

$$LCS_P(r, c) = \frac{LCS(r, c)}{|r|} \tag{14}$$

  In order to define ROUGE-L for whole summaries, we define each sentence in both the candidate and reference summary to be a separate sequence of words. The ROUGE-L score of a candidate summary C and a reference summary R is defined as

$$ROUGE - L(R, C) = \frac{\sum_{r \in R} LCS_{\cup}(r, C)}{|R|} \tag{15}$$

  where the nominator is divided by the number of words in the reference summary in order to measure the recall and $LCS_{\cup}$ denotes the union-longest common subsequence length which, more formally, is defined as

$$LCS_{\cup}(r, C) = |\cup_{c \in C} \{largest\_subsequence(r, c)\}| \tag{16}$$

- ROUGE-W generalizes the ROUGE-L metric by assigning a different credit to each LCS depending on how consecutive its words are, this way penalizing the LCS with many non-consecutive words. The weighted LCS is calculated using a dynamic programming table that stores at each pair of word indices i, j (iterating over the reference and candidate summary, respectively) the length of consecutive matches at position i, j and a dynamic programming table that calculates the weighted LCS up to the indices i, j, awarding bigger scores to index pairs that correspond to a bigger length of consecutive matches.

We report on ROUGE-1 (unigrams), ROUGE-2 (bigrams), and ROUGE-3 (trigrams) for informativeness and ROUGE-L and ROUGE-W for fluency.

5.1.2. Automatic Evaluation Process

The automatic evaluation process involves pre-processing both the reference and evaluator highlight summaries along with the generated summaries. The Court decision's that we evaluated are sampled from the test split of the AreiosPagos subset, which is annotated with category tags. The ROUGE metrics are then utilized to compare the generated and evaluator highlight summaries with the reference summaries. Figure 6 presents a schema of the automatic evaluation process. We modified a python re-implementation (https://github.com/Diego999/py-rouge, accessed on 16 April 2023) of the original Perl ROUGE script, by adding options for stemming Greek words and improving the tokenization and sentence segmentation on our dataset. We also inserted options for removing Greek stopwords and/or stemming every word.

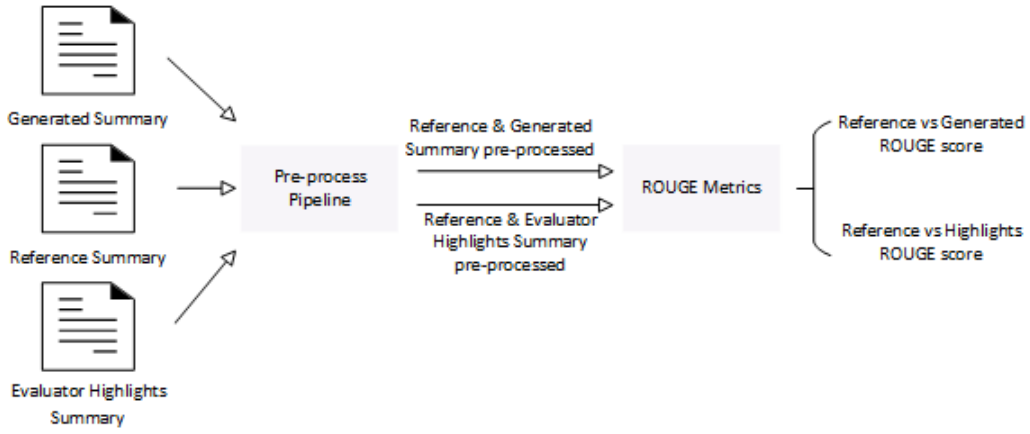


Figure 6. A schema of the automatic evaluation process. Both reference, evaluator highlight summaries and generated summaries are pre-processed. The ROUGE metrics are used to compare generated and evaluator highlight summaries versus the reference summaries.

5.1.3. Automated Evaluation Results

Results of our methods using the ROUGE automatic evaluation metric are shown in Table 4. A higher ROUGE score reflects a higher similarity between the automatically generated summary and the reference summary. The extractive summarization methods extract the most-important sentences until three-times the length of the reference summary is reached. A random sentence model, which randomly selects sentences from the document to form the summary, was used as baseline. The abstractive summarization methods generate summaries of arbitrary length, as they have learned when to end a summary during the training phase. The plain BERT model, without any special input pre-processing, was used as the baseline, to compare these methods.

We firstly evaluated our methods on the full AreiosPagos dataset, which has 8395 documents, using the dataset's train/val/test split. The first group of methods, extractive methods, are based on the LexRank algorithm:

- The random sentence method achieved the highest score for ROUGE-1 (71.50), but lower scores for all other ROUGE metrics, compared with the other extractive methods, indicating that it performed worse than the LexRank methods in summarizing the original documents.
- $\text{LexRank}_{\text{tf-idf}}$ achieved a ROUGE-1 score of 71.19, which means that it can generate summaries with around 71% overlap with the reference summaries at the unigram level. The methods achieved the highest score for ROUGE-2 (42.38), ROUGE-3 (23.27), ROUGE-L (16.84), and ROUGE-W (8.11), indicating that it outperforms the other methods in terms of capturing the content of the original documents. This method uses the tf-idf weighting scheme to determine the importance of sentences in the document.
- $\text{LexRank}_{\text{com}}$ achieved slightly lower scores than $\text{LexRank}_{\text{tf-idf}}$ for all ROUGE metrics, indicating that it is slightly less effective at capturing the content of the original documents. This method uses the intersection set of the common words between two sentences to determine which sentence is important.

The second group of methods, abstractive methods, are based on the BERT model:

- BERT: This baseline model achieved scores very close to the medium of all variation methods, with a ROUGE-1 score of 60.58 and ROUGE-2, ROUGE-3, ROUGE-L, and ROUGE-W scores of 39.48, 21.17, 14.18, and 5.29, respectively. The results indicated that, while other method variations do provide a significant improvement in generating summaries compared to the baseline, the others do not.
- BERT (RE): Rearranging text so that the case result is always included and at the start of input, this achieved slightly lower scores compared to BERT, with a ROUGE-1 score of 59.76 and ROUGE-2, ROUGE-3, ROUGE-L, and ROUGE-W scores of 38.81, 20.79, 14.79, and 5.13, respectively, indicating that rearrangement solely did not improve the effectiveness of the method.
- BERT (RE + RM): Out of the four methods evaluated, this particular method attained the highest level of performance, with scores of ROUGE-1: 60.92, ROUGE-2: 40.05, ROUGE-3: 22.08, ROUGE-L: 14.57, and ROUGE-W: 5.43. We observed that this method excelled at capturing the content of the original documents compared to the other methods.
- BERT (RE + RM + LR): This method achieved similar scores to the baseline BERT with a ROUGE-1 score of 60.60 and ROUGE-2, ROUGE-3, ROUGE-L, and ROUGE-W scores of 39.75, 21.75, 14.27, and 5.31, respectively. Reducing the input document into half of its original size, using our $LexRank_{tf-idf}$ method, actually deteriorated the performance in this case.

Afterwards, we retrained and evaluated our methods on the subset of the AreiosPagos dataset that was annotated with category tags, which had 6370 documents, using the dataset's train/val/test split. The aim of this experiment was to investigate the impact of using category tags when summarizing Court decisions. The $\text{LexRank}_{\text{tf-idf}}$, $\text{LexRank}_{\text{com}}$, Random Sentence, BERT, BERT (RE), and BERT (RE + RM + LR) methods do not utilize the case category tags and could have been omitted. Instead, we chose to include them in this experiment for the sake of comprehensiveness.

The first group of methods, extractive methods, are based on the LexRank algorithm:

- $\text{LexRank}_{\text{tf-idf}}$ was the best-performing extractive model, also in this dataset.
- $\text{LexRank}_{\text{com}}$ achieved slightly lower scores than $\text{LexRank}_{\text{tf-idf}}$ for all ROUGE metrics, except ROUGE-1 with a score of 71.51.
- Biased LexRank achieved an ROUGE-1 score of 67.73, a ROUGE-2 score of 41.05, a ROUGE-3 score of 22.06, a ROUGE-L score of 15.50, and a ROUGE-W score of 7.33. Although it performed better than the baseline model (random sentence), it underperformed with respect to the other LexRank variations, indicating that the case's tags may not always be the most-valuable feature in extracting a high-quality summary.

As before, the second group of methods, abstractive methods, are based on the BERT model:

- BERT (RE): This method achieved a slightly lower ROUGE-1 score compared to BERT (62.80). However, the other ROUGE scores were similar to those of the baseline BERT.
- BERT (RE + RM): The ROUGE-1 score decreased to 62.08, but the ROUGE-2 and ROUGE-L scores increased slightly. However, the ROUGE-3 and ROUGE-W scores were similar to those of BERT and BERT (RE).
- BERT (RE + RM + C): This method adds Category tags (C) to BERT (RE + RM). It achieved the highest ROUGE scores among all the methods, with a ROUGE-1 score of 64.24, a ROUGE-2 score of 40.40, a ROUGE-3 score of 22.27, a ROUGE-L score of 15.34, and a ROUGE-W score of 5.64. Once more, this specific method achieved the highest level of performance among the four methods that were evaluated.
- BERT (RE + RM + LR): This method did not perform as well as BERT (RE + RM + C) and achieved lower scores than the baseline BERT on most of the ROUGE metrics. As previously noted, reducing the input document into half of its original size, using our $LexRank_{tf-idf}$ method, actually deteriorated the performance.
- BERT (RE + RM + LR + C): This achieved the second-best ROUGE scores after BERT (RE + RM + C). The ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-L, and ROUGE-W scores were 63.98, 39.85, 21.90, 15.33, and 5.64, respectively.

Table 4. Automatic evaluation results for the whole dataset and its subset containing classes, presented in two segments of the table corresponding to (a) automatic extractive summarizers and (b) automatic abstractive summarizers. The extractive methods extract sentences until they reach three-times the length of the reference summary. In the abstractive models, we modified the inputs and labeled the models accordingly; *RE*: the text is rearranged so the case result is always included and at the start of the input, *RM*: less important parts of the text are removed, *C*: the case's category tags are included at the start of the input, *LR*: the input document is halved using $\text{LexRank}_{\text{tf-idf}}$. The ROUGE scores are F1-scores given in percentage (%) form. The ROUGE-L/W scores are reported without stopword removal for the BERT methods. The best-performing automatic method in each category is in **bold**.

AreiosPagos full (n = 8395)					
Method	**ROUGE-1**	**ROUGE-2**	**ROUGE-3**	**ROUGE-L**	**ROUGE-W**
Random Sentence	**71.50**	40.72	20.01	13.38	6.21
$\text{LexRank}_{\text{tf-idf}}$	71.19	**42.38**	**23.27**	**16.84**	**8.11**
$\text{LexRank}_{\text{com}}$	71.10	41.71	21.78	14.88	7.03
BERT	60.58	39.48	21.17	14.18	5.29
BERT (RE)	59.76	38.81	20.79	**14.79**	5.13
BERT (RE + RM)	**60.92**	**40.05**	**22.08**	14.57	**5.43**
BERT (RE + RM + LR)	60.60	39.75	21.75	14.27	5.31
AreiosPagos w/classes (n = 6370)					
Method	**ROUGE-1**	**ROUGE-2**	**ROUGE-3**	**ROUGE-L**	**ROUGE-W**
Random Sentence	70.64	40.26	19.77	13.41	6.28
$\text{LexRank}_{\text{tf-idf}}$	71.46	**42.90**	**23.78**	**17.29**	**8.35**
$\text{LexRank}_{\text{com}}$	**71.51**	42.10	22.02	15.09	7.09
Biased LexRank	67.73	41.05	22.06	15.50	7.33
BERT	62.90	38.52	20.64	14.37	5.28
BERT (RE)	62.80	38.51	20.39	14.19	5.21
BERT (RE + RM)	62.08	38.83	21.26	14.42	5.28
BERT (RE + RM + C)	**64.24**	**40.40**	**22.27**	**15.34**	**5.64**
BERT (RE + RM + LR)	62.01	37.89	20.32	13.71	4.99
BERT (RE + RM + LR + C)	63.98	39.85	21.90	15.33	**5.64**

Overall, we observed that the regular LexRank method outperformed the biased LexRank method, suggesting that the category tags may not be as important for sentence extraction. Additionally, among the abstractive methods, BERT (RE + RM + C) achieved the highest scores on all metrics. We observed significant improvements in the quality of the generated summaries by rearranging the input text to place the Court case's result at the beginning, incorporating the case's category tags, and removing general text that is less relevant to the case (such as dates, names of appellants and judges, etc.).

Finally, we note that higher ROUGE scores attained by extractive summarization methods when compared to abstractive ones do not necessarily imply that abstractive methods' summaries are subpar. Extractive summaries, which are composed of verbatim sentences from the original text, are more likely to have a higher n-gram overlap with the reference summary than abstractive summaries, which generate new sentences that may not contain the same n-grams as the reference summary. Furthermore, while ROUGE is a widely used metric for summarization evaluation, it is not always the most-appropriate metric to use. For instance, if the objective of the summarization task is to produce summaries that are more similar to the way humans write, and that are both fluent and grammatically correct, alternative metrics or methods (e.g., human evaluation) may be more relevant than ROUGE metrics.

5.2. Manual Evaluation

The automatic evaluation metrics we presented allow for the fast evaluation of a large number of summaries, without any human supervision. However, as those metrics factor in only lexical overlaps, their scores may not necessarily be indicative of a summary's quality, as we saw in Section 2.3. In order to gain a better understanding of our top-performing methods (in terms of automated evaluation), as well as study the correlation between the automatic metrics and human judgment, we performed a human evaluation study. The human evaluation study was conducted through our web application interface, which is presented in Appendix A.

5.2.1. Manual Evaluation Metrics

For the manual evaluation metrics, we modified the metrics introduced in [16] to our Court decisions domain:

- Relevance: The degree to which a summary has captured the important content from the judicial decision.
- Consistency: The factual alignment between the summary and the judicial decision's main text.
- Fluency: The degree to which the summary contains individually fluent/high-quality sentences.
- Coherence: Coherence measures the degree to which the main ideas of the judicial decision summary are meaningfully organized into different sentences.

In the case of extractive summaries, the evaluators are asked to evaluate the summaries only on the relevance metric, since the other metrics are not applicable to extractive summaries. The abstractive summaries generated by the BERT model are in lowercase form and contain no diacritics. Therefore, in order to avoid biasing the human evaluators, we lowercased and removed the diacritics from the reference summaries as well.

5.2.2. Manual Evaluation Process

We constructed extractive summaries using the human evaluators' highlights from each text. The human evaluators' highlights were used to extract from each text the corresponding segments and construct extractive summaries, which after being truncated to match the length of the extractive summaries generated by our extractive summarization methods, can be directly compared to them. All evaluators were assigned the same five Court judgments, and their human evaluation scores were analyzed for inter-evaluator agreement and their correlation with the ROUGE metrics measured. Furthermore two of

the human evaluators were asked to evaluate summaries for a total of fifteen additional Court judgments, thus expanding the scope of our evaluation to a wider range of Court judgments. Figure 7 provides an outline of the human evaluation pipeline.

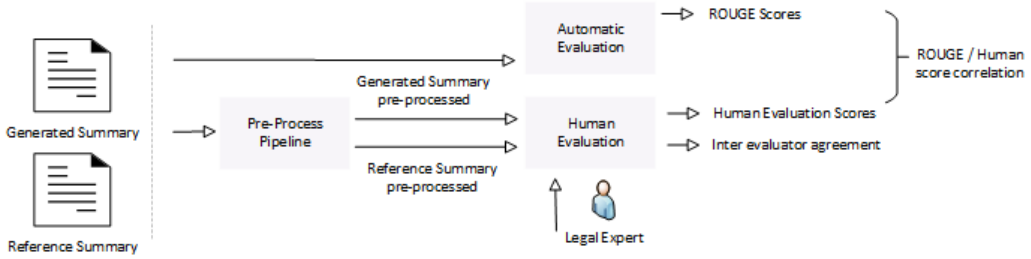


Figure 7. A schema of the human evaluation pipeline. Reference summaries were pre-processed to match the BERT-generated summaries. Human evaluation metrics were used to measure the relevance, consistency, coherence, and fluency of the produced summaries and capture the inter-evaluator agreement.

5.2.3. Study Design

Human evaluation of Court judgments summaries is a challenging task. In order to ensure high-quality standards in the evaluation, we had to limit our human evaluator pool exclusively to people with at least undergraduate-level experience in the legal domain. Furthermore, since evaluators have to devote a significant amount of time reading the Court judgment texts and evaluating summaries, using their legal domain knowledge and abilities, we selected Court judgments that had a length less than the average judgment length in the test dataset, to ensure the survey's estimated completion time was reasonable. The actual resulting average completion time was 42.5 min (std: 3.30).

The evaluators, who worked independently, i.e., without consulting one another, were instructed to:

1. Read the Court's judgment text and highlight the sentences they believed were important in creating a summary.
2. Evaluate abstractive summaries of the judgment.
3. Evaluate extractive summaries of the judgment.

5.2.4. Evaluation Interface

The web interface for the user evaluation consists of two parts: (1) Part 1, where the structure of the survey is explained, an answering guide is presented, and the participants answer questions about their legal domain knowledge; (2) Part 2, where the participants extract highlights from each judicial judgment text and evaluate extractive and abstractive summaries corresponding to each text. A flowchart schema of the survey page web interface is presented in Figure 8.

Part 1: Participants' general information and questionnaire guide. On the landing page, the survey structure and purpose are explained, and the participant is given a guide on how the questions will be formatted and how to answer them. Afterwards, the participant is asked to answer questions related to general information about him/her, such as their age, his/her legal domain educational level, and the time he/she spends each week reading Court decisions. In the following page, the participant is presented with a guide that explains the questionnaire question's format and how the participant must answer each question. First, the metrics that the participant must use are defined. Afterwards, a mock Court judgment text is given to the participant in order to familiarize him/her with the segment highlight task they will have to complete in each Court judgment text. Finally, the participant is presented with mock abstractive and extractive summaries of the text and a metric evaluation table, so they understand their format in the rest of the survey.

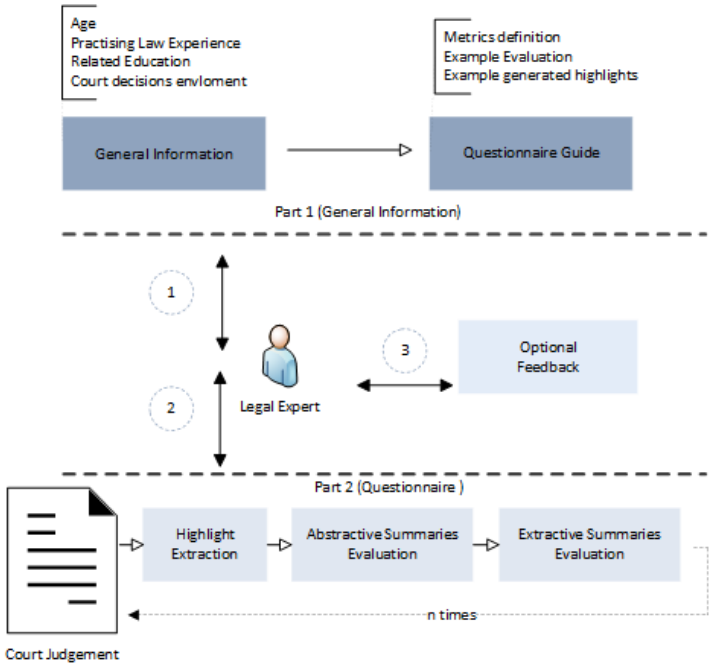
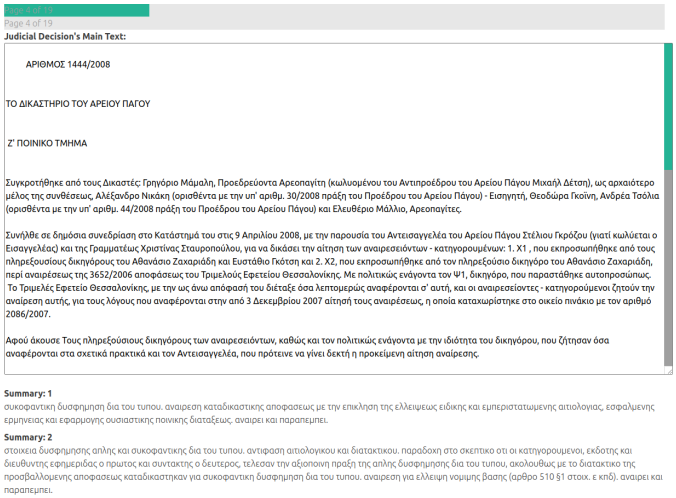


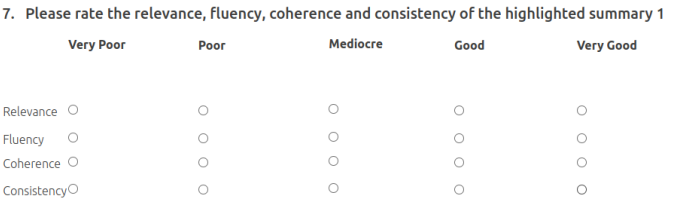
Figure 8. A flowchart schema of the survey page web interface. Each participant is firstly asked to answer questions related to general information about him/her and then presented with a guide that explains the questionnaire question's format and how to answer each question. Afterwards, the participant iteratively evaluates the extractive and abstractive summaries of the judicial judgment texts. Additional feedback may be provided at the end.

Part 2: Questionnaire on automatic summarization of court judgments. In this part of the questionnaire, the participant evaluates extractive and abstractive summaries of the judicial judgment texts. In the case of abstractive summaries, the participant is presented, in random order, the reference summary of the Court's judgment and the summary generated by our abstractive summarization method. The participant is given the Court judgment's main text and is asked to extract the segments he/she assesses to be relevant to a potential summary of the judgment. The participant selects each segment separately, which registers as a highlight after the corresponding "highlight" button is pressed. Afterwards, the participant evaluates abstractive summaries of the judicial judgment. Those consist of a reference summary, generated by the Court's legal editors, and an automatically generated summary, generated by our abstractive summarization method. The aforementioned process can be seen in Figure 9.

In the case of extractive summaries, the human evaluators were presented with the main text of the Court decision with the extracted summaries highlighted in bright yellow color (Figure 10). After reading each extractive summary separately, the participant rated it and proceeded to the next extractive summary for the same judgment text.



(**a**) The main text and a set of abstractive summaries in our survey.



(**b**) The evaluation metrics interface for abstractive summarization in our survey.

Figure 9. Two screen captures of the abstractive summarization evaluation in our survey.

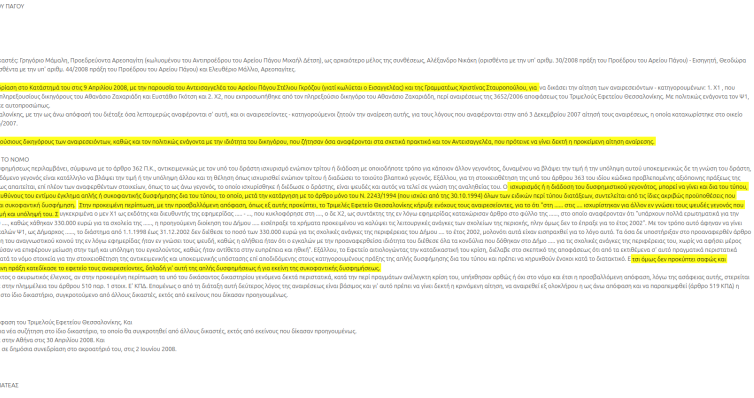


Figure 10. A screenshot of the extractive summarization evaluation in our survey. Extracted summaries are highlighted in bright yellow color.

5.2.5. Human Evaluation Results

In order to evaluate our methods' performance, we asked the human evaluators to evaluate our abstractive summaries and the reference summaries in terms of relevance, consistency, coherence, and fluency. The evaluators also evaluated our LexRank and Biased LexRank extractive summaries in terms of relevance. Aggregate results are presented in Table 5

Our expert evaluators believed that the reference summaries of the AreiosPagos dataset were relevant, fluent, coherent, and consistent with the input text. The scores for relevance, fluency, coherence, and consistency were 4.1, 3.7, 3.9, and 4.3, respectively. The $BERT(RE + RM + C)$ method achieved a relevance score of 2.6, indicating that the generated summary was not very relevant to the input text. However, it achieved relatively high scores for fluency (3.4) and coherence (3.4), indicating that the summary was well-

written and had a logical flow. The consistency score was only 2.6, indicating that the summary was not, in many cases, consistent with the input text.

Table 5. Human evaluation results, on the modified human evaluation metrics using a 1–5 Likert scale. The first section of the table compares our BERT abstractive summarization method with reference summaries. The second section of the table compares the human-evaluated relevance score of the summaries generated by the vanilla LexRank and the Biased LexRank algorithms.

Summary	Relevance	Fluency	Coherence	Consistency
Reference	4.1	3.7	3.9	4.3
BERT (RE + RM + C)	2.6	3.4	3.4	2.6
$\text{LexRank}_{\text{tf-idf}}$	3.3	-	-	-
Biased LexRank	3.4	-	-	-

We note that, in terms of fluency and coherence, our abstractive model had a similar, but lower performance to the reference summaries. That indicated that the generated text can be read easily and is internally coherent and similar, in those regards, to the reference summaries. However, our model under-performed compared to the reference summaries, in terms of relevance and consistency. This means that, compared to a reference summary, it failed to capture the relevant information from the judicial judgment and also may be factually inconsistent with it by referencing information not found in the original text. The reference summaries appeared to be much better at capturing the relevant context of the Court decision and were more factually consistent with it.

It is important to note that the reference summaries had surprisingly mediocre scores in the fluency and coherence metrics, and what probably enabled the model to have comparable scores in those metrics was the pre-training phase. Furthermore, the relevance and consistency scores of reference summaries were above average, but not perfect, indicating the need for better-curated datasets and standardized practices in manual summary writing in the Greek Court judgments domain.

The extractive summaries performed relatively well, as the relevance score was above average and close to the abstractive reference summary's relevance score. However, a straight-forward comparison between those methods is not sensible as extractive summaries are very different from abstractive summaries. The vanilla LexRank with the tf-idf similarity function and the biased LexRank had no statistical important difference in terms of the relevance score.

Overall, the extractive methods exhibited average performance, while the abstractive methods produced text that was moderately fluent and coherent. However, the abstractive methods received low scores in terms of the relevance and consistency metrics. This may suggest that legal professionals favor methods that are factually aligned with the judgment text. Since factual accuracy is of utmost importance, in the legal domain, legal professionals need to rely on summaries that provide an accurate and complete representation of the original text. Therefore, summarization methods that prioritize factual accuracy and align with the original text are more likely to be preferred by legal professionals.

5.3. Correlation of Human Evaluation and Automatic Metrics

In order to assess the performance of the ROUGE automatic metrics in evaluating summaries, we measured the Pearson correlation of each ROUGE metric score with each human evaluation metric score and present the results in Table 6.

This analysis can serve as a way of finding which ROUGE metrics can substitute which human evaluation metrics when the latter are not easily available. We averaged the fluency and coherence metrics, creating a metric for internal readability. Similarly, we averaged the relevance and the consistency metrics, creating a metric for external summary factuality and relevance.

Table 6. Pearson's correlation of human evaluation scores and ROUGE metrics' F1-scores of automatically generated abstractive summaries. For each human evaluation metric, the most-correlated automatic metric is highlighted in **bold**, while the less-correlated is underlined.

Metrics	Relevance	Fluency	Coherence	Consistency	Internal	External	Average
ROUGE-1	0.1621	**0.5001**	**0.1632**	0.0929	**0.3899**	0.1335	0.2995
ROUGE-2	0.3608	0.4718	0.1224	0.2707	0.3496	0.3292	**0.4153**
ROUGE-3	**0.3793**	0.2916	−0.0738	**0.2864**	0.1323	**0.3436**	0.3105
ROUGE-L	0.1303	0.3546	-0.1155	0.2332	0.1465	0.1870	0.2084
ROUGE-W	−0.1273	0.1909	−0.2944	0.1996	−0.0513	0.1685	0.0909

We found that the internal readability metrics fluency and coherence showed moderate and low, respectively, correlation with the ROUGE metrics. Specifically, fluency had moderate correlation with all ROUGE metrics, except for ROUGE-W, where the correlation was positive, but low. With the exception of ROUGE-1, this was expected, as those metrics measure large lexical overlap with large (common) sequences of words, and thus, a summary that scores high on those metrics is expected to have fluent and coherent sentences. Coherence had low, but positive correlation with ROUGE-1 and ROUGE-2, indicating the need for metrics that capture better a summary's coherence.

The external metrics showed moderate correlation with the ROUGE metrics, which was expected as the relevance and consistency are not properties of a summary that can be easily measured by lexical overlaps between a candidate and a reference summary. In both cases, the ROUGE-3 metric seemed to have a higher correlation. However, we note the need to develop new metrics for Court judgment text summarization that correlate better with human judgment in terms of the text's relevance and consistency.

We note that the existence of metrics, such as ROUGE-W, seemed to offer little in terms of human evaluation prediction capacity, as they showed very small positive or even negative correlations with the human evaluation.

5.3.1. Human Evaluators' Highlights Analysis

In order to further assess our extractive methods, we analyzed the highlights that the human evaluators extracted from each text.

In Table 7, we present the automatic evaluation scores of both the original highlight summaries and the highlight summaries truncated to three-times the length of the reference summary, similar to the extractive summaries that were generated by our methods for the automatic evaluation (Section 5.1.2).

Table 7. Average length statistics and ROUGE F1-scores of the extractive summaries generated using the human evaluators' highlights and the automatically generated extractive summaries versus the reference abstractive summaries. The second row represents the evaluators' highlights summaries truncated to three-times the length of the reference summary, matching the extractive summaries in Table 4. The last two columns present the token-level length statistics of the summaries compared to the Court's judgment main text and reference summary, respectively.

Human/Auto Summaries	R-1	R-2	R-3	R-L	R-W	Sum/Doc	Sum/Ref
Eval.Highlights	64.56	40.72	23.21	13.93	6.56	0.170	6.43
Eval.Highlights (capped)	69.44	42.17	24.09	14.93	7.14	0.088	2.54
$\text{LexRank}_{\text{tf-idf}}$	61.45	36.90	19.52	12.15	6.41	0.079	3.00
Biased LexRank	55.57	35.41	18.68	14.58	7.63	0.079	3.00

We note that the evaluators summaries were, on average, 6.4-times the size of the reference summary, which is significantly larger than the size constraint of 3.0-times the reference summary we set for our extractive summarizers. This finding may indicate that legal experts prefer longer extractive summaries.

In terms of the ROUGE score, the evaluators' highlights summaries scored higher than our extractive summarization methods. Considering the mediocre human metric scores of our extractive methods, the ROUGE scores seemed able to capture the quality of an extractive summary as they assigned a higher score to a legal expert summary and to a automatically generated extractive summary that legal experts rated as mediocre.

Considering the mediocre human metric scores of our extractive methods and the fact that the legal-expert-generated extractive summaries scored higher in terms of the ROUGE metrics, than automatically generated extractive summaries, we can conclude that the ROUGE metric can be useful in assessing an extractive summary's quality. However, we note that, when the ROUGE metric scores are close, as is the case for LexRank and Biased LexRank, the ROUGE metrics may not align with human judgment. In our human evaluation survey, the extractive methods have similar relevance scores (Table 5), while having small, but noticeable differences in terms of the ROUGE score (Table 4).

We also compared, using the ROUGE metrics, the human-generated highlight summaries with the automatically extracted summaries, considering the first to be reference extractive summaries. In Table 8, we present the results.

We found that the vanilla LexRank method clearly outperformed the Biased LexRank method. However, taking into consideration that the legal experts assigned similar relevance scores to those methods (Table 5), we note that small or even moderate differences in terms of the ROUGE score did not necessarily translate to differences in terms of human judgment. This may be explained by the fact that extractive summarization is an under-constrained task and extractive summaries showed great lexical overlap with the reference summary, thus having high ROUGE scores may not be the only type of summaries that perform well in terms of human judgment.

Table 8. ROUGE metrics' comparison of automatic extractive summarization methods using the human evaluators' highlights summaries as a reference. We report the average ROUGE F1-score, over all evaluators and all Court judgment summaries, in our human evaluation study.

Auto Summaries	R-1	R-2	R-3	R-L	R-W
$\text{LexRank}_{\text{tf-idf}}$	80.24	46.66	24.86	16.14	7.26
Biased LexRank	74.23	42.49	21.81	16.84	8.00

5.3.2. Human Evaluators' Agreement

The inter-evaluator agreement is crucial in ensuring the reliability and validity of human evaluation metrics. Krippendorff's alpha metric [41] for interval variables is a useful tool for measuring the level of agreement among evaluators. Measuring the agreement can be helpful as a way of quantifying which human metrics are well-defined, and thus, human evaluators give similar scores. It can also be used to find human metrics that are ambiguously defined or naturally more subjective, and thus, inter-evaluator agreement is low. Those results may be used to inform our interpretation of human evaluator metric results downstream. Furthermore, low inter-evaluator agreement results can lead to more thorough metric definitions for the low-agreement metrics. In Table 9, we present the results.

We found that the human evaluators systematically agreed on the external metrics relevance and consistency. Their evaluations seemed to be more unreliable in terms of the internal metrics, especially fluency. Our findings were similar to [27] and indicated that the task of evaluating the summary's inclusion or not of all the relevant information from the main text, as well as its factual consistency with it is more objective than the task of evaluating a summary's fluency and inner coherence, which can be more subjective due to differences in personal reading/writing style.

In order to measure the agreement on the highlights each evaluator had extracted from the Court's decision text, we calculated for each text the average pairwise highlight agreement between each pair of evaluators. Let N_{evals} be the number of evaluators and $H^{(i)}$

the set of the N_{evals} highlight sets collected for text i, then the average highlight pairwise agreement score for the $i-th$ text is given by

$$H_{\text{avg}}^{(i)} = \frac{1}{N_{\text{evals}}(N_{\text{evals}}-1)} \sum_{H^{(i)}} \sum_{H^{(j)} \neq H^{(i)}} ||H^{(i)} \cap H^{(j)}|| / ||H^{(i)} \cup H^{(j)}|| \quad (17)$$

The results in Table 10 show large differences of highlighting style between the evaluators. Furthermore, there was large variance in the highlights agreement between each question, which may be attributed to the different highlighting style of each evaluator and also qualitative differences in the texts. This result further supported the position that extractive summarization is an under-constrained task, as each evaluator had a different approach in generating an extractive summary. This remark, however, as we saw in Table 9, does not imply that human evaluation of extractive summaries is under-constrained, as manually generating an extractive summary is quite different from evaluating an automatically constructed one.

Table 9. Krippendorff's alpha agreement metric on each human evaluation metric for each summary type. The internal metric is the average of fluency and coherence metrics. The external metric is the average of relevance and consistency metrics. In the abstractive summaries category, we included both reference and generated abstractive summaries as human evaluators were evaluated both in the same way and in a randomized order without knowing if any of the summaries were written by legal experts.

Type	Relevance	Fluency	Coherence	Consistency	Internal	External	Average
Abstractive	0.6405	−0.0215	0.0709	0.6400	0.0260	0.6754	0.4332
Extractive	0.4250	-	-	-	-	-	-

Table 10. Average pairwise highlights agreement on each question over all human evaluators. The pairwise agreement is calculated as the ratio between the intersection and the union of the two sets of highlights.

q1	q2	q3	q4	q5	Average
0.2619	0.0908	0.1531	0.2556	0.3957	0.2314 ± 0.141

6. Conclusions and Future Work

Automated text summarization systems are needed in the legal domain to assist law practitioners, judges, and scholars in searching for relevant statutes and case laws. Summarizing legal texts is a challenging task as the texts are often lengthy and contain legal terminology. In this paper, we studied the problem of automated summarization of Greek legal documents. Currently, there is no available dataset in the Greek language that is designed for this task. To address this challenge, we presented a new dataset of Greek Court judgments that has been specifically created for this task. The availability of this dataset is crucial for training and evaluating summarization models in the legal domain, which often require large amounts of labeled data. We adopted and compared the performance of several state-of-the-art methods from extractive summarization (LexRank and Biased LexRank), as well as an abstractive summarization approach using an encoder–decoder deep learning model based on the BERT architecture. We evaluated the effectiveness of these methods using both human and automatic evaluation metrics, providing a detailed comparison of various variations of extractive and abstractive summarization techniques and studied the correlation between automatic and human evaluation metrics. Although this study makes an initial contribution to the field of automated summarization of Greek legal documents, further research is required to advance our understanding in this area. Specifically, future studies should focus on creating better datasets, improving the evaluation metrics, and exploring advanced techniques such as deep learning.

In future work, we plan to further study how various different neural network architectures can benefit the generated summaries' quality. Different attention mechanisms, which reduce the quadratic complexity of the original self-attention layer, such as the Longformer [42] and the Reformer [43] architectures, may enhance the quality of the generated summaries. Furthermore, we aim to incorporate negative summary samples during training with contrastive learning, as in [44], to prevent our model from deviating/being unfaithful to the input text. Last but not least, we intend to utilize hierarchical Transformer networks, which can bypass the Transformer's quadratic complexity, by first generating segment-level representations and, afterwards, merging them into a document-level representation. The document-level representations can be constructed either naively by concatenation or averaging or by further Transformer transformation, as in [45].

Author Contributions: M.K.: conceptualization, methodology, original draft preparation, review and editing; D.G.: methodology, software, validation, data curation, original draft preparation, review and editing; E.G.: supervision, review and editing; P.T.: supervision, funding acquisition, review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: We gratefully acknowledge the support of the European-Union-funded Project PolicyCLOUD under Grant Agreement No. 870675.

Data Availability Statement: The dataset presented in this study is openly available at https://huggingface.co/datasets/DominusTea/GreekLegalSum, accessed on 16 April 2023. The code, materials, baseline, and best-performing models presented in this study are openly available at https://github.com/DominusTea/LegalSumPaper, accessed on 16 April 2023.

Acknowledgments: We express our gratitude to the legal experts who participated in the evaluation for their valuable time and expertise. We would also like to thank the Supreme Civil and Criminal Court of Greece for providing the texts of their decisions, as well as their notation, indexing, and summaries.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of the data; in the writing of the manuscript; nor in the decision to publish the results.

Appendix A. Web Application

For the purposes of disseminating the data collected for our Court judgment dataset and conducting a human evaluation study on automatic summarization methods for Court judgments, we developed a web application that has built-in support for automatic building, testing, and deployment. Next, we describe the technology stack of the frameworks we utilized for developing our web application.

We used Angular as our main web application development framework for Typescript. In order to develop responsive front-end interfaces, we utilized the Bootstrap CSS. The server-side application is a complete Spring application, using Spring to create a REST MVC application, utilizing Spring Security for authentication and access-control and Spring Data JPA for the JPA-based data access layers framework. We used the SurveyJS framework in order to generate and display our dynamic survey for each participant in the study. In our case, the survey was dynamically and independently generated for each participant in the study, inserting the questions that correspond to that particular participant and localizing the study to the participant's web client's currently selected language. After the survey was completed, each participant's answers were stored in the database.

Figure A1 provides a high-level overview of the utilized data model. Each survey entity is uniquely identified with the user it corresponds to. Each survey entity is related to multiple BERT or LexRank summary ratings, with each BERT/LexRank summary rating entity being related—using a many-to-one relationship—to a BERT summary or a LexRank summary, respectively. Each summary entity is related many-to-one to a case-law entity, which corresponds to a judicial judgment text, its corresponding reference summary, as well as metadata such as the classification tags corresponding to the judgment. Our database

schema allows for surveys independently customized to each legal expert, by including different judgments and summaries to be evaluated.

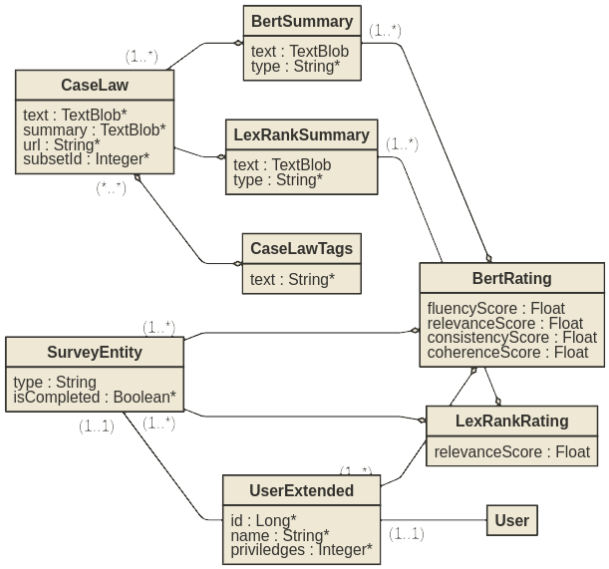


Figure A1. A schema of the entity fields and the cross-entity relationships in our database.

Having defined the database model, we generated a RESTful API that corresponds to CRUD operations on the database entities. We exposed the API to the users by generating a web interface that consists of the API complete documentation, as well as an interface for sending and displaying the results of an API request. Figure A2 includes a screen capture of our API's documentation interface, and Figure A3 presents the graphical interface of sending an API request through our documentation's interface page.

Figure A2. Our REST API's documentation interface. The screen capture displays the interface for two entities of our database.

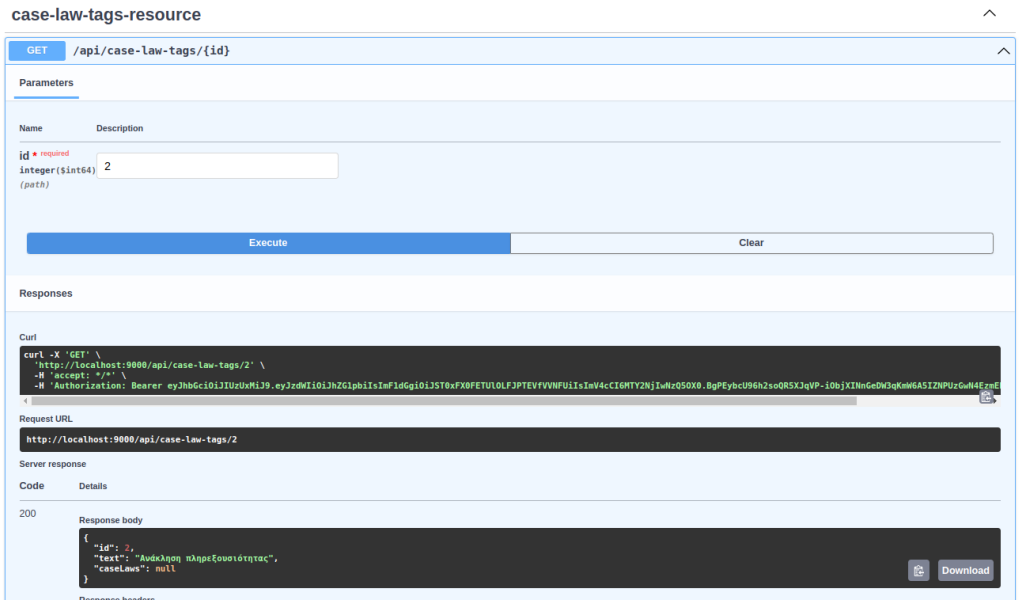


Figure A3. Screen capture of sending a REST API request and displaying its result through our documentation page's interface.

References

1. Nenkova, A. Automatic Summarization. *Found. Trends® Inf. Retr.* **2011**, *5*, 103–233. [CrossRef]
2. Jain, D.; Borah, M.D.; Biswas, A. Summarization of legal documents: Where are we now and the way forward. *Comput. Sci. Rev.* **2021**, *40*, 100388. [CrossRef]
3. Solan, L.M. *The Language of Judges*; University of Chicago Press: Chicago, IL, USA, 1993. [CrossRef]
4. Turtle, H. Text retrieval in the legal world. *Artif. Intell. Law* **1995**, *3*, 5–54. [CrossRef]
5. Ermakova, L.; Cossu, J.V.; Mothe, J. A survey on evaluation of summarization methods. *Inf. Process. Manag.* **2019**, *56*, 1794–1814. [CrossRef]
6. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the Text Summarization Branches Out*; Association for Computational Linguistics: Barcelona, Spain, 2004; pp. 74–81.
7. Ganesan, K. ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks. *arXiv* **2018**, arXiv:1803.01937.
8. 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, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318. [CrossRef]
9. Tianyi, Z.; Varsha, K.; Felix, W.; Kilian Q., W.; Yoav, A. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
10. Kornilova, A.; Eidelman, V. BillSum: A Corpus for Automatic Summarization of US Legislation. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, Hong Kong, China, 4 November 2019; pp. 48–56. [CrossRef]
11. Bhattacharya, P.; Hiware, K.; Rajgaria, S.; Pochhi, N.; Ghosh, K.; Ghosh, S. A Comparative Study of Summarization Algorithms Applied to Legal Case Judgments. In Proceedings of the Advances in Information Retrieval, Cologne, Germany, 14–18 April 2019; Lecture Notes in Computer Science; Azzopardi, L., Stein, B., Fuhr, N., Mayr, P., Hauff, C., Hiemstra, D., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 413–428. [CrossRef]
12. Shen, Z.; Lo, K.; Yu, L.; Dahlberg, N.; Schlanger, M.; Downey, D. Multi-LexSum: Real-World Summaries of Civil Rights Lawsuits at Multiple Granularities. *arXiv* **2022**, arXiv:2206.10883.
13. Xu, H.; Šavelka, J.; Ashley, K.D. Using Argument Mining for Legal Text Summarization. In Proceedings of the JURIX, Brno, Czech Republic, 9–11 December 2020; Volume 334, pp. 184–193. [CrossRef]
14. Zhong, L.; Zhong, Z.; Zhao, Z.; Wang, S.; Ashley, K.D.; Grabmair, M. Automatic Summarization of Legal Decisions using Iterative Masking of Predictive Sentences. In Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law (ICAIL'19), Montreal, QC, Canada, 17–21 June 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 163–172. [CrossRef]
15. de Vargas Feijó, D.; Moreira, V.P. RulingBR: A Summarization Dataset for Legal Texts. In *Lecture Notes in Computer Science*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 255–264. [CrossRef]
16. Kryscinski, W.; Keskar, N.S.; McCann, B.; Xiong, C.; Socher, R. Neural Text Summarization: A Critical Evaluation. 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), Hong Kong, China, 3–7 November 2019; pp. 540–551. [CrossRef]

17. Grusky, M.; Naaman, M.; Artzi, Y. Newsroom: A Dataset of 1.3 Million Summaries with Diverse Extractive Strategies. *arXiv* **2020**, arXiv:1804.11283.
18. Anand, D.; Wagh, R. Effective deep learning approaches for summarization of legal texts. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 2141–2150. [CrossRef]
19. Xu, H.; Savelka, J.; Ashley, K.D. Toward summarizing case decisions via extracting argument issues, reasons, and conclusions. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, Sao Paulo, Brazil, 21–25 June 2021; Association for Computing Machinery: New York, NY, USA, 2021; pp. 250–254.
20. Papantoniou, K.; Tzitzikas, Y. NLP for the Greek Language: A Brief Survey. In Proceedings of the 11th Hellenic Conference on Artificial Intelligence, ACM, Athens, Greece, 2–4 September 2020. [CrossRef]
21. El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* **2021**, *165*, 113679. [CrossRef]
22. Farzindar, A.; Lapalme, G. Letsum, an automatic legal text summarizing system. In *JURIX 2004, the Seventeenth Annual Conference*; IOS Press: Amsterdam, The Netherlands, 2004; pp. 11–18.
23. Polsley, S.; Jhunjhunwala, P.; Huang, R. CaseSummarizer: A System for Automated Summarization of Legal Texts. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, Osaka, Japan, 11–16 December 2016; The COLING 2016 Organizing Committee: Osaka, Japan, 2016; pp. 258–262.
24. Kim, M.Y.; Xu, Y.; Goebel, R. Summarization of Legal Texts with High Cohesion and Automatic Compression Rate. In Proceedings of the New Frontiers in Artificial Intelligence, Kanagawa, Japan, 27–28 October 2013; Lecture Notes in Computer Science; Motomura, Y., Butler, A., Bekki, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 190–204. [CrossRef]
25. Manor, L.; Li, J.J. Plain English Summarization of Contracts. *arXiv* **2019**, arXiv:1906.00424.
26. Feijo, D.; Moreira, V. Summarizing Legal Rulings: Comparative Experiments. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), Varna, Bulgaria, 2–4 September 2019; INCOMA Ltd.: Varna, Bulgaria, 2019; pp. 313–322. [CrossRef]
27. Fabbri, A.R.; Kryściński, W.; McCann, B.; Xiong, C.; Socher, R.; Radev, D. SummEval: Re-evaluating Summarization Evaluation. *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 391–409. [CrossRef]
28. Maynez, J.; Narayan, S.; Bohnet, B.; McDonald, R. On Faithfulness and Factuality in Abstractive Summarization. *arXiv* **2020**, arXiv:2005.00661.
29. Karampatzos, A.; Malos, G. The Role of Case Law and the Prospective Overruling in the Greek Legal System. In *Ius Comparatum-Global Studies in Comparative Law*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 163–184. [CrossRef]
30. Dodge, J.; Ilharco, G.; Schwartz, R.; Farhadi, A.; Hajishirzi, H.; Smith, N. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv* **2020**, arXiv:2002.06305.
31. Luijtgaarden, N.V.D. Automatic Summarization of Legal Text. Master's Thesis, Utrecht University, Utrecht, The Netherlands, 2019. Available online: https://studenttheses.uu.nl/handle/20.500.12932/34261 (accessed on 16 April 2023).
32. Erkan, G.; Radev, D.R. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *J. Artif. Intell. Res.* **2004**, *22*, 457–479. [CrossRef]
33. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*; Technical Report 1999-66; Stanford InfoLab: Stanford, CA, USA, 1999.
34. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 404–411.
35. Otterbacher, J.; Erkan, G.; Radev, D.R. Biased LexRank: Passage retrieval using random walks with question-based priors. *Inf. Process. Manag.* **2009**, *45*, 42–54. [CrossRef]
36. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [CrossRef]
37. Liu, Y.; Lapata, M. Text Summarization with Pretrained Encoders. 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), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 3730–3740. [CrossRef]
38. Koutsikakis, J.; Chalkidis, I.; Malakasiotis, P.; Androutsopoulos, I. GREEK-BERT: The Greeks Visiting Sesame Street. In Proceedings of the 11th Hellenic Conference on Artificial Intelligence (SETN 2020), Athens, Greece, 2–4 September 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 110–117. [CrossRef]
39. Allahyari, M.; Pouriyeh, S.; Assefi, M.; Safaei, S.; Trippe, E.D.; Gutierrez, J.B.; Kochut, K. Text Summarization Techniques: A Brief Survey. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*. [CrossRef]
40. Gambhir, M.; Gupta, V. Recent automatic text summarization techniques: A survey. *Artif. Intell. Rev.* **2017**, *47*, 1–66. [CrossRef]
41. Krippendorff, K. Computing Krippendorff's Alpha-Reliability. 2011. Available online: https://repository.upenn.edu/asc_papers/43 (accessed on 16 April 2023).
42. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *arXiv* **2020**, arXiv:2004.05150.
43. Kitaev, N.; Kaiser, L.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:2001.04451v2.

44. Liu, W.; Wu, H.; Mu, W.; Li, Z.; Chen, T.; Nie, D. CO2Sum: Contrastive Learning for Factual-Consistent Abstractive Summarization. *arXiv* **2021**, arXiv:2112.01147.
45. Zhu, C.; Xu, R.; Zeng, M.; Huang, X. A Hierarchical Network for Abstractive Meeting Summarization with Cross-Domain Pretraining. *arXiv* **2020**, arXiv:2004.02016.