Step Up Your Game: Summarizing Long Regulatory Documents Using a Two- or Multi-Step Method

MIKA SIE, Author, Utrecht University, The Netherlands
ALBERT GATT, First supervisor, Utrecht University, The Netherlands
SJAAK BRINKKEMPER, Second supervisor, Utrecht University, The Netherlands

ABSTRACT

Automatic Text Summarization (ATS) is the process of automatically summarizing a text. Advancements in neural models in Natural Language Processing (NLP) have significantly enhanced summarization capabilities, making it a crucial tool for processing extensive regulatory documents. Long regulatory texts are challenging to summarize due to their length and complexity. To address this, a two- and multi-step extractive-abstractive architecture is proposed to handle lengthy regulatory documents more effectively. This research shows that the effectiveness of a two-step architecture for summarizing long regulatory texts varies significantly depending on the model used. Specifically, the two-step architecture improves the performance of decoder-only models. For abstractive encoder-decoder models with short context lengths, the effectiveness of an extractive step varies, whereas for long context encoder-decoder models, the extractive step worsens their performance. This research also highlights the challenges of evaluating generated texts, as evidenced by the differing results from human and automated evaluations. Most notably, human evaluations favoured legal language models, while automated metrics preferred general language models. The results underscore the importance of selecting the appropriate summarization strategy based on model architecture and context length. Broadly, this research contributes to the development of more efficient and accurate tools for summarizing complex regulatory documents, enhancing accessibility, and aiding in compliance and decision-making processes in dynamic industries such as the green molecule sector. All code and models are available on GitHub and HuggingFace.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Prof. Dr. Albert Gatt for supervision and help during my thesis. Albert's intricate understanding and knowledge were invaluable to the completion of this project. But most importantly, Albert's enthusiasm and curiosity for the NLP field inspired and motivated me. I thoroughly enjoyed our weekly one-on-one sessions.

I would also like to thank Prof. Dr. Sjaak Brinkkemper for agreeing to be my second supervisor and for his feedback during the first phase of the thesis project.

A big thanks goes to Ruby, Michiel, and Andy for the guidance and help as supervisors of Power2X. The amount of time and effort that Ruby and Michiel have spent guiding me during my thesis is enormous. I would like to thank them for explaining new green molecule topics to me, for listening endlessly to difficult AI topics of which they had no prior knowledge, and for the enjoyable weekly meetings we had.

I am grateful to Power2X for the opportunity to write my thesis during this internship. Besides working on my thesis, I was able to learn a lot on a whole host of new topics that I wasn't previously familiar with. I would like to thank all my colleagues for the coffees, lunches, interesting conversations, post-lunch walks, vrijmibo's, junior-weekends and the fun we had. Your personalities left a lasting impression on me.

A big thanks to Sjors for helping me out with a lot of my technical issues. For every message that was sent out of the blue, Sjors was happy to help me out and send me on my way with all the issues I encountered on GPU platforms.

Thanks to all my friends for supporting me during my thesis. Their joy and friendship made the thesis project feel much more enjoyable and rewarding.

I would like to thank my family for their unconditional support during the entirety of my studies. Without all of you, I wouldn't have been where I am today. I thank you all deeply.

Last but not least, I want to thank my wonderful girlfriend Sofie. Sofie's support and pride have been with me every step of the way. Thank you for all the laughter we share as it made this thesis project feel like a breeze.

1 INTRODUCTION

Text summarization is the task that involves generating a compressed, concise, and fluent version of an input text while preserving its main key points. A summary of a text proves useful because it helps people process and understand texts faster and better. Summarizing regulatory texts is important for making complex legal language more accessible and ensuring compliance by condensing information into a concise, understandable format. It facilitates efficient decision-making, risk management, and communication across stakeholders. This is especially relevant in dynamic industries, such as the green molecule industry, where staying informed about evolving regulations is essential. But the summary must be of high quality, factually correct and that doesn't leave out important information. Otherwise, inaccurate or incomplete summaries could result in the wrong information and consequently wrong actions.

A regulatory text is a formal document issued by a government or regulatory body that outlines rules, guidelines, or standards intended to govern the conduct, practices, or operations within a specific industry, sector, or jurisdiction. These texts serve to establish and enforce compliance with legal and regulatory requirements. Regulatory documents are difficult to process due to their extensive size, unique structure, numerous citations and references, ambiguity, and domain-specific vocabulary. These characteristics cause manual summarization of regulatory texts to be a time-consuming and challenging task. Summaries can also contain human bias because of the knowledge and opinions of the person summarizing the text.

Automatic Text Summarization (ATS) has been developed to replace the manual process of text summarization. Early ATS methods showed some positive results but they were still subpar to manual summarizations. Advancements in neural models and deep learning have improved performance on ATS. Notably, the adaptation of Large Language Models (LLMs) has significantly enhanced summarization results. Current automatic summarization tools face challenges with regulatory texts because their length exceeds the context length of LLMs. When the context length is surpassed, the model tends to prioritize recent information, leading to a bias that excludes early information from the document in the summary. Leaving out important information from a regulatory or legal document in the summary could have major negative effects. As summarizing regulatory texts has shown to be valuable, it can be of great importance to investigate methods to adapt current summarization tools such that longer texts can be handled. This paper proposes a two- and multi-step method that is used to summarize long regulatory documents. It also compares the effectiveness of different neural model architectures and combinations for the summarization task.

Current ATS methods use either extractive or abstractive summarization. Extractive summarization identifies salient words or sentences that are directly extracted and combined to form the final summary. Abstractive summarization generates a summary based on the understanding of the words and sentences from the source document, capturing salient information of the document. Both summarization types have been visualised in Figure 1. An advantage of extractive summarization is that it captures sentences and information literally, resulting in a factually consistent summary. However, the summary is harder to read and feels less intuitive as sentences are copied and combined. Abstractive summaries are more coherent and fluent as they summarize texts in a human-like fashion. But it also has disadvantages because an intricate understanding of the original text is required and the summary can be factually inconsistent.

This research proposes an extractive-abstractive architecture that utilizes the strengths of both techniques and mitigates their weaknesses to summarize long, regulatory documents. First, the document will be segmented into smaller, more manageable units called chunks. This process is therefore called chunking. Each chunk is then processed by an extractive summarization model and all resulting summaries are concatenated. After this step, it could be required to perform another extractive summarization step to compress the document even more. This new text is then summarized in an abstractive manner, creating a final summary. Combining these two summarization steps could prove useful to handle the large size of the original text. It uses extracted salient

sentences to create a coherent, fluent summary. The process is visualized in Figure 2. Similar architectures have been used on different types of texts and they have shown promising results. However, summarizing long regulatory documents using this architecture has been researched less extensively. Thus, this paper researches the use of a two- or multi-step architecture to summarize long, regulatory documents. This research also evaluates various models used for each step to identify the most effective combination of models for the summarization task.

Research questions. The paper is structured around a main research question (RQ) and several research sub-questions (RSQ):

RQ Does a two- or multi-step extractive-abstractive architecture summarize long, regulatory documents better than not using any extractive steps
RSQ1 Does a two-step or multi-step architecture perform better?
RSQ2 Does the use of a domain-specific legal language model for the extractive summarization step produce better results compared to when no domain-specific language model is used?
RSQ3 Does a long context length in the extractive summarization step provide better results?
RSQ4 Does a long context length in the abstractive summarization step provide better results?

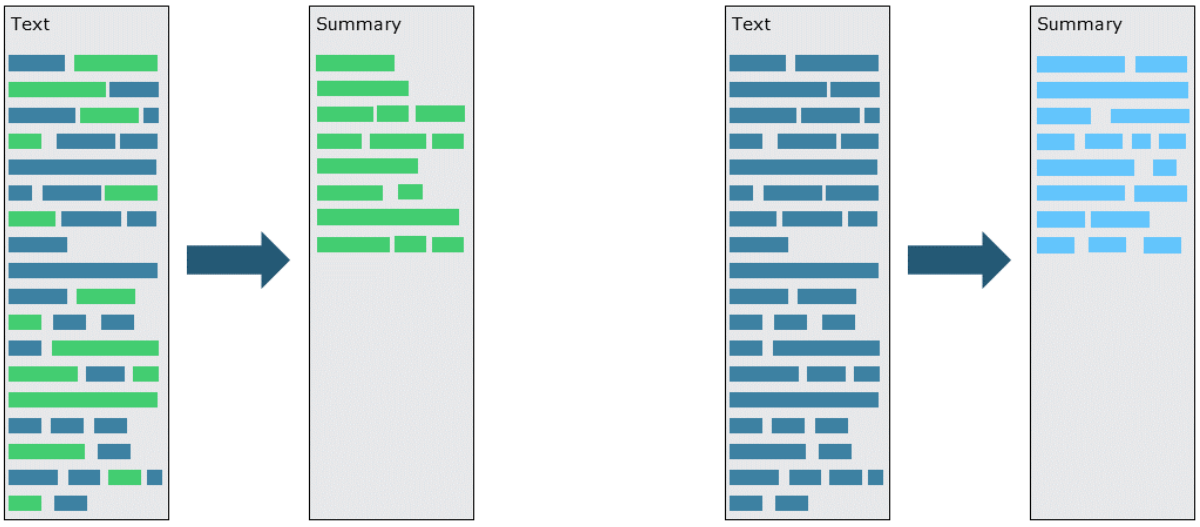
Role of Power2X. The thesis will be written at Power2X, a company specializing in project development and consulting in the renewable energy sector. The regulatory team at Power2X has to analyze and summarize long documents manually to obtain knowledge from them. It is important to have a deep understanding of regulatory documents because they have a great effect on Power2X's strategy and future work. Currently, documents are processed manually through multiple steps, referred to as the 'knowledge discovery process':

(1) A rough summary is created by extracting important information from the document.
(2) Based on this information, a more refined summary is created which can be distributed to the rest of the company as an information source. This summary is easy to read and is based on certain viewpoints that are important for Power2X.
(3) The regulation's implications on Power2X's business are identified to be able to strategize and anticipate accordingly.

The documents are very long, often containing over 50 pages. They contain difficult, legal text and a lot of cross-references. This process is time-consuming and requires a lot of knowledge and effort. Summaries can also contain human bias because of the knowledge and opinion of the person summarizing the text. Automatically summarizing the document can reduce the time taken to process the document and remove human bias. This research can improve the first step of the knowledge discovery process and to some extent the second step.

Power2X contributes to this research in several ways. Power2X's regulatory team gives insights into regulatory documents by explaining how documents are processed. The regulatory team of Power2X also helps with human evaluation for preliminary results. Furthermore, Power2X offers guidance on writing and researching during the thesis project.

Structure. This paper is structured as follows. Section 2 describes related work in the field of automatic text summarization. Section 3 explains the methods and working of the models used in this study. Section 4 reports the results and findings. In Section 5, the research questions are answered. Additionally, the ethical implications, limitations, and potential future work of this study are discussed. Finally, in Section 6, the research is summarized and concluded.



(a) Extractive summarization.

(b) Abstractive summarization.

Fig. 1. The two types of summarization techniques. Dark blue lines represent the original text from the source document. Green lines represent text that has been identified as salient sentences which are then extracted to form an extractive summary. Light blue lines represent a new text that has been rewritten using the original text from the source document.

2 RELATED WORK

Supervised learning and unsupervised learning can both be employed for ATS. With the development of large-scale summarization datasets, supervised learning for ATS is frequently used for extractive summarization ([24], [94], [95], [132]) and abstractive summarization ([118], [134], [23]). Unsupervised learning offers benefits for ATS because it eliminates the need for an annotated dataset and it is more general for various ATS situations. Unsupervised learning can be implemented for extractive summarization([35], [134], [78]) and abstractive summarization ([121], [74], [100]). In this study, the focus lies on supervised learning because of the availability of a dataset containing golden-reference summaries, making it feasible to train a summarization model. Consequently, every section is written with a focus on supervised learning. We refer to Khosravani et al. [68] for a more detailed overview of unsupervised ATS.

This section offers a look at related work in the field of ATS. This section is structured as follows. Section 2.1 explains the techniques of extractive and abstractive summarization. It also informs how these techniques functioned in earlier stages. Section 2.2 explains the introduction and implementation of neural models in ATS. Section 2.3 describes methods used to summarize long documents that differ from the methods used in this research. Section 2.4 discusses papers that use two- or multi-step methods to summarize long documents. Section 2.5 gives an overview of research done on the summarization of regulatory and legal texts. Section 2.6 describes different datasets used for ATS. Section 2.7 discusses various evaluation metrics for automatically generated summaries.

2.1 Early extractive and abstractive summarization

ATS is a component of the broader field of Natural Language Processing (NLP). NLP encompasses two key components: Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU involves

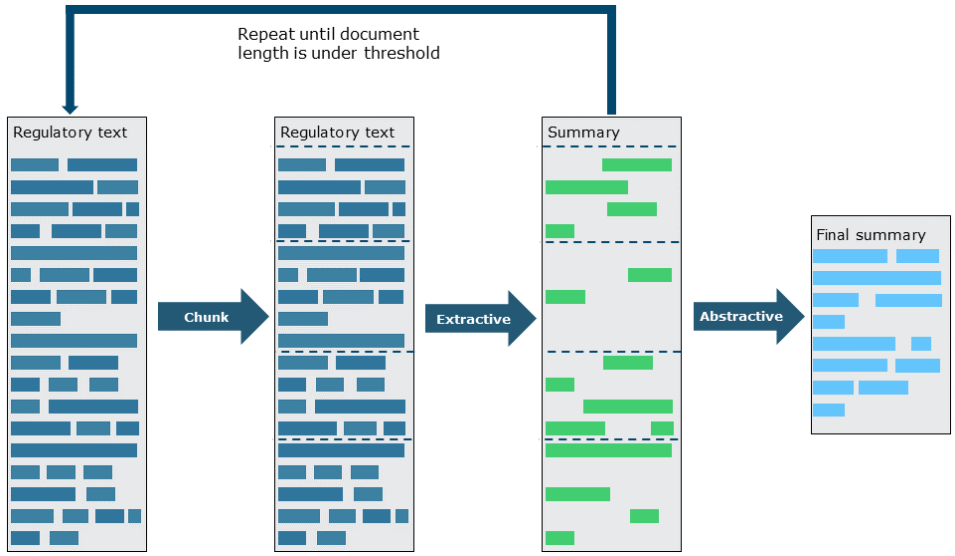


Fig. 2. Summarization process proposed by this research. Dotted lines indicate the borders of the chunks.

the comprehension of human language by computers or machines. NLG focuses on the creation of human-like text by computers or machines. ATS is concerned with both NLU and NLG because the source text needs to be understood first before a corresponding summary can be generated. The earliest research done on ATS started in the late 1950's by Luhn. [85]. Luhn proposed a technique to automatically summarize academic and technical papers. Luhn's method for ATS involves identifying and extracting key sentences from a document based on the frequency of important terms. Luhn introduced a concept known as "significant words" to determine the importance of terms, considering their occurrence in the document relative to a predefined set of keywords. The algorithm selects sentences containing these significant words, creating a concise summary that captures the essential information of the original text. Luhn states that the creation of summaries requires intellectual effort and knowledge. He also argues that achieving consistency and objectivity is challenging because the background and opinions of the person summarizing the text differ. Luhn proposes that automatizing this summarization process will remove human effort and human bias.

ATS can be divided into two categories; extractive and abstractive summarization. Extractive summarization can be defined as the process of identifying salient words and sentences which are then extracted and grouped to create the summary. Abstractive summarization can be defined as the process of identifying salient information, based on a Language Model's understanding of language, and paraphrasing this information to create the final summary. Both types of summarization are visualised in Figure 1. Extractive summarization is more likely to be factually consistent because sentences are copied from the source text. This is because the meaning of a specific sentence is not altered. However, the selection process of sentences heavily influences the factual consistency of the final summary. If an important sentence is not extracted, then the summary could have a very different meaning.

Abstractive summaries are however more intuitive to read because the text is paraphrased from the source document. Extractive summaries may lack fluency and coherency as the extracted sentences are directly copied and concatenated. But to be able to paraphrase texts and create a new text and summary, an intricate understanding of natural language is required [44], making abstractive summarization more difficult[116]. However, NLG models

can generate text that is nonsensical, or unfaithful to the source text [64]. This undesirable text is known as hallucination. Abstractive summarization models can suffer from hallucination resulting in an inconsistent and incorrect summary [88] [73] [13]. The following two paragraphs will discuss the earlier methods of extractive and abstractive summarization models before the introduction of neural models to ATS.

2.1.1 Extractive summarization methods. The earliest adaptions of ATS were mostly focused on extractive summarization. Extractive methods rely less on actual understanding of the text and can usually be implemented by selecting sentences that score highest on some criteria [43]. Extractive summarization is typically performed in three steps [96]:

(1) Create a representation of the original text.
(2) Score all sentences, where the score reflects the importance of a sentence.
(3) Select the highest-scoring sentences to be part of the summary.

One of the first known versions of ATS also relied on this principle [85]. There are many methods and approaches for extractive summarization [44] [37] [34]:

- Statistical-based methods.
- Graph-based methods.
- Fuzzy-based methods.
- Semantic-based methods.
- Machine Learning-based methods.

Statistical-based methods extract important sentences and words from the source text, based on a statistical analysis of a set of features. They require little processing power and memory capacity. The biggest disadvantage is that summaries are often low-quality because they fail to consider semantics.

Graph-based methods are widely used methods for extractive summarization [34]. Words or sentences are shown in graphs as nodes. Edges between the nodes connect sentences based on semantic relevance, using for example the cosine similarity as done by Erkan et al. [38]. Afterwards, a ranking algorithm finds the most important sentences which are then extracted. LexRank [38] is an unsupervised learning approach that was introduced in 2004 and has been used as an important baseline for ATS. LexRank represents the document as a graph, where each sentence is a node in the graph. Sentences are represented as an N-dimensional vector using a bag-of-words model, with N representing the number of possible words. The strength of edges between nodes is determined based on the similarity of the sentences. The similarity is calculated based on the cosine similarity pair-wise between each sentence. So, a stronger edge means that two sentences are more similar. Once the representation of sentences has been set and the graph has been created, a link analysis algorithm is used that assigns a numerical importance score to each node. Sentences that have higher edge scores will receive a higher importance score. Then, the top-ranked sentences are selected to form the extractive summary. LexRank was a highly effective approach. It is often used as a benchmark for summarization tasks.

Fuzzy-based approaches are based on fuzzy logic [137]. Sentences are scored using a fuzzy logic system along with pre-defined features to give every sentence a score between 0 and 1. Then, the most important sentences are extracted to form the final summary.

Semantic-based methods are based on the usage of Latent Semantic Analysis (LSA). LSA is an unsupervised learning technique that represents semantics based on the occurrence of words observed [37]. LSA assumes that a high number of similar words implies that sentences are semantically related. LSA creates dense representations of words based on their surroundings. These representations can then be used for a multitude of tasks. The process is as follows. First, a word-to-phrase matrix is constructed. Then the correlation between terms and phrases is calculated using Singular Value Decomposition on the word-to-phrase matrix. Singular Value Decomposition reduces the dimensionality of the matrix by encoding the dense representation of the words based on their

environments. Then, a ranking algorithm identifies the most important sentences based on their connections and similarity scores. The most important sentences are extracted to form the final summary.

Machine Learning-Based methods have also been used to summarize texts. Machine learning algorithms can be supervised, unsupervised, or semi-supervised. Machine learning methods can for example make use of hidden Markov Models [30], Naive Bayes [97] and decision trees [97]. However, more machine learning algorithms can be used for this task. Machine learning methods leverage hand-crafted features such as word frequencies, sentence position, named entities, and sentence length, to find important information. Features are usually constructed at the word level or sentence level. The summarization task is often formulated as a binary classification problem, where each sentence is classified as either 'Summary' or 'Non-Summary.'

Besides the mentioned approaches there are other different approaches for extractive summarization. However, all approaches rely on the same summarization process, which relies exists out of the following three steps; create a representation of all sentences, score all sentences, and finally select the highest scoring sentences. Extractive summarization was not difficult to implement before the introduction of neural models. Older methods and algorithms could already achieve the three-step process. A lot of current systems, that do use neural models, still rely on the same three-step process.

2.1.2 Abstractive summarization methods. In earlier development of ATS, less emphasis was placed on researching abstractive summarization. Abstractive methods are highly complex because they require extensive NLP [44], more so than extractive summarization. NLU is required to thoroughly understand the meaning of a text. NLG can then be applied to the meaning representations to create a coherent summary. However during earlier development stages of ATS, NLU, and NLG still faced many hard challenges which had not been adequately addressed. NLU struggled with semantic representation and inference. NLG faced numerous challenges in document planning, aggregation, grammatical structuring, referring expressions, and more. NLG systems mainly relied on rule-based systems that were difficult to create and scale. Before the introduction of neural models, abstractive summarization methods used two approaches to understand text [93] [51]:

- Structure-based methods.
- Semantic-based methods.

Structure-based methods find the most important information from a text by using different kinds of psychological feature schemas. These schemas make use of the structure of the document. They can use trees, templates, ontology, lead and body phrases, graphs, or rules [93] [51] to find the most important information in a document. So, Structure-based methods rely on the document's structure to extract meaning from a text. They may not capture the deeper semantic meaning of the content because they do not consider the actual meaning of the text itself.

Semantic-based methods focus on the semantic meaning of a document to be able to understand the text. Semantic-based methods aim to create a semantic representation of the text via information items, predicate arguments, semantic graphs, or multi-modal semantic methods. All four methods predominantly make use of verbs, nouns, subjects, and objects in sentences. However multi-modal methods make use of other types of media such as images to be able to create a semantic representation. Semantic-based methods are less reliant on document structure as they focus on capturing the meaning of the text at a semantic level.

After having found the most vital information of a document using a structure-based method or a semantic-based method, a summary has to be generated. Sentence compression, concept fusion, and calculation of path scores for graph-based methods or NLG tools are used to create the final summary. Sentence compression aims to reduce the length of the sentence without compromising its meaning. Concept fusion tries to find similar topics across a text and fuses topic-associated sentences. Calculation of path scores can be used to find paths in graphs that are the most readable and fluent while containing enough information. Summary generation using NLG tools is mostly used because they increase fluency and decrease grammatical mistakes [51].

Compared to extractive summarization, there are fewer approaches for abstractive summarization. The mentioned abstractive approaches have various downsides which negatively impact the quality of the generated summary. Challenges to understanding text didn't have any capable solutions and generation of the final summary also remained difficult. This caused early research to be less focused on abstractive summarization. However, when neural models and deep learning were introduced, abstractive summarization became more feasible and gained more attention.

2.2 The introduction of Neural Models

This section will discuss the introduction and implications of neural models and deep learning to ATS. Although the first ideas of neural networks date back to the 1950s, neural networks and deep learning methods have gained popularity since 2012. It was discovered that neural networks performed drastically better by using many layers instead of just a few. Around this same period, better hardware emerged and data became more easily available, paving the way for neural models and deep learning. Neural models have shown great results in many domains, and NLP is one of them. Feedforward Neural Networks were used to create word embeddings such as word2vec [91], one of the first major applications of neural models for NLP usage. Word embeddings represent words as continuous vector spaces, which enables them to represent sentences much better than earlier used techniques such as Bag-of-Words or *n*-grams. An *n*-gram is usually a sequence of adjacent words. The *n* denotes the amount of words that the *n*-gram contains.

Recurrent Neural Networks (RNNs) were introduced to address the limitations of traditional feedforward neural networks in handling sequential data and variable-length sequences. RNNs can handle such variable-length sequences and were subsequently implemented for several NLP tasks. They were implemented in many NLP tasks because RNNs are versatile in their inputs and outputs. RNNs can handle variable-length sequences as input and output. The GRU [25] and LSTM architecture [56], a variant of RNNs, was especially useful as it deals with the vanishing gradient problem, enabling it to capture long-term dependencies better than regular RNNs. However difficult sequence-to-sequence tasks such as ATS still proved to be a stretch for RNNs. Eventually, encoder-decoder architectures [120] [25] were introduced. A simplified visualisation of this architecture can be seen in Figure 3.

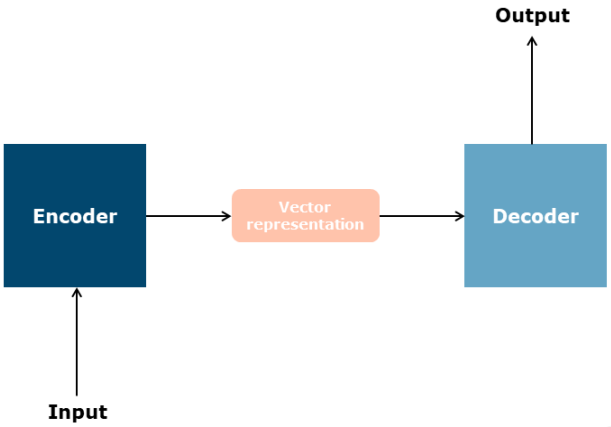


Fig. 3. Encoder-Decoder Architecture.

These architectures make use of an encoder, to encode sequential data into a single fixed-length vector representation, and a decoder to decode this vector representation into the desired output. In combination with RNNs

they prove effective for sequence-to-sequence NLP tasks [120] [25] such as machine translation or ATS. However, encoder-decoder architectures can be used for many other tasks and types of data. With this new architecture more complex NLP tasks suddenly became more feasible.

The introduction of the Transformer architecture [126] marked a paradigm shift in NLP. Self-attention mechanisms of Transformers enabled even better capture of long-range dependencies and parallel computation, making them highly effective. For self-attention mechanisms, each word in a sequence is associated with a set of weights that determine how much attention it should pay to every other word in the sequence. The transformer-based model BERT [33] incorporated pre-training on large corpora, resulting in a very capable and effective base model that could be fine-tuned for specific downstream tasks. BERT achieved state-of-the-art results in various NLP benchmarks and had a great effect on subsequent model designs. For example, RoBERTa [84] is a replication of BERT with an optimized pre-training strategy, achieving greater results than BERT. The Transformer architecture outperformed other architectures for NLP tasks but it is also widely adopted in other AI domains. Transformers can be used in multiple forms in combination with the encoder-decoder architecture, which is common practice in current LLMs. For example, BERT [33] is encoder only, the GPT series [106] [131] [12] [98] are decoder only models, and BART [76] consists of an encoder-decoder architecture. While Transformer-based models can be implemented in various configurations, it's worth noting that many of the most prominent large language models (LLMs) in recent years have adopted a decoder-only architecture. This trend is largely due to the efficiency and scalability of decoder-only models in handling autoregressive language generation tasks. Decoder-only models have shown remarkable performance in generating coherent and contextually appropriate text across a wide range of applications. This architectural choice has been exemplified by models like the GPT series. GPT-2 [106] and GPT-3 [131] showcased that increasing the model size architectures caused major performance increases. This was especially prevalent for decoder-only models while retaining high performance on downstream tasks. Large models also began to show so-called emerging capabilities, which can be defined as an ability that is not present in smaller models but is present in larger models [129]. Performance of the LLMs was then increased even further by using techniques such as Instruction Tuning [128] and Reinforcement Learning from Human Feedback [119] [99] to align the performance of LLMs with human preferences. Consequently, decoder-only models have become much more popular than encoder-decoder models in research and application.

Recently, Transfer learning and fine-tuning of pre-trained language models on downstream tasks such as ATS have become common practices in NLP [80]. Models pre-trained on massive datasets can be adapted to downstream tasks with limited task-specific training data. This approach is highly effective because it shows better results and offers easy scalability. Often LMs are fine-tuned on a multitude of downstream tasks, giving them the ability to perform multiple NLP tasks. Often, LMs are then fine-tuned further on one specific task, such as ATS, to improve the LMs capability on this task. This can be especially beneficial for more specific and harder tasks, such as in this research where long regulatory documents need to be summarized.

The following two paragraphs will discuss what important developments were made for extractive and abstractive summarization throughout the development of neural models and deep learning.

2.2.1 Extractive summarization methods. Early approaches to extractive summarization often relied on traditional methods and features mentioned in Section 2.1. However, the introduction of neural models brought about improvements in capturing semantic representations. As mentioned in Section 2.1, the extractive summarization process typically involves creating representations for all sentences, scoring each sentence according to importance, and ultimately selecting those with the highest scores. This strategy is still used with the introduction of Neural models. Using word embeddings has proven effective in the initial representation step [111]. Word embeddings play an important role in enhancing the efficacy of ATS methods because embeddings can create a better representation of all sentences. Word embeddings provide dense vector representations that effectively capture the semantic relationships among words. Additionally, embeddings empower models to understand contextual

nuances, resulting in summaries that are more informative and contextually relevant. Different methods can be employed to determine which sentences should be extracted. One example is BERTSUM [82], which stacks several transformer layers on top of BERT outputs, where the final layer is a sigmoid classifier that performs a binary classification task, determining if the sentence should be extracted or not. BERTSUM is an example of using a pre-trained LM which is then fine-tuned for the downstream task of extractive summarization. Miller [92] uses BERT to create word embeddings which are then used in a K-Means clustering task to identify the sentences that should be extracted. Different models can be used such as GPT-2 [106] and S-BERT [109] to create the embeddings. However, the current state of extractive summarization models encounters limitations, particularly in the absence of models able to handle long context lengths for more efficient and content-aware extractive summarization. The new neural architectures also paved the way for abstractive summarization, which is often considered more interesting as it is more human-like.

2.2.2 Abstractive summarization methods. Up until the development of encoder-decoder architectures, research was still mostly focused on extractive summarization. But with the introduction of neural models and deep learning, issues related to NLU and NLG, mentioned in Section 2.1 could properly be dealt with. The encoder-decoder model could handle more complex sequence-to-sequence tasks which resulted in abstractive summarization becoming the main focus of ATS research. Rush et al. [112] was one of the earlier adopters of neural models to perform abstractive summarization, performing headline generation. This network consisted of an encoder, followed by a neural generative language model. The encoder could be a bag-of-words but a complex model such as a convolutional network was used by Rush et al. This network was then quickly improved upon by Chopra et al. [26] using an RNN with an attention mechanism[5] for the decoder. Nallapati et al. [94] used the same type of RNN to function as the encoder. See et al. [116] introduced a Pointer-Generator Network that dynamically switches between generating words, hence Generator, and copying words directly from the source text, hence pointer. This innovative approach aimed to handle out-of-vocabulary words and incorporated extractive elements into the generated summaries. This type of network was able to deal with the factual inaccuracy of earlier abstractive models. But it also dealt with the issue of models repeating themselves by introducing coverage.

Ling et al. [81] propose a novel coarse-to-fine attention model that uses coarse attention to select top-level chunks and fine attention to read the words of the chunks. This method improved on computational complexity and handling of long documents but empirically lagged behind state-of-the-art baselines. This overview shows that research in ATS started focusing more and more on the attention mechanisms of RNNs. Eventually, Transformers [126] were introduced which eschew recurrence as a whole and solely use attention mechanisms.

As mentioned earlier, it is common practice to use a pre-trained Transformer-based LM which is then fine-tuned on a downstream task [105] [80]. This strategy shows great results for ATS purposes. Models such as BART [76], T5 [107], the GPT-series[106] [131] [12] [98].

PEGASUS [139] differs from the previously mentioned fine-tuning strategy. It was created with extra pre-training objectives which were focused on abstractive text summarization. By adding these ATS-specific pre-training objectives, Zhang et al. hoped to improve performance on downstream ATS tasks. After pre-training, PEGASUS was evaluated on ATS datasets, which are described in Section 2.6.

2.3 Processing long documents

Neural models and deep learning improved performance on all NLP tasks including ATS. But still, the issue of how to process long documents remained. BERT [33] and T5 [107] have a context length of 512 tokens while PEGASUS's [139] and BART's [76] context length is 1024 tokens. Extending the context length of new LLMs is an active research area. This inability to handle long documents is inherently an issue for summarization because summarization is required when source texts need to be condensed and shortened. After all, the texts are long

themselves. Other methods such as breaking up the task or using different architectures for long documents are required.

2.3.1 Breaking up the task. The methods described in this section break up the large summarization task into smaller summarization parts. By segmenting a long text, it becomes feasible to perform summarization on the smaller parts. Breaking up a text or document into smaller parts is called chunking. Chunking is useful for performing operations on long texts. But it is also widely used for saving documents in vector databases. Instead of saving a whole document in a single vector, a document is split into different chunks and every chunk is saved into the database. When a semantic search is performed, the chunking causes the semantic search to be more effective and accurate.

For the chunking operation, two different methods can be employed:

- Fixed-Size chunking with overlapping.
- Context-Aware chunking.
- Recursive chunking.
- Specialized chunking.

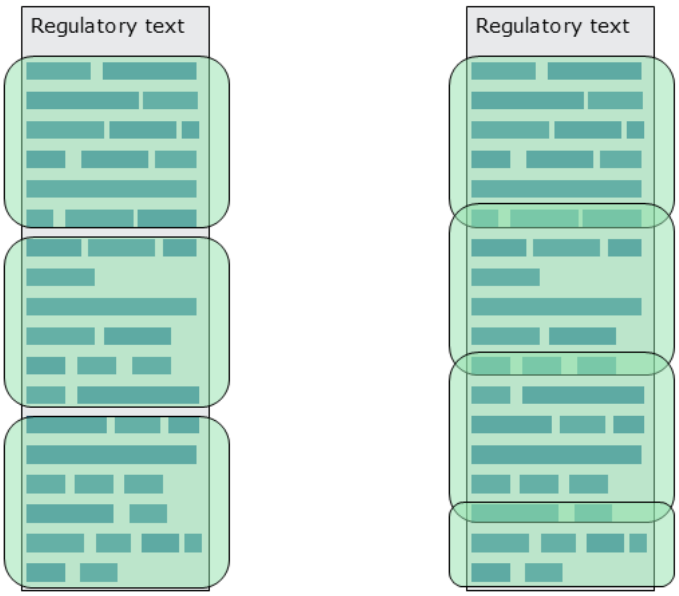
Fixed-size chunking breaks down the text into chunks of a specified number of characters or tokens. No content or structure is taken into consideration, making it cheap and easy. A longer context length could potentially perform better because it enables a model to process longer input texts. It could therefore be easier for an LM with a longer context length to obtain the global context of a text. The global meaning and context of a text can be missed when only a part of the text is processed. The length of the chunks is therefore dependent on the context length of the LM that's going to process the chunk. Fixed-size chunking with overlapping creates chunks of a fixed size that have an overlap of text with adjacent chunks. Overlapping refers to the practice of letting adjacent chunks share an amount of information. Figure 4a illustrates chunking without overlapping, while Figure 4b illustrates chunking with overlapping. This information is expressed in text using the number of tokens, words, or characters. By allowing adjacent chunks to share text, the loss of global context between chunks can be prevented.

Another chunking option is Context-Aware chunking. This method is also known as Content-Aware chunking. It focuses on creating chunks based on the context of the text. By considering the context, chunking can result in more accurate and refined segmentation of the text. An example of no Context-Aware chunking and Context-Aware chunking can be seen in Figure 5.

Recursive chunking divides a text into chunks hierarchically and iteratively using a set of separators. If the initial split doesn't produce chunks of the required size, the resulting chunks are split again using a different separator. This is repeated until the required chunk size is achieved.

Other methods can be used which leverage the specific, structured content of a text. These methods take advantage of Markup languages such as HTML or LaTeX. These methods are sometimes referred to as Structured Chunking or Specialized Chunking.

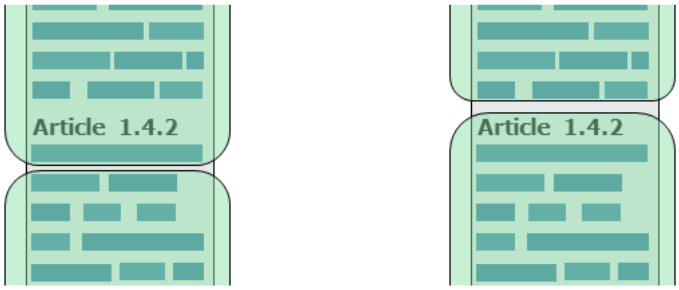
Gidiotis et al. [45] use context-aware chunking with their Divide-ANd-ConquER (DANCER) approach [45]. DANCER was introduced to be able to deal with long academic articles. In the divide phase, sections are extracted and classified into section types using the discourse structure of the article. In the conquer phase each section is summarized and all summaries are then concatenated to form the final summary. To train the model used for DANCER [45], a long document and its coherent summary are split up into multiple source-target pairs using ROUGE scores. ROUGE is an evaluation metric that measures the similarity between texts by measuring the number of matching n-grams. ROUGE scores range from 0 to 1, with a higher score indicating a better summary. More information on ROUGE scores and evaluation metrics can be found in Section 2.7. The source-target pairs are then used to train a model that learns to summarize each part of the document separately. This approach breaks down the total summarization task into smaller summarization problems and reduces computational



(a) Chunking without overlapping.

(b) Chunking with overlapping.

Fig. 4. Chunking without overlapping and with overlapping. An opaque green rectangle represents a chunk.



(a) No Context-Aware Chunking.

(b) Context-Aware Chunking.

Fig. 5. No Context-Aware chunking compared to Context-Aware chunking.

complexity. Shen et al. [118] propose an improved version of the divide-and-conquer approach by directly learning which section belongs to which part of the summary instead of using ROUGE for this alignment. It can also be argued that matching the source and the target through ROUGE is a weakness. ROUGE only measures syntactic overlap and does not consider semantic overlap. But still, it measures the syntax in a suboptimal sense because the syntax is limited to the span of the n-gram. By training a summarizer on source-target pairs, based on ROUGE, you create a summarizer that focuses on creating summaries that would obtain high ROUGE scores. In reality, a

summarization model should produce a concise, fluent version of the source text while preserving its main key points. Besides this drawback, it is also very computationally expensive to match parts of the source and the target.

Celikyilmaz et al. [14] use deep communicating agents to deal with long text. Each agent gets tasked with summarizing a chunk of the text. The agents broadcast their encoding to other agents, allowing them to share information which results in a more coherent summarization.

2.3.2 Long document architectures. The two previously mentioned approaches focus on breaking up the summarization task into multiple smaller summarization tasks. This paragraph will discuss specific transformer-based models that are specialized in handling long documents. Transformers are very powerful but memory and computation requirements of the self-attention mechanism grow quadratically with sequence length. For self-attention mechanisms, each word in a sequence is associated with a set of weights that determine how much attention it should pay to every other word in the sequence. This specific attention mechanism where attention is calculated between every part of the sequence is subsequently called full attention. These attention weights are computed through a dot product between the query, key, and value vectors associated with each word. The attention score for a pair of words is essentially the dot product of their corresponding query and key vectors, scaled by the square root of the dimensionality of the key vectors. This score is processed by a softmax operation and the resulting attention scores are then used to compute a weighted sum of the value vectors, producing the output for that word. So, a longer sequence implies that more operations have to be performed, making the computational requirements bigger. But more importantly, a longer sequence requires more pairwise operations. A sequence of length N involves pairwise computations for each pair of words, resulting in a total of N^2 attention scores. This quadratic growth in computational requirements becomes a problem for long sequences. As the sequence length increases, the number of attention computations grows quadratically, leading to a significant increase in computational cost and memory requirements. This is a limitation for applications that involve long sequences such as long document summarization.

Long document architectures differ in their attention mechanism by calculating attention between specific parts of the sequence instead of calculating the attention for every possible combination of the sequence. This enables them to process long sequences because the computation requirements will not grow quadratically. Note that the long document architectures focus on changing the attention mechanisms. The different attention mechanisms can be used for any pre-trained transformer model without the need to change the model architecture itself.

Longformer [8] is an encoder-only architecture based on RoBERTa [84]. Longformer [8] was designed to handle long-range dependencies more efficiently than standard transformers. It employs a combination of global attention and sliding window attention instead of full attention. The proposed attention mechanisms scale linearly with the input sequence. These attention mechanisms give Longformer the capability to process texts up to 4096 tokens. LED [8] adds a decoder to the Longformer architecture, turning it into the Longformer Encoder-Decoder model. The decoder does use the full attention mechanism but LED retains its linear computation capability. Subsequently, LED can effectively perform sequence-to-sequence tasks.

BigBird [138] incorporates sparse attention patterns. This enables the model to attend to only a subset of tokens, making it scalable for longer sequences up to 4096 tokens. This allows the model to focus on relevant parts of the input sequence while significantly reducing computational demands for processing long documents. Other examples of long document architectures are LongT5[50] or PegasusX [102], both having a context length of 16384 tokens. Besides using a different attention mechanism than full attention, PegasusX also utilizes additional long input pre-training tasks.

Extending the context length of LLMs is an active research area. Newly released LLMs often have improved context lengths. These base models are often adjusted to create new This can be seen for the GPT series where the context length of their base models is consistently increased. Furthermore, models are adjusted to create new

versions with increased context lengths. For example, Together.ai released LLaMA-2-7B-32k [1], which is an LLM based on LLaMA-2 [122] with a context length of 32768.

2.4 Two- or multi-step methods

Where section 2.3 discussed methods like breaking up the task or using long document architectures, this section discusses two- or multi-step methods to summarize long documents. This is the method that will be implemented in this research. The two-step and multi-step methods will be discussed. Also, two specific papers will be discussed more intensively because they share a resemblance to this work.

2.4.1 Two-step methods. Pilault et al. [103] present a method that can produce a coherent and fluid abstractive summary by adding one simple extractive step before generating the abstractive summary. The extractive parts are then used beside the original text as input for the transformer. This method achieves higher ROUGE scores and was preferred for coherence and fluency by human evaluation. This research experimented with four different document datasets. CreativeSumm [70] is created for automatic summarization for creative writing. Its task is to summarize movie scripts, which is challenging due to their long length and complex format. Additionally, the script is pre-processed to enhance performance. Results show an improvement over baseline models. Liu et al. [82] conduct a similar research by first doing an extractive step. While creating the final abstractive summary of Wikipedia articles, the extracted sentences are added as an extra input. The model significantly outperformed traditional transformer models which did t use the extractive step. Another two-step summarization method was proposed by BleiWeiss et al. [10] where a two-step method is proposed for long biographical novels. Similar or superior performance was achieved compared to six baseline models. Their proposed method performed abstractive summarization while focusing on factual consistency.

Klaus et al. [71] make use of a two-step method to summarize legal regulatory documents. Klaus et al. use TextRank [90], a graph-based extractive summarization approach, for the first extractive step and BERT [33] or RoBERTa [84] for a second extractive step. TextRank is a graph-based extractive summarization approach, which works similarly to LexRank. Compared to just using TextRank or BERT, the two-step methods showcase better ROUGE scores. However, Klaus et al. only evaluate using ROUGE scores and do not consider any other evaluation metrics which can give a skewed view. More evaluation metrics are discussed in Section 2.7, which also discusses the pros and cons of each evaluation metric. Klaus et al. make use of the same type of document-summary pairs that this research plans on using; the EUR-Lex database which consists of documents from the European Union law platform with corresponding manually curated summaries. They create document-summary pairs using a scraper which results in a dataset containing 4595 document-summary pairs. This dataset has seen little usage in other papers and articles. However, for this research, it could prove useful due to its size. Section 3.1 discusses in more detail which dataset will be used in this research.

Table 1. ROUGE scores of Summ^N on different datasets. The table is taken directly from Zhang et al. [140]

	AMI			ICSI			QMSum-All			QMSum-Gold		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
PGNet	42.60	14.01	22.62	35.89	6.92	15.67	28.74	5.98	25.13	31.52	8.69	27.63
TopicSeg	51.53	12.23	25.47	-	-	-	-	-	-	-	-	-
HMNET	52.36	18.63	24.00	**45.97**	10.14	18.54	32.29	8.67	28.17	36.06	11.36	31.27
TextRank	35.19	6.13	16.70	30.72	4.69	12.97	16.27	2.69	15.41	-	-	-
HAT-BART	52.27	20.15	50.57	43.98	10.83	41.36	-	-	-	-	-	-
DDAMS	53.15	**22.32**	25.67	40.41	11.02	19.18	-	-	-	-	-	-
Summ^N	**53.44**	20.30	**51.39**	45.57	**11.49**	**43.32**	**34.03**	**9.28**	**29.48**	**40.20**	**15.32**	**35.62**

2.4.2 Multi-step methods. An extension to the two-step model was proposed in the form of Summ$^{\text{N}}$ [140] which is a simple, flexible, and effective multi-stage framework for long input texts. Summ$^{\text{N}}$ splits the data samples and generates coarse extractive summaries, possibly over multiple stages (N), before producing a final fine-grained abstractive summary. This method outperformed previous state-of-the-art methods on different datasets, which can be seen in Table 1.

Using more coarse stages increases ROUGE scores until the product of the final coarse stage does not exceed the context length of the abstractive model. This can be seen in Table 2 where BART is the backbone model, meaning that it is the model used for the fine-grained step. On different datasets, it can be seen that a Summ^{N} method outperforms just using BART as the backbone model on long summarization tasks. The model will receive more information this way than by simply truncating the source text. Summ$^{\text{N}}$ estimates the amount of coarse stages as follows:

$$\hat{N} = \frac{1 + \log K - \log d_1}{\log c_1 - \log K} \tag{1}$$

Where K represents the context length of the backbone model, d_1 and c_1 the average length of the source text and coarse segments. This formula is adjusted so that the final coarse stage outputs $2K$ tokens instead of K tokens. Intermediary results found that giving the fine-grained stage a document of $2K$ tokens instead of K improved ROUGE scores.

Table 2. Improvements of Summ^{N} over BART as backbone model on AMI, ICSI, and QMSum datasets. More information on these datasets can be found in Section 2.6. The table is taken directly from Zhang et al. [140]

		R-1	R-2	R-L
AMI	Backbone	46.57	16.41	44.61
	Summ^{N}	**53.44**	**20.30**	**51.39**
ICSI	Backbone	39.91	9.98	38.17
	Summ^{N}	**45.57**	**11.49**	**43.32**
QMSum-All	Backbone	29.20	6.37	25.49
	Summ^{N}	**34.03**	**9.28**	**29.48**
QMSum-Gold	Backbone	32.18	8.48	28.56
	Summ^{N}	**40.20**	**15.32**	**35.62**

Summ^{N} showcases that summarizing for multiple steps or stages improves the final summary. However, Summ^{N} makes use of abstractive summarization for both the coarse stages and the final fine-grained stage. This research wants to perform extractive summarization for the coarse stages and abstractive summarization for the final fine-grained stage. A comparison of our method against Summ^{N} could be interesting to evaluate if extractive summarization or abstractive summarization works better for the coarse stages. In Summ^{N} the coarse summarizer is trained by matching source segments and target segments using ROUGE. However, this is not a good method to teach a model how to summarize a piece of text, for the same reasons mentioned in Section 2.3 for DANCER [45]. This method is also computationally intensive because the best target segment must be found for every input segment. Also, Zhang et al. evaluate Summ^{N} only on ROUGE scores and do not consider any other evaluation metrics which can give a skewed view of the performance. More evaluation metrics are discussed in Section 2.7, which also discusses the pros and cons of each evaluation metric. Zhang et al. also do not use Summ^{N} for regulatory or legal texts, which will be done in this research.

2.5 Summarization of legal and regulatory texts

Section 2.3 and section 2.4 discussed methods on how to handle long documents for summarization purposes. This section will discuss methods on how to process and summarize legal and regulatory texts. Identifying how the NLP field processes legal and regulatory texts can prove beneficial for the performance of our proposed summarization model. This section will first discuss the characteristics of legal and regulatory texts and what types of NLP tasks are performed on them. Then, different types of legal LMs will be discussed. Finally, specific legal text summarization methods will be discussed.

2.5.1 Legal and regulatory texts. Regulatory texts are a subset of legal texts. Regulatory texts specifically contain rules and standards established by regulatory bodies to implement and enforce laws within a particular domain. Legal texts, on the other hand, encompass a broader range of documents that include laws, regulations, (court) cases, contracts, and more. This research will focus on regulatory texts. Numerous types of research involving NLP in the legal field leverage multiple types of legal texts.

Standard NLP tasks like summarization, named entity recognition, text classification, question answering, and others are commonly applied to legal texts, aiding in comprehension and text processing. In addition to these conventional NLP tasks, the legal domain often requires specialized applications unique to its domain. Zhong et al. [144] identify Legal Judgment Prediction (LJP), Similar Case Matching (SCM), and Legal Question Answering (LQA) as unique applications of AI in legal scenarios. LJP concerns the prediction of judgment results from the facts of a case [144], statutory articles from the Civil Law system [139], and other information such as arguments and claims [42]. SCM focuses on finding pairs of similar cases [144], varying on the definition of similarity. Finding similar cases can be important because, for Common Law systems, judicial decisions are made according to previous similar and representative cases. But SCM also proves useful for legal practitioners to obtain information on similar cases. LQA is similar to 'regular' question answering but it specialises in the legal domain.

Legal texts differ from other types of texts in several aspects, which makes them harder to process and summarize. There are multiple aspects which make legal documents uniquely difficult [123] [66] [62]:

- **Size**: The length of legal documents tends to be longer than documents in other domains.
- **Structure**: Legal documents contain internal structures. These structures can be exploited but the structure varies greatly per type of legal document.
- **References and citations**: Legal documents contain many references and citations. References refer to a piece of information inside of a text. Citations refer to information outside of the text. References and citations play a major role because they can point towards important information.
- **Ambiguity:** Legal documents can be ambiguous because of the importance of their origin. A document is deemed more important if it originates from the Supreme Court. This ambiguity also makes references and citations more complicated.
- **Vocabulary:** The vocabulary of legal documents is different as it uses specific, sometimes complex, terminology.

These difficult characteristics of legal documents ask for different types of solutions and approaches to perform traditional NLP tasks.

2.5.2 Legal Language Models. Most LMs are general-purpose models which have limited capabilities on certain specialized texts. Domain-specific LMs are created by adding additional domain-specific texts during the pre-training process or by replacing existing data with domain-specific data. The addition or replacement of data can be done after the initial pre-training process. But the model can also be pre-trained from scratch. Domain-specific LMs can be created by multiple strategies:

- Further pre-training on domain-specific corpora.

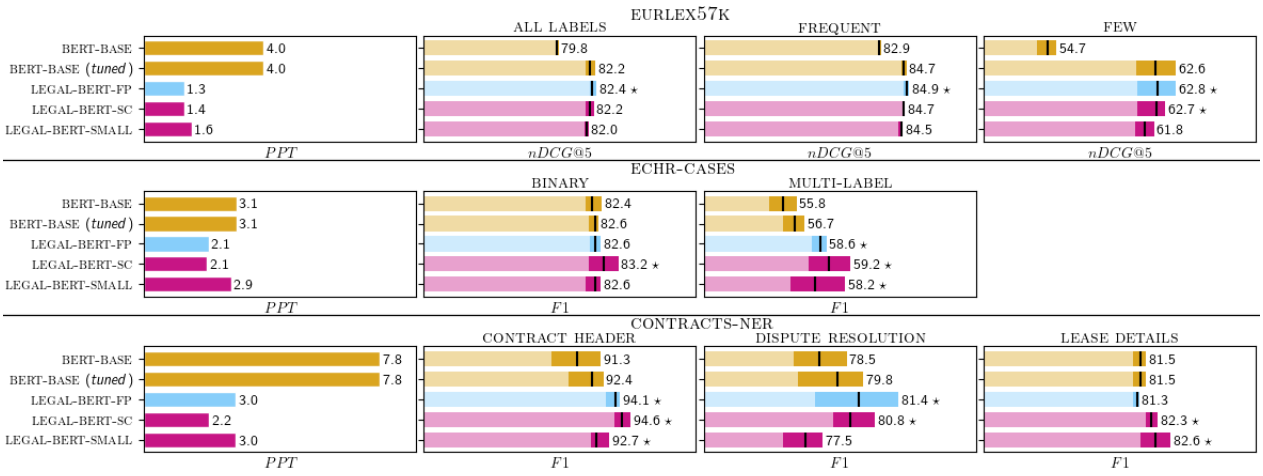


Fig. 6. Perplexities (*PPT*), *nDCG@5* and *F1-score* on test data for three evaluation datasets for all models. *PPT* is a measure of uncertainty in the prediction of the sample, indicating that a lower score means the model is more certain of its output. *nDCG@5* is the Normalized Discounted Cumulative Gain for the first top 5 ranked items. It is a metric that measures the ranking quality and Chalkidis et al. state that it is a good metric for Large-Scale Multi-Label text classification [17] [18]. Multiple runs have been performed with the black line indicating the average of the runs. The opaque sections indicate the range of the minimum and maximum over the runs. A star indicates that the version of Legal-BERT outperformed the tuned BERT-BASE on average. The tuned version of BERT-BASE was tuned on one of the three datasets. Figure directly taken from Chalkidis et al. [17]

- Pre-train on domain-specific corpora from scratch.
- Fine-tune on domain-specific tasks.

The first two options involve changing or adding the pre-training procedure of an LM. Besides changing the pre-training process, fine-tuning is also an option to create domain-specific LMs. However, datasets are required on which the LM can be fine-tuned. Results [52] have shown that pre-training a model on a domain-specific task or corpus improves performance on these domains. Domain-specific LMs are therefore interesting because they have an expert view of a domain while retaining their capability to perform a wide range of NLP tasks. Domain-specific models have been developed for a multitude of different domains such as the healthcare domain (ClinicalBERT [59]), biomedical sciences (BioBERT[75]), the scientific domain (SciBERT[7], the finance domain (FinBERT [135], BloombergGPT [130]). But pre-trained LMs for law purposes also exist: Lawyer LLaMA[61], Lawformer [133], LegalLongformer [87], PEGASUS-Billsum [139], LegalBERT [17], CaseLawBERT [143], PoL-BERT [54] and LexLM [20]. An overview of legal LMs can be seen in Table 3. Results from Chalkidis et al. (2023) have been implemented to compare a subset of the legal LMs on their performance.

Lawyer LLaMA [61] and Lawformer [61] are Chinese-based LMs that showed a performance increase after additional pre-training on a legal corpus. Lawyer LLaMA also showed that its performance on general tasks improved slightly. This is a positive feature because it shows that pre-training a model on a domain will not cause it to perform worse on other general NLP tasks. Lawformer showed slight improvements in judgment prediction and legal question answering. However, it showed significant improvements in legal case retrieval. But Lawformer is also more capable of handling long documents because it is based on the Longformer architecture [133].

Mamakas et al. [87] created LegalLongformer to handle long legal texts, able to process up to 4096 tokens. Mamakas et al. also introduced a version that can handle up to 8192 tokens. LegalLongformer is initialized with LegalBERT's [17] parameters and Longformer's [8] attention mechanism. Then LegalLongformer is fine-tuned on LexGLUE [21]. LexGLUE contains several datasets focused on downstream legal tasks. LegalLongformer shows improved results over LegalBERT [17] and seems to perform well on LexGLUE tasks. Unfortunately, there is no version of LegalLongformer available online. PEGASUS-BillSum [139]is a version of the PEGASUS model [139], which has been fine-tuned on the BillSum dataset.

Chalkidis et al. [17] investigate how to adapt BERT [33] to the legal domain. Legal-BERT will be discussed extensively as it was an early domain-specific LMs while proving to be effective. Chalkidis et al. experimented with further pre-training on domain-specific corpora (Legal-BERT-FP) and with pre-training on domain-specific corpora from scratch (Legal-BERT-SC). Legal-BERT-FP is pre-trained on a combination of multiple legal text domains. But other versions of Legal-BERT-FP are also created which are pre-trained only on one legal text subdomain. The types of legal texts are EU and UK legislation, two types of court cases, and US contracts. Chalkidis et al. also create Legal-BERT-FP versions that are just pre-trained on one of the sub-domains of EU legislation (EURLEX), European Court of Human Rights (ECHR) cases, and US contracts.

After their training procedure, Legal-BERT-SC and Legal-BERT-FP were evaluated on several legal NLP tasks using different datasets. EURLEX57K [18] was used for Large-Scale Multi-Label text classification. ECHR-CASES [15] was used for binary and multi-label text classification on cases from the European Court of Human Rights [3]. CONTRACTS-NER [16] [19] was used for named entity recognition (NER) on US contracts. The results on these datasets can be seen in figure 6. It can be seen that in all cases some variant of Legal-BERT outperforms the tuned version of BERT-BASE on average. For EURLEX57K the difference is less substantial. For ECHR-CASES and CONTRACTS-NER, Legal-BERT outperforms the tuned version of BERT-BASE less and more substantially, varying per dataset per task. It is however evident that a version of Legal-BERT outperforms BERT-BASE in all cases. The overall gains are small when considering the substantial data and computational requirements needed to pre-train Legal-BERT.

The optimal performance of Legal-BERT varies depending on the dataset and task at hand. It must be noted that the version of Legal-BERT-FP in figure 6 is dependent on which version of Legal-BERT-FP performed best. Legal-BERT-FP could either be further pre-trained on all legal domains or only on one legal subdomain. The results were dependent on the dataset and the amount of training steps. For EURLEX57K, the difference was marginal: Pre-training just on the subdomain gave a *nDCG@5* score of 82.7 while pre-training on all domains resulted in a score of 82.6 [17]. For this research, Legal-BERT-SC or the Legal-BERT-FP version trained on the EUR-Lex dataset will be used because this research involves European Legislation texts.

Besides Legal-BERT, there are more legal LMs based on BERT. CaseLawBERT [143] is pre-trained from scratch for over 2 million steps on the Harvard Law case corpus. Zheng et al. [143] discuss when full pre-training is appropriate. Full pre-training of a model is a substantial effort, computationally and financially. Zhang et al. argue that for legal NLP better benchmarks are required to be able to research if pre-training is justified. In their research, they not only make this argument but introduce a new benchmark to address this gap. CaseLawBERT shows improved results over BERT on the newly introduced benchmark and two other benchmarks.

PoL-BERT [54] is based on Roberta and trained from scratch on the Pile of Law corpus [54], which is compiled of legal and administrative text. The size of the Pile of Law corpus is the biggest of all the legal BERT variants.

Chalkidis et al. present two pre-trained legal LMs called LexLM-Base and LexLM-Large. But Chalkidis et al. [20] also introduce a multinational English legal corpus, LeXFiles [20] and a legal knowledge benchmark, LegalLAMA [20]. LegalLAMA is focused on the prior knowledge of an LM by testing its prior knowledge in the legal domain. Both LexLM models are further pre-trained on the LeXFiles corpus. They are then evaluated on LegalLAMA and LexGLUE [21]. The LexLM models are then compared to RoBERTa-Base and RoBERTa-Large [84], LegalBERT [17], CaseLawBERT [143] and PoL-BERT[54]. In the evaluation on LegalLAMA, the top-performing model is

LexLM-Large, followed by LexLM-Base in second place, and LegalBERT [17] in third place. It is surprising to see that RoBERTa-Large outperforms CaseLawBERT and PoL-BERT. All models are fine-tuned on LexGLUE for 1 epoch. LexLM-Large performs best with RoBERTa-Large performing second best. LexLM-base comes in as 6th best, surprisingly. After fine-tuning, all models are again tested on their prior knowledge using LegalLAMA. The same pattern can be seen as before where the LexLM models and RoBERTa-Large perform better than CaseLawBERT and PoL-BERT. Chalkidis et al. state that RoBERTa-L and the LexLM models have better prior legal knowledge, which also results in better performance downstream tasks. Besides the regular LexLM models based on RoBERTa, Chalkidis et al. also introduce a version that uses the Longformer attention mechanism [8].

Table 3. Different types of Legal LMs. FP stands for Further Pre-training, SC stands for pre-training from SCratch and FT stands for Fine-Tuned. Models under the dashed line are of interest for this research. LegalLAMA (1) represents the score on LegalLAMA before fine-tuning on LexGLUE, whereas LegalLAMA (2) represents the score on LegalLAMA after fine-tuning on LexGLUE.

Model	Training method	LegalLAMA [20] (1)	LexGLUE [21]	LegalLAMA [20] (2)
Lawyer LLaMA [61]	FP	-	-	-
Lawformer [133]	FP	-	-	-
LegalLongformer [87]	FT	-	-	-
Pegasus-BillSum [139]	FT	-	-	-
LegalBERT-FP [17]	FP	-	-	-
LegalBERT-SC [17]	SC	3	6	3
CaseLawBERT [143]	FP	5	7	6
PoL-BERT [54]	SC	6	3	5
LexLM-Base [20]	FP	2	4	2
LexLM-Large [20]	FP	**1**	**1**	**1**

2.5.3 Specific legal text summarization methods. Domain-independent ATS methods mentioned in Section 2.1 and Section 2.2 can also be applied for legal text summarization [62] [66]. However different methods have also been proposed with a specific focus on legal text summarization. The methods found were all extractive-based summarization methods. The domain-specific legal text summarization methods found are as follows:

- **Citation-based methods** focus on using documents that the source text cites and documents that cite the source text. These documents are used directly or as additional help to create the summary.
- **Ripple down rules-based methods** are created incrementally by a domain expert without a knowledge engineer. The knowledge base is refined over time.
- **Rhetorical role-based methods** focus on understanding the semantic function of sentences by associating them with specific rhetorical roles. By aligning sentences with their rhetorical roles, the source text can be better understood for the summarization task.
- **Nature-based methods** draw inspiration from nature to address summarization as an optimization challenge. A fitness function serves as a metric to find the best possible solution for the task.

For further explanation of these methods, Jain et al. (2021) and Kanapala et al. (2019) provide detailed insights. This research focuses on using legal language models, as previously mentioned. This approach is chosen due to the limited research on domain-independent abstractive text summarization (ATS) methods. Additionally, recent years have seen a scarcity of studies in this area.

Table 4. Different types of datasets. Size is expressed in the amount of document/summary pairs. Only the English parts of XL-sum and EUR-Lex-Sum have been considered. * indicates that QMSum uses query-summarization pairs.

Dataset	Domain	± Size
CNN/DailyMail [94]	News	285,000
NYT [113]	News	655,000
Multi-News [41]	News	56,000
Gigaword [48]	News	4,000,000
XSum [95]	News	227,000
XL-sum [53]	News	301,000
WikiHow [72]	Website texts	230,000
Reddit TIFU [69]	Website texts	120,000
PubMed [27]	Scientific articles	133,000
Arxiv [27]	Scientific articles	215,000
AMI [89]	Meetings	137
ISCI [63]	Meetings	59
QMSum* [146]	Meetings	1808
BigPatent [117]	Legal and Regulatory	1,300,000
BillSum [36]	Legal and Regulatory	20,000
GovReport [60]	Legal and Regulatory	20,000
EUR-Lex-Sum [4]	Legal and Regulatory	1500
EurLexSum [71]	Legal and Regulatory	4595

2.6 Datasets

To fine-tune a pre-trained model a large dataset is required containing document-summary pairs. Fortunately, more data is available nowadays because of massive amounts of text residing on the internet. Obtaining high-quality summaries alongside the texts can pose a challenge, however. Ideally, summaries would be obtained by letting experts create summaries but this is time-consuming and expensive. Often summaries are created by using headlines, TL;DR (Too Long; Didn't Read) sections, abstracts, or introductions. Besides this, there are also many types of domains to be used. Fine-tuning a summarization model on a domain that differs from the domain that the model that will be used can result in lower performance. This section will discuss different commonly used summarization datasets. The subsections are divided per domain and discuss the available datasets and their properties. Table 4 provides a reference for specific datasets. It presents a list of datasets, along with their respective domains and sizes. Finally, the EUR-Lex-Sum [4] will be discussed, which is used in this research.

2.6.1 News. Most previous research done on ATS involves the usage of texts from the news domain. The CNN/DailyMail dataset was constructed by Nallapati et al [94] by modifying an existing dataset, created by Hermann et al. [55]. The CNN/DailyMail dataset contains human-generated summaries alongside news articles which are the original text. The summaries were created by scraping all bullets of each story in the original order to create a multi-sentence summary. This dataset is used extensively in the summarization domain because it contains over 285,000 document/summary pairs and the summaries are of high quality and human-written. The dataset also covers many different news article topics which offers great generalization purposes across the summarization domain. Overall, this dataset is one of the most used datasets in the field of ATS. Besides the CNN/DailyMail dataset, there are other multiple news-focused datasets such as NYT [113], Multi-News [41],

Gigaword [48], Newsroom [49], XSum [95], XL-sum [53] and the datasets [31] [77] [28] [29] ranging from 2008 till 2011.

2.6.2 Website texts. WikiHow [72] comprises a dataset containing over 230,000 pairs of articles and summaries. These are extracted and compiled from a diverse range of topics covered by various human authors in the WikiHow online knowledge base. The articles exhibit a broad spectrum of styles, reflecting the high diversity within the dataset. Reddit TIFU [69] consists of 120,000 posts from the online discussion forum Reddit.

2.6.3 Scientific Articles. Previously mentioned datasets contain relatively short reference texts. This makes sense due to the nature of the type of text. Datasets containing longer reference texts are PubMed [27] and Arxiv [27]. Both of these datasets consist of scientific articles in their respective categories.

2.6.4 Meetings. AMI [89] and ICSI [63] are datasets containing meeting scripts. The scripts are generated by Automatic Speech Recognition, which results in an error rate of around 36-37% for both datasets. AMI's scripts are collected from product design meetings from a company. ISCI's scripts are collected from academic group meetings. QMSum [146] is a query-based meeting summarization dataset. QMSum contains annotated query-summarization pairs of committee meetings of the Welsh Parliament and the Parliament of Canada. QMSum also includes the AMI and ISCI datasets besides the committee meetings. QMSum focuses on creating the right summary based on a query. Summarization of meetings can be very valuable in a wide range of contexts. These types of datasets are also interesting for long document summarization because of the length of the meetings. However, a difficult aspect of these datasets is that they contain the viewpoints of multiple speakers. Summarization methods have to consider multiple viewpoints in a meeting while also retaining an idea of what the global meeting is about and its outcomes. Meeting datasets and summarization tasks are therefore unique, making them harder to use for the task of this research.

2.6.5 Legal and Regulatory. BigPatent [117] is another long document summarization dataset. It is made up of 1,3 million records of U.S. patent documents along with human-written abstractive summaries. BillSum [36] and GovReport [60] also contain long documents. BillSum is composed of US Congressional and California state bills. GovReport is a large-scale dataset, consisting of U.S. government reports with expert-written abstractive summaries. Both datasets contain around 20,000 document/summary pairs.

2.6.6 EUR-Lex-Sum. This research will make use of the EUR-Lex-Sum dataset [4], which is composed of multi- and cross-lingual datasets for long-form summarization in the legal domain. This dataset consists of documents from the European Union law platform with corresponding manually curated summaries. It contains up to 1500 document/summary pairs for English. The English part of the dataset will be used to fine-tune the abstractive model, meaning that our abstractive model is based on supervised learning. The EUR-Lex-Sum dataset has been chosen because the EUR-Lex-Sum contains even longer reference texts than GovReport and BillSum. GovReport's mean document length is around 9000 tokens, Billsum's mean document length is around 1382 words and EUR-Lex-Sum's mean document length is around 12,000 tokens. EUR-Lex-Sum's maximum document length also surpasses the maximum document length of GovReport and BillSum. Klaus et al. [71] introduce a dataset containing 4595 document-summary pairs, on the topic of European legislation. Because this dataset is bigger than the EUR-Lex-Sum dataset introduced by Aumiller et al. [4].

2.7 Evaluation metrics

Evaluation of generated texts is crucial but difficult; In what sense is summary *A* better than summary *B*? This evaluation process can suffer from the same flaws as the actual summarization process itself, mentioned in 2.1. Consistency and objectivity are hard to keep consistent over multiple summaries. However, it remains of extreme importance that generated summaries are evaluated correctly.

Table 5. Criteria for evaluating generated texts along with new descriptions based on the meta-review of Howcraft et al [57].

Criterion	Description
Fluency	Assessment of the text's smoothness and ease of reading in terms of form, content, and grammar.
Readability	Evaluation of how useful, understandable, and clear the summary is.
Coherence	Measure of logical a text is to its linguistic context.
Naturalness	Assessment of how closely the summary resembles natural language.
Quality	Overall judgment of the summary.
Correctness	Evaluation of correct text is relative to the input text.
Usability	Assessment of how practical and user-friendly the summary is.
Clarity	Determination of how understandable a text is.
Informativeness	Evaluation of how much valuable information is in the text.
Accuracy	Assessment of the precision and correctness of the information in the text.

Without good evaluation metrics, it cannot be determined if a model performs well. So, generated texts are often evaluated along four criteria against a golden standard text: *Coherence*, *Consistency*, *Fluency* and *Relevance* [73] [40]. But other criteria can also be used to evaluate texts such as *Informativeness*, *Factuality*, *Semantic Coverage* or *Adequacy* [136]. However, Howcroft et al. (2020) found that there is little shared practice in human evaluation within NLG research. This lack of consensus arises because there are no standardized definitions for naming the aspects of quality that are to be evaluated. Furthermore, Howcraft et al. observed significant inconsistency in the use of quality criterion names across different research, with the same names often denoting different aspects and vice versa. Howcraft et al. also found that more than half of the research lacked clear definitions for the evaluated criteria, and approximately a quarter failed to explicitly mention the criteria under evaluation. Howcraft and colleagues standardized various evaluation names into a set of normalized criterion names. Table 5 has listed the criteria and descriptions with a short description.

Despite the criterion flaws, human evaluation along the criteria of Table 5 is still used to assess generated texts and summaries. Evaluation is done by crowd-sourced annotators and/or domain experts. Besides the inconsistency in criteria, Gillick and Liu [46] state that summary evaluations from non-experts can differ greatly from domain experts. Besides this, human evaluators can be biased, preferring certain phrases and sentences [147]. Using annotators is also expensive and time-consuming. Therefore, automatic evaluation metrics are a good alternative. These are a lot cheaper and will produce less noisy evaluations. Many automatic evaluation metrics have been proposed. A summary must incorporate both precision, by conveying only factual information, and recall, by including the most crucial details from the source text. Combining both metrics into the F-score ensures accuracy and completeness in capturing the essential elements of the original content. The other part of this section will discuss various automated evaluation metrics that could be of importance for this research.

2.7.1 n*-gram-based metrics.* *n*-gram based approaches use *n*-grams to evaluate machine generated texts. While originally designed for Machine Translation, BLEU is an *n*-gram matching approach that compares a generated text with a reference text. BLEU [101] is one of the most used evaluation metrics for NLG. BLEU calculates the precision of a generated text by comparing the *n*-grams in the generated text with the *n*-grams of the reference text. A higher precision score indicates better performance. The formula for precision (P) is given by:

$$P = \frac{\text{\# Matching } n\text{-grams}}{\text{\# } n\text{-grams in generated text}} \tag{2}$$

BLEU does not consider recall, meaning it does not penalize for missing relevant information present in the reference texts. The formula for recall (R) is given by:

$$R = \frac{\text{\# Matching } n\text{-grams}}{\text{\# } n\text{-grams in reference text}} \tag{3}$$

Because BLEU solely uses n-grams, it does not take syntactic integrity or grammatical correctness into account. Also, BLEU fails to consider semantic correlation. More recent works have shown that BLEU does not have a strong correlation with human evaluation and that BLEU should not be used to evaluate other types of NLP tasks outside of machine translation [110]. METEOR [6] is another metric used for evaluating machine-generated translations. METEOR aims for a more comprehensive evaluation by considering additional linguistic features such as recall, stemming, synonymy, and explicit ordering of words. CHRf [104] is another n-gram-based approach which focuses on the F-score of character n-grams. The F-score is the harmonic mean of the precision and recall. The formula for the F-score (F) is given by:

$$F = \frac{2 \times P \times R}{P + R} \tag{4}$$

Because it is a character-level metric, CHRf can handle variations in word forms and is less sensitive to word segmentation errors.

ROUGE [79] is designed specifically for evaluating automatic summarization. ROUGE measures the number of matching n-grams or skip-grams between the reference summary and the model-generated summary. ROUGE-1 refers to the overlap of unigrams between the two summaries, while ROUGE-2 refers to the overlap of bigrams. ROUGE-L represents the ROUGE score of the Longest Common Subsequence. These three variations are the most used evaluation metrics for summarization tasks. ROUGE scores range from 0 to 1, with a higher score indicating a better summary. ROUGE calculates the recall of a generated text as it is designed to evaluate how well machine-generated summaries capture important content from the reference text. By focusing on recall, ROUGE provides insights into the effectiveness of a summarization model in terms of information coverage and content retention. ROUGE scores are easy and inexpensive to calculate and they are a proper indication of the quality of a summary. ROUGE shows some positive correlation with human evaluation according to earlier work[79]. However, more previous work has criticised this assumption and many other evaluation metrics correlate much better with human judgment. But still, ROUGE scores do not say everything about a summarization. If the reference summary is extractive by nature and an abstractive summarization model is evaluated, low ROUGE scores might be obtained. This occurs because ROUGE measures syntactic overlap rather than semantic overlap. Even if the meaning of the reference summary and the generated summary align, the lack of syntactic similarity results in lower ROUGE scores. Often ROUGE is combined with human evaluation, but this combination still fails to consider other features such as factual correctness [73]. Still, ROUGE will be used as one of the evaluation metrics but it is important to use other evaluation metrics which offset ROUGE's drawbacks.

2.7.2 Embedding-based metrics. Embedding-based metrics for ATS evaluate the quality of generated summaries by comparing the semantic similarity between the generated summary and one or more reference summaries using vector representations, embeddings, of words or sentences. These metrics aim to capture the content and semantics of the text while n-gram-based approaches focus on surface-level matching of word sequences, disregarding the semantic relationships. BERTScore [139] is an embedding-based metric. It uses BERT embeddings

to calculate a score for similarity between a generated sentence and a target sentence. BERTScore addresses two drawbacks of *n*-gram-based approaches. First, BERTScore can accurately evaluate paraphrases. Secondly, BERTScore can accurately capture distant dependencies and penalize semantically-critical ordering changes. The process of BERTScore is as follows. A BERT embedding is computed for a reference text and a generated text. Then for all reference tokens, the cosine similarity is calculated pairwise with all candidate tokens. Thereafter, each token of the generated sentence is matched to the most similar token of the reference sentence and vice versa. This is done to compute the recall, precision, and F1 score. Then the selected token matches are weighted by the Inverse Document Frequency to incorporate rare words' importance. Finally, the BERTScore values are rescaled to improve human readability and make them fall in a range of [0,1], with higher scores indicating higher similarity. BERTScore can capture semantic similarity between texts better but still, it can miss subtle differences and miss higher level concepts in a summary, especially when summarizations get longer. BERTScore shows a better correlation with human judgement on Machine Translation on the WMT 2018 dataset [11] and on Image Captioning on the COCO Captioning Challenge [22]. For our research, summaries are often quite long so BERTScore might not give a proper evaluation of the summary because it relies on a context length of 512 tokens, just like BERT. The mean summary length of the EUR-Lex-Sum [4] dataset is around 800 tokens. This means that BERTScore is unable to evaluate every summary as a whole.

MoverScore [141] uses contextualized representations with a distance measure to compare generated texts to reference texts. The distance measure captures the amount of shared content between two texts, which is the union between text A and text B. But it also captures how much the generated text deviates from the reference text by subtracting the intersection of text A and B from the union between text A and B.

Zhao et al. [141] use different types of embeddings such as static embeddings and contextualized embeddings. MoverScore can also vary in other types of dimensions such as Granularity, Fine-tuning Tasks, and Aggregation. The first variant of MoverScore is Word Mover. It makes use of Unigram-based word embeddings. The other is Sentence Mover which makes use of sentence embeddings. Both versions can vary among the other three different dimensions. For text summarization, Zhao et al. [141] found that WordMover performed best, showing a higher correlation with human judgments than ROUGE on the TAC-2008 dataset [31] and near similar correlation on the TAC-2009 dataset [77]. Generated and reference summaries had less than 100 words. In this research, summaries are longer so it is unclear how well MoverScore performs as an evaluation metric.

2.7.3 Generation Probability-based metrics. BARTScore [136] is based on BART [76] and it is a generation probability-based evaluation metric. Given a golden reference text and a target text, BARTScore calculates the generation probability of the target text being conditioned on the reference text. BARTScore makes use of four different methods for using BARTScore based on different generation directions. These methods are Faithfulness, Precision, Recall, and F-score. They take the source text, golden-reference text, and the generated text into consideration in multiple directions. These different ways of using BARTScore offer a better evaluation for specific tasks. Summarization evaluation mostly relies on the Faithfulness and Recall methods of BARTScore. Faithfulness assesses the likelihood of the generated text being generated from the source text, and it can also serve as a gauge for evaluating specific aspects of the target text, such as *Coherence* and *Fluency*. The Recall version quantifies the probability with which a golden-reference text could be created based on the generated text, providing a measure of the *Relevance* of the generated text. BARTScore is based on log-likelihoods and doesn't adhere to a fixed scoring range. The scores are typically negative, with higher (less negative) values indicating better quality. The exact range and distribution of scores will depend on the specifics of the task and the texts being evaluated. BARTScore shows an overall better correlation with human judgment than ROUGE, BERTScore, and MoverScore on the QAGS-CNN [127], QAGS-SUM [127] and WMT 19 dataset [86]. BARTScore is less effective at distinguishing the quality of an extractive summarization but it is more effective at distinguishing the quality of an abstractive summarization [136].

Table 6. Overview of evaluation metrics and, if available, their Pearson (r), and Kendall-Tau (τ) correlations with human evaluation scores on the SummEval dataset [40]. A higher value indicates that there is a higher correlation with human judgments of summaries. The horizontal lines separate the different types of evaluation metrics.

	Coherence		Consistency		Fluency		Relevance		Average	
Evaluation Metric	ρ	τ	ρ	τ	ρ	τ	ρ	τ	ρ	τ
ROUGE-1 [79]	0.167	0.126	0.160	0.130	0.115	0.094	0.326	0.252	0.192	0.150
ROUGE-2 [79]	0.184	0.139	0.187	0.155	0.159	0.128	0.290	0.219	0.205	0.161
ROUGE-L [79]	0.128	0.099	0.115	0.092	0.105	0.084	0.311	0.237	0.165	0.128
BLEU [101]	-	-	-	-	-	-	-	-	-	-
METEOR [6]	-	-	-	-	-	-	-	-	-	-
CHRf [104]	-	-	-	-	-	-	-	-	-	-
BERTScore [139]	0.284	0.211	0.110	0.090	0.193	0.158	0.312	0.243	0.225	0.175
MOVERScore [141]	0.159	0.118	0.157	0.127	0.129	0.105	0.318	0.244	0.191	0.148
BARTScore [136]	0.448	0.342	0.382	0.315	0.356	0.292	0.356	0.273	0.385	0.305
BLANC [124]	-	-	-	-	-	-	-	-	-	-
QuestEval [114]	-	-	-	-	-	-	-	-	-	-
UniEval [145]	0.575	0.442	0.446	0.371	0.449	0.371	0.426	0.325	0.474	0.377
G-EVAL-4 [83]	**0.582**	**0.457**	**0.507**	**0.425**	**0.455**	**0.378**	**0.547**	**0.433**	**0.514**	**0.418**

2.7.4 Referenceless metrics. All automated metrics mentioned so far need a reference text to be able to evaluate the generated text. Referenceless metrics remove the need for high-quality reference summaries which can suffer heavily from layout bias. Many evaluation metrics using reference texts disregard the source text which contains viable information for evaluation. Furthermore, summaries can be written in multiple styles and many summaries can be correct for one document. Referenceless metrics focus on testing how helpful a summary is to its readers. This paragraph will discuss multiple referenceless metrics. More research has been done in previous years on these metrics because they show promising results due to the development of question-answering and Question-generation abilities of LLMs.

BLANC [124] is a measure of how well a summary helps an independent, pre-trained LM understand a document. Vasilyev et al. use BERT [33] as the pre-trained LM for a language understanding task. Words in a text are masked and the LM predicts what specific word is masked. The idea is that the addition of the summary would help BERT understand what word is masked. The addition of a high-quality summary should improve BERT's ability to accurately predict the masked word, in contrast to scenarios where no summary is used. Vasilyev et al. [124] propose two versions of BLANC. BLANC-help uses the summary text by directly concatenating it to each sentence during the prediction task. By reading the summary alongside the masked sentence, the pre-trained LM should get a better idea of what words are masked. The BLANC scores can range from -1 to 1, but the scores are typically from 0 to 0.3 [125] For BLANC-tune, the pre-trained LM is fine-tuned on the summary text, which contains masked words. After learning what the summary means and entails, the masked sentence is given to the pre-trained LM. Vasilyev et al. evaluated their evaluation metric by giving BERT the original summary texts or random sentences. When the original summary texts were used, BLANC scores improved compared to using random sentences. This indicates that BLANC can inform us how good a summary is. BLANC is more correlated to human evaluation than ROUGE-L for a Machine Translation task but only slightly [124], combining BLANC with ROUGE-L showed more correlation with human evaluation however.

QuestEval [114] accounts for both factual consistency and relevance of the generated text. It combines a precision-oriented framework from Wang et al. [127] and a recall-oriented framework from Scialom et al. [115].

Both frameworks consist of a Question Generation component and a Question Answering component. Both components used a pre-trained T5 model [107] which was fine-tuned on two question-answering datasets. It is important to note that the context length of T5 is 512 tokens long. This can cause difficulties in the evaluation process because summaries are often longer than 512 tokens. For the Precision framework questions were generated using the summary and they were answered using the source document. The Recall framework generated questions from the source document and answered these questions using the summary. The generated questions and the generated answers were weighted to account for the fact that an effective summary should contain the most important information from the source. Both frameworks gave scores as output which QuestEval uses to calculate the F-score. Results showed that QuestEval correlates much better to human judgment than metrics such as BERTScore, ROUGE, and other Question Answering evaluation metrics on the SummEval [40], QAGS-XSUM [127] and QAGS-CNN [127] datasets. These three datasets are often used to evaluate the correlation of an evaluation metric with human judgment. SummEval contains summaries from the CNN/Daily Mail [94] dataset, which was annotated by experts along four criteria; Consistency, Coherence, Fluence and Relevance. QAGS [127] is a dataset containing either summaries created by BART [76] on XSUM [95] (QAGS-XSUM) or CNN/Daily Mail [94] (QAGS-CNN). Annotators measured the factual consistency of the summary, corresponding to the consistency criterion of SummEval.

UniEval [145] is a unified multi-dimensional evaluator for text generation tasks. It formulates the multi-dimensional evaluation task as a unified Boolean Question-answering (QA) problem. This enables UniEval to evaluate a generated text using just a single model. UniEval uses T5 [107] as its backbone model for the QA Task. The generated text, the reference source text, and a context are fed to the backbone model. The generated text is then evaluated along four criteria; Coherence, consistency, fluency, and relevance. Given the input and a question involving one of the four criteria, the backbone model will answer "Yes" or "No". UniEval has shown an improved correlation with human judgment on SummEval [41] over *n*-gram-based approaches, embedding-based approaches, and BARTScore.

G-Eval [83] is a prompt-based evaluator of three components.

(1) A prompt that contains the definition of the evaluation task and the desired evaluation criteria
(2) A chain-of-thoughts (CoT) sequence of intermediate instructions generated by the LLM describing the detailed evaluation steps
(3) A scoring function that calls the LLM with the designed prompt, CoT, the input text, and the target text that needs to be evaluated. This function returns the score of the generated text, which is rated along the desired evaluation criteria.

G-Eval uses GPT-4 [98] as the backbone model. The advantage of using GPT-4 is that a longer context length of 4096 tokens can be used. A disadvantage is that GPT-4 is not an open-source model. On SummEval [40], QAGS-CNN [127], QAGS-XSUM [127] and Topical-Chat [47] G-Eval showcases a better correlation with human judgments compared to ROUGE, BERTScore, MOVERScore, and BARTScore. G-Eval is also used in combination with GPT-3.5 which has mixed results when comparing it to previously mentioned metrics. Table 6 lists all the evaluation metrics, mentioned in this section. If available, Table 6 also lists the correlations of the evaluation metric with human judgment on the SummEval dataset.

3 METHOD

This section will discuss the various methods that this research uses to construct the proposed architecture. Section 3.1 describes the properties of the dataset used in this research. Section 3.2 discusses different chunking operations that this research proposes. Section 3.3 is divided into three different subsections, providing a more detailed insight into the total architecture. Section 3.4 will describe the metrics used to evaluate the final summary F. Figure 2 is an intuitive visualisation of the summarization process proposed by this research. The process includes two

types of summarization steps, defined as the extractive step and the abstractive step. The extractive step involves the chunking of the document and the generation of an intermediary summary by an extractive summarization model. The abstractive step involves the generation of the final summary by an abstractive summarization model. The extractive step was repeated until the intermediary summary was under some threshold, after which the final summary was generated by the abstractive step.

To formulate the proposed architecture and the task, a source document D is composed as $D = \{D_1, D_2, D_3, ...D_m\}$, where D_i represents one chunk of the source document D. The source document D is composed of m chunks. A chunk is summarized by an extractive summarization model, which produces an intermediary summary $E^j = E_1^j \oplus E_2^j \oplus E_3^j \oplus, ..., \oplus E_M^j$, where E_i^j represents an intermediary summary of chunk D_i of extractive step j. The intermediate summary E^j is made up of the summaries of all the chunks, which are concatenated in the same order as in the original text. The extractive summarization model has a compression ratio R, that can be set between $[0, 1]$. This research defines the compression ratio by dividing the length of an article by its corresponding summary, like Grusky et al. [49]. Before the summarization is performed, the amount of extractive steps N that was taken is calculated using Formula 5. Then, the N extractive summarization steps are undertaken, and for the next extractive summarization step, the intermediary summary is again summarized, resulting in $D = E^j$ and $j = j + 1$. After all N steps have been completed, the final summary F was created by an abstractive summarization model. A visualisation of the formalization can be seen in Figure 7. All the formulations can be concluded as follows:

- $\boldsymbol{D}$: The source document. It is composed as:
  $D = \{D_1, D_2, D_3, ...D_m\}$, where D_i represents one chunk of source document D and D contains m chunks.
- $\boldsymbol{E^j}$: The intermediate summary of iteration j created by the extractive summarization model. It is composed of: $E^j = E_1^j \oplus E_2^j \oplus E_3^j \oplus, ..., \oplus E_M^j$, where E_i^j represents an intermediary summary of chunk D_i and E^j is composed of m summaries.
- $\boldsymbol{N}$: The amount of extractive steps.
- $\boldsymbol{F}$: The final summary created by the abstractive summarization model.
- $\boldsymbol{K}$: The context length of the abstractive summarization model.
- $\boldsymbol{R}$: The compression ratio of the extractive summarization model, ranging between $[0, 1]$.

This research explores the potential benefits of using one or multiple extractive steps before creating the final summary using a single abstractive summarization step. This comparison and investigation addresses the RQ 1 and RSQ1 1. Additionally, the study investigates the impact of using a domain-specific legal LM on the final summary's quality. This research compares situations where a legal LM is utilized against situations where no legal-specific LM is employed, addressing RSQ2. Because long documents are handled, the context length of the models is very important. Therefore, the impact of the context length of the extractive and abstractive models is also researched, addressing RSQ3 and RSQ4.

3.1 Dataset

Dataset characteristics. The dataset used to fine-tune the abstractive model is the EUR-Lex-Sum dataset [4]. This dataset consists of documents from the European Union law platform with corresponding manually curated summaries. The dataset addresses limitations in existing summarization datasets by diversifying content domains beyond news articles and wiki-like texts and enhancing language inclusivity through the incorporation of multilingual data. For this task, only the English part of the dataset, consisting of 1504 document/summary pairs was used. It has been divided into a training, validation, and test set, containing 1129 pairs, 187 and 188 pairs respectively.

In Figure 8 the distribution of tokens of the English training set can be seen. In Table 7 the dataset properties of EUR-Lex-Sum [4] can be seen.

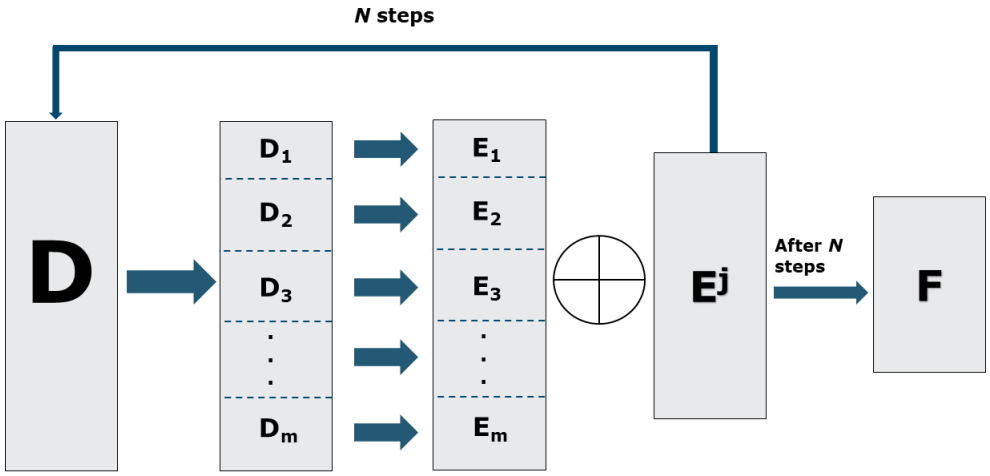
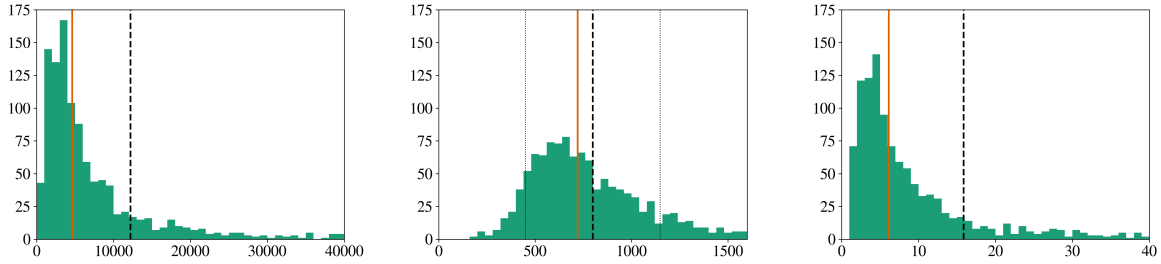


Fig. 7. Visualisation of the summarization process. N represents the amount of extractive steps and the $\oplus$ symbol represents the concatenation process.



(a) Reference tokens. (b) Summary tokens. (c) Token compression ratio.

Fig. 8. Histograms of the English training set of EUR-Lex-Sum [4], directly taken from from Aumiller et al. [4]. They showcase the token lengths of reference texts, summaries, and the compression ratio. The orange line represents the median length, the dashed black line represents the mean length and the dotted black lines represent the standard deviation.

An example of a summary from the dataset can be seen in Appendix B. The summaries have a clear structure. Each summary begins with a mention of the main legislative documents that establish and govern the respective programmes. This is followed by an outline of the aim or purpose of the regulation, along with any related decisions, highlighting the primary objectives and scope. The key points of the programme are then detailed, specifying the main objectives and areas of focus. In this part, summaries can provide additional information on topics such as governance structures, funding, and the roles of various agencies involved in the implementation. Finally, summaries conclude with the application date of the regulations and offer background resources and

Table 7. EUR-Lex-Sum [4] dataset properties. All values are taken from the English subset. The table is taken directly from Aumiller et al. [4]

	Reference tokens		Summary tokens		Comp.	% novel n-grams in summary			
	Min	Max	Min	Max	Ratio	1-gram	2-gram	3-gram	4-gram
Train	385	1,087,217	173	3021	16 ± 62	44.10	65.97	78.85	84.96
Val.	1,143	199,405	354	5136	18 ± 17	36.65	58.23	72.74	79.96
Test	1,544	403,319	369	2987	18 ± 20	36.78	58.46	72.83	80.07

references to related documents for further information. This approach offers readers a consistent structure across all approaches.

Outlier removal. Outlier removal was performed to address the presence of extreme outliers in the EUR-Lex-Sum dataset, as noted by Aumiller et al. [4]. Excessively long summaries are marked as outliers and removed. To ensure consistency and accurate evaluation, the words in each summary are counted, and those with word counts exceeding two standard deviations above the mean were removed. This research opted to perform outlier removal based on words instead of tokens. When tokens would have been used, the number of outliers removed would differ per model as models utilize different tokenization methods. The outlier removal results in a total of 62 instances being removed. The training set size was reduced from 1504 to 1129 to 1091, the validation set from 187 to 172 and the test set from 188 to 179.

3.2 Chunking the Document

Before the intermediary summary E^j can be created by the extractive summarization model, the document will be chunked into smaller segments. Chunking is necessary because the summarization models have a limited context length. Different context lengths of the extractive and abstractive models can be seen in 8 and Table 9 respectively. Each chunk was then summarized separately by the extractive summarization model. The size of the chunk is therefore dependent on the context length of the extractive and abstractive model. The summaries are concatenated for either another extractive step or the final abstractive step, which produces the final summary. For this research, Fixed-Size chunking with overlapping was used.

3.3 Architecture

This subsection will discuss the general architecture proposed by this research. A visualisation of the proposed architecture can be seen in Figure 7. The proposed architecture consisted of one, and possibly more, extractive steps, followed by a final abstractive step. This subsection is set up in line with the structure of the architecture. Section 3.3.1 provides a detailed description of the extractive step, including various compression ratio strategies. Section 3.3.2 elaborates on the specifics of the abstractive step.

3.3.1 Extractive Step. The extractive step in the proposed summarization process involves calculating the amount of extractive steps that are required, to break down the source document D into smaller chunks D_i, generating an extractive summary E_i for each chunk and then concatenating the summaries to form the intermediate summary E^j. This process was repeated until the amount of extractive summarization steps N had been reached.

This subsection will discuss the several aspects of the extractive step. First, the three different extractive compression ratio strategies will be discussed. Then a variety of models will be discussed which can be implemented for the extractive summarization of the chunks.

Extractive compression ratio strategies. Three different compression ratio strategies were used for the extractive step; fixed, dependent, and hybrid. The fixed ratio uses N amount of extractive steps with a fixed extractive

compression ratio. The dependent ratio always performs one extractive step, with the extractive compression ratio being dependent on the size of the document and the context length of the abstractive model. The hybrid ratio performs $N-1$ extractive steps with a fixed ratio, but the final extractive step N uses a dependent ratio.

The comparison between the fixed and the hybrid ratio against the dependent ratio is interesting because it checks the effectiveness of one extractive summarization step compared to multiple extractive summarization steps. The hybrid ratio could potentially be more effective than the fixed ratio because it is focused on ensuring that the final intermediary summary optimally fits the context length of the abstractive model. Researching the ratio type addresses the RQ and RSQ1 and enables this research to find the optimal extractive model.

Fixed ratio. The fixed ratio used one fixed compression ratio that was used for all extractive summarization steps for all documents. The fixed ratio was set to 0.4, which implies that 40% of the sentences of a text will be used for an extractive summary. When the fixed ratio was used, the amount of extractive steps was calculated before the extractive summarization was performed. The amount of extractive steps can be estimated by using the token length of the document and the context length of the abstractive summarization model. The estimation for the amount of extractive summarization steps will be performed individually for each document because the amount of tokens varies greatly per document, as can be seen in Table 7. The calculation of the amount of extractive summarization steps has been broken down into the following steps:

(1) The length of the intermediary summary $|E^j|$ after the first step is $R \cdot |D|$. After the second step, it is $R^2 \cdot |D|$ and so on. This implies that the length of the intermediary summary after N steps is:

$$|E^N| = R^N \cdot |D|$$

(2) Extractive steps are performed until the length of the intermediary summary is within the context length of the abstractive summarization model, K:

$$R^N \cdot |D| \leq K$$

(3) To estimate N, take the logarithm on both sides:

$$N \cdot \log(R) \leq \log\left(\frac{K}{|D|}\right)$$

(4) Then, solve for N:

$$N \leq \frac{\log\left(\frac{K}{|D|}\right)}{\log(R)}$$

(5) N is then rounded up to the highest integer. So, the formula for estimating the number of extractive steps N needed before the final abstractive step can be taken is:

$$N = \left\lceil \frac{\log\left(\frac{K}{|D|}\right)}{\log(R)} \right\rceil \tag{5}$$

N is rounded up to the highest integer because N needs to be an integer. If N is rounded down, then the length of the source document $|D|$ will exceed the context length of the abstractive model K. If a document already fits in the context length of the abstractive model K, N will be set to zero as no extractive steps need to be taken.

Dependent ratio. When the dependent ratio was used, only one step of extractive summarization was performed. The dependent ratio uses an extractive compression ratio which is dependent on the context length of the abstractive model K and the token length of the source document $|D|$. The dependent ratio is defined as the ratio that needs to be used to compress a document in such a manner that it fits the context length of the abstractive summarization model in one extractive summarization step. The formula for the dependent ratio is as follows, where R_d is the dependent ratio:

$$R_d = \frac{K}{|D|} \tag{6}$$

Hybrid ratio. The hybrid ratio combined both the fixed and dependent ratios. Initially, a fixed ratio of 0.4 is applied for the first $N-1$ steps. The number of extractive steps required was determined in the same way as for the fixed ratio, as detailed in Equation 5. However, in some cases, using a fixed ratio for step $N-1$ may result in an intermediary summary that is smaller than the context length of the abstractive model, which is suboptimal because the context length should be utilized as fully as possible. To address this, a dependent ratio was used for step N, as shown in Equation 6. This approach ensured that the final intermediary summary fit the context length of the abstractive model optimally. The hybrid ratio can be expressed as follows:

$$R = \begin{cases} 0.4 & \text{for steps } 1, 2, \ldots, N-1 \\ \frac{K}{|D|} & \text{for step } N \end{cases} \tag{7}$$

Models. Domain-specific legal LMs and non-domain-specific LMs were tested to address RSQ1. Non-domain-specific LMs will be referred to as 'general' LMs. By using both types of LMs in the extractive summarization step, a comparison was made to determine whether the quality of the final summary was improved by the additional knowledge and expertise of a legal LM. Additionally, two models with a longer context length were used to address RSQ2. Table 8 lists all the extractive summarization models that this research will use.

Table 8. Extractive summarization models used. The context length is expressed in tokens.

Model	Legal LM	Context length
RoBERTa [84]	✗	512
Longformer [8]	✗	4096
LegalBERT-SC [17]	✓	512
LexLM [20]	✓	512
LexLM - Longformer [20]	✓	4096

Every extractive summarization model was combined with the three ratio types to process the dataset. This resulted in fifteen different models to be fine-tuned. Subsequently, the baseline abstractive model BART [76] was fine-tuned using the results from the extractive models. All fine-tuned BART models were evaluated using metrics outlined in Section 3.4. A baseline comparison was also created by fine-tuning the baseline BART [76] model without any extractive steps. So, in total sixteen model configurations were compared. Testing all possible combinations of extractive models, ratio types, and abstractive models was not undertaken because it would result in over 150 different model configurations, which is not feasible or practical given the resource capabilities. Therefore, the extractive summarization model and the ratio type that yielded the highest scores on these evaluation metrics for the fine-tuned BART model were considered the optimal extractive summarization model. The optimal extractive model was used to fine-tune all abstractive models. RSQ1, RSQ2, and RSQ3 can be addressed by examining the results obtained from different combinations of ratio types and extractive models.

3.3.2 Abstractive step. The abstractive step was only performed once when the length of the intermediary summary $|E^j|$ is shorter than the context length of the abstractive summarization model K. The abstractive step involves creating the final summary F by a fine-tuned abstractive summarization model based on the intermediary summary E^j. Research and testing on the abstractive step address the RQ and RSQ4. This subsection will discuss several aspects of the abstractive step. First, a selection of models will be discussed which can be implemented for the abstractive summarization of the intermediary summary E^j. Finally, the fine-tuning procedure of the abstractive summarization model will be described.

Models. A selection of abstractive summarization models was considered for the abstractive step. Table 9 has listed all the models, their corresponding context length, and their architecture. The context length of the abstractive summarization model is an important consideration as it affects the number of extractive steps. A longer context length implies that fewer extractive steps need to be taken, which could improve the understanding of the global context of the source document by our proposed architecture. For this comparison, T5 [107] were compared against LongT5 [50] and Pegasus [139] were compared against PegasusX [102]. Both model pairs mostly differ in the attention mechanisms that they use, but LongT5 and PegasusX were also pre-trained with additional long-context tasks. This comparison will answer RSQ4. Besides researching what abstractive model performs best, it was interesting to research what effect one or multiple extractive steps have compared to not using any extractive steps. This comparison was made by evaluating each abstractive model without any extractive steps to an abstractive model using the optimal extractive model, addressing the RQ.

Table 9. Abstractive summarization models used. The context length is expressed in tokens.

Model	Context length	Architecture
BART[76]	1024	Encoder-decoder
T5 [107]	512	Encoder-decoder
LongT5 [50]	16384	Encoder-decoder
Pegasus [139]	1024	Encoder-decoder
PegasusX [102]	16384	Encoder-decoder
Llama3 [2]	8192	Decoder-only

Data preparation. The model architecture was of importance for how the data is preprocessed after the extractive summarization has been performed. For encoder-decoder models, the reference texts and summaries were given to the model as corresponding input-output pairs. During the fine-tuning process, the model learns how to summarize the long, regulatory documents by using the input-output pairs as a guide.

For decoder-only models, such as Llama3 [2], the input data must be processed differently to fine-tune the model. Instead of passing separate input and output sequences, only a single input sequence is passed. The decoder-only model will then learn to predict this sequence of tokens, given a text. So, the reference text and the summary were combined into a single input sequence to fine-tune the model. Additionally, a prompt was provided to guide the model during the summarization process. The input sequence to fine-tune the decoder-only model looks as follows:

```
Summarize the following text.
### Text:
{reference text}
### Summary:
{golden reference summary}
```

During testing and inference, the golden reference summary was not appended, forcing the model to generate a summary based on the reference text. Because the entire input sequence needs to fit into the context length, balancing the lengths of the reference text and the summary is crucial. By subtracting 1500 from the context length of the abstractive model, enough space was left for a summary of 1500 tokens. The maximum amount of generated tokens for one text had been set to 1500, as can be seen in Table 10. For Llama3, this implied that the effective context length is 6692 as 1500 is subtracted from its context length of 8192 tokens.

Instead of just truncating the intermediary summaries to 6692 tokens, the extractive summarization process was adjusted such that the reference text is summarized until 6692 tokens. Therefore, the effective token length for the reference text was reduced to 6692 tokens, ensuring that the model could process the combined input with minimal truncation. For the decoder-only models, the input sequence was cut off during evaluation on the test set, to enable a fair comparison between the golden-reference summaries and the predicted summaries.

The abstractive models have been fine-tuned using two or four NVIDIA A40 GPUs. Because of resource limitations, not all possible combinations of ratio types, extractive summarization models, and abstractive summarization models can be tried. After the optimal ratio type and extractive summarization model have been found, this combination will be used to research what abstractive model performs best.

Full parameter fine-tuning . The training set of the EUR-Lex-Sum dataset [4] will be used to fine-tune the abstractive summarization model of the proposed extractive-abstractive architecture. If the combination of the abstractive model and the hardware configuration is sufficient, the abstractive model is fully fine-tuned, as prior work shows that this is most effective [9]. Full parameter fine-tuning implies that every parameter in a model will be adjusted during the training process. In table 10 the hyperparameters can be seen that are used to fine-tune the abstractive models.

Table 10. Hyperparameter settings.

Hyperparameter	Setting
Learning rate	$5e^{-05}$
Epochs	40
Effective batch size	16
Warmup ratio	0.1
Weight decay	0.01
Early stopping patience	5
Metric for best model	Validation loss
Maximum generation length	1500

Early stopping was used with a patience of five epochs, monitoring the validation loss. If the validation loss failed to improve for five consecutive epochs, the training process was stopped to save resources and avoid overfitting. Additionally, the model with the lowest validation loss was retained as the final model. The maximum generation length of the models was set to 1500 tokens, allowing the model to generate up to 1500 tokens for the summary.

QLoRA. Llama3 [2] was fine-tuned using QLoRA [32], as full parameter fine-tuning was not feasible due to its size. QLoRA uses Low-Rank Adaptation (LoRA)[58] to fine-tune a **Q**uantized pre-trained LLM. LoRA and QLoRA are a form of parameter efficient fine-tuning (PEFT), a technique used for fine-tuning large neural models by adjusting only a limited number of parameters while other parameters are left unchanged. LoRA and QLoRA make training more efficient and more accessible because the hardware requirements are lower than full parameter fine-tuning.

Full parameter fine-tuning focuses on retraining all model parameters, which requires significant computational resources because modern LLMs have a substantial amount of parameters, for example; GPT-4 [98] has around 1.7 trillion parameters, Llama3-70B [2] has 70 billion parameters and Mixtral-8x7B[65] has over 46 billion parameters.

Instead of updating the entire high-dimensional weight matrix, LoRA inserts trainable low-rank matrices into specific layers of the neural network. During fine-tuning, the matrices track the changes to the original weights while the original weight matrices remain frozen. By using low-rank matrices, the number of parameters that need to be updated is significantly reduced, making the fine-tuning process more efficient in terms of both computation and memory usage. After the fine-tuning process, the low-rank matrices can be merged with the frozen weights to implement the adjustments made by the fine-tuning process. Figure 9 depicts the process of LoRA fine-tuning. It shows a pre-trained weight matrix W of dimensions $d \times d$. During fine-tuning, the low-rank matrices, A and B, are introduced. They have rank r which is smaller than dimension d of the original layer, reducing the computational requirements. After training, A can be multiplied with B, to create a matrix of similar size to W with dimensions d. All the weight changes can then be merged into the original frozen weights to implement the changes learned during fine-tuning.

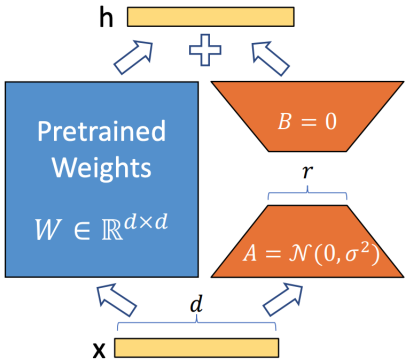


Fig. 9. LoRA reparametrization, directly taken from Hu et al. [58]

QLoRA is an extension of LoRA and uses a quantized version of the pre-trained model. Quantization reduces the precision of the model weights, which further decreases memory usage and computational requirements. Using an fp32 precision is optimal as it is the most precise but also the most memory-intensive. Quantization can reduce the precision to 8-bit or even 4-bit integers, which significantly lowers memory usage and computational load.

While LoRA and QLoRA significantly reduce the computational and memory requirements for fine-tuning large models, it is often less effective than full parameter fine-tuning [9]. However, the difference in effectiveness differs per domain, dataset, task, and parameter settings as can be seen in works from Biderman et al. [9] and Hu

et al. [58]. Additionally, quantization techniques can introduce quantization noise and reduce numerical precision, potentially impacting model accuracy [32].

Fully Sharded Data Parallel (FSDP) [142] was leveraged to fine-tune Llama3, making use of the Hugging Face implementation. As these implementations are relatively new, some issues arose during training when combining FSDP and QLoRA, which have been reported to the Hugging Face team. Therefore, the best-performing model could not be loaded at the end. Consequently, the early stopping patience and the best model metric were not set. To mitigate overfitting, 10 epochs were used instead of 40. The number of epochs for Llama3 was based on preliminary results, showcasing that models converge around 4-20 epochs. To accommodate for QLoRA, low-rank matrices are injected into the query, key, and values matrices, and the linear layers of Llama3. Additional QLoRA settings are based on prior research [108] [58] and can be found in Table 11.

Table 11. Llama3 settings.

Hyperparameter	Setting
Learning rate	$5e^{-05}$
Epochs	10
Effective batch size	16
Warmup ratio	0.1
Weight decay	0.01
Early stopping patience	-
Metric for best model	-
LoRA rank (r)	8
LoRA alpha	16
LoRA dropout	0.1
Precision for frozen model weights	4-bit NormalFloat
Precision for low-rank matrices	bfloat16
Precision for calculations	bfloat16
Double Quantization	True

3.4 Evaluation

After fine-tuning the abstractive summarization model, the proposed architecture was evaluated using the test set of EUR-Lex-Sum [4]. The results of the proposed architecture were compared against golden-reference summaries from the test set. The combination of evaluation metrics included multiple types of approaches to ensure that the proposed architecture is evaluated from different aspects. Table 12 lists all the different evaluation metrics that this research has used. Besides automated metrics, this research has also performed a small-scale human evaluation. The human evaluation provides insights into the quality of the summaries, complementing the quantitative data from automated metrics with qualitative feedback. However, the primary focus of the evaluation relies on automated evaluation metrics, as this approach is more commonly used in NLG research than human evaluation. Additionally, the human evaluation was conducted on a small scale, so it cannot provide definitive answers. Including more evaluators would be necessary for conclusive results. All automated evaluation metrics are described in Section 3.4.1, and the human evaluation procedure is described in Section 3.4.2.

3.4.1 Automated evaluation metrics. The HuggingFace evaluate library was used for the implementation of ROUGE, which implements ROUGE as introduced by Lin et al. [79]. This research uses ROUGE-1, ROUGE-2, and

Table 12. Evaluation metrics used in this research.

Evaluation Metric	Type of approach
ROUGE [79]	*n*-gram-based
BERTScore [139]	Embedding-based
BARTScore [136]	Generation Probability-based
BLANC [124]	Referenceless
Domain expert evaluation	Human evaluation

ROUGE-L to compare the predictions against the golden reference summaries. This research uses the F-score of the ROUGE metric.

BERTScore [139] also leverages the HuggingFace evaluate library. Different encoder-only models can be used for evaluating the predictions. This thesis made use of the Longformer [8] architecture when using BERTScore [136]. Longformer has a long context length, which enables it to evaluate the entire summary without having to the summary. This feature is important as many summaries can be longer than the context length of models such as RoBERTa [84], see 8b. In this thesis, the F-score, seen in Equation 4, was used for BERTScore.

Stanford's string2string library was used to implement the BARTScore [136]. The BART version used for the evaluation was BART[76] fine-tuned on the CNN/Daily Mail [94] dataset. By calculating the log-likelihood of the probability that the prediction is generated given the golden reference summary, the precision is obtained. By calculating the log-likelihood of the probability that the golden reference summary is generated given the prediction, the recall is calculated. By combining these two values in Equation 4, the F-score can be calculated, which was used in this research. BART [76] has a context length of 1024 tokens. This limited context is a limitation of BARTScore as summaries longer than 1024 tokens cannot be fully evaluated.

BLANC-help [124] was implemented using the BLANC package, which is available on GitHub. A gap of two was used in this research, as research has shown that this correlates the most with human evaluation [125]. It must be noted that the BLANC score suffers from the same limitation as BARTScore; its context length. BLANC makes use of the BERT base model [33] to calculate the BLANC scores, meaning that it only has a context length of 512 tokens. The BLANC scores were calculated by concatenating the predicted summaries to each sentence from the source document. But a sentence plus the entire predicted summary often exceeds the context length of 512 tokens. Valuable information at the end of the summary was truncated, potentially resulting in a skewed BLANC score.

3.4.2 Human evaluation. Besides the proposed evaluation metrics, this research made use of evaluation by domain experts of Power2X. The evaluation by domain experts can give insights into the quality of the final summary but it was not performed at a large scale. The qualitative evaluation by domain experts adds a layer of depth and domain-specific understanding. Their feedback was used for identifying nuances or domain-specific intricacies that might be left out by only using automated metrics.

Various summarization models were employed to assess their performance. The baseline model used was BART without an extractive step. Additionally, BART was evaluated in combination with the optimal performing extractive summarization model. Another model included the best-performing legal LM combined with BART. The evaluation also included the best-performing long context length extractive model paired with BART, the best-performing long context abstractive model, and finally, the best decoder-only variant. These diverse models were chosen to understand the effectiveness of the different approaches in summarization tasks. The chosen models also reflect the research questions and provide additional information besides the automated evaluation metrics.

Table 13. Criteria for human evaluation.

Criterion	Description
Factual Correctness	Evaluation of how factually correct the summary is relative to the source document.
Usability	Assessment of how practical and user-friendly the summary is.
Accuracy	Assessment of the precision and correctness of the information in the summary.
Fluency	Assessment of the summary's smoothness and ease of reading in terms of form, content, and grammar.
Coherence	Measure of how logical the summary is to it is linguistic context.

The generated summaries were evaluated based on several key criteria, as outlined in Table 5. This research has chosen to adjust Correctness to Factual Correctness to check if the summaries contain false statements. The criteria used in this research for the human evaluation can be found in Table 13. Each summary was rated on a scale of 1 to 5, where 1 indicates poor performance and 5 indicates excellent performance. Participants were also allowed to provide additional feedback for each summary.

After selecting the optimal extractive model and training all abstractive models, an out-of-dataset text was used for evaluation. The Carbon Border Adjust Mechanism document [39] was selected to test the models' performance at inference because the evaluators were familiar with regulatory documents in general and the global context of this specific document. The summaries were assessed using the evaluation form provided in Appendix B. The evaluation process was conducted independently, without any supervision from the researchers, to ensure unbiased feedback.

4 RESULTS AND EVALUATION

This section provides an overview of the results obtained in this research. The section has been split up into Section 4.1, Section 4.2, and Section 4.3. Section 4.1 describes the results of the research questions that are correlated to the extractive models. It has been divided into Sections 4.1.1, 4.1.2, 4.1.3 and 4.1.4, where each section gives a precise overview of the results correlated to a specific research question. Section 4.2 reports the results of the research questions, correlated to the abstractive models. It has been split up into Section 4.2.1 and Section 4.2.2, to provide results for the research questions. Section 4.3 presents the findings of the human evaluation on an out-of-dataset text. Section 4.3.1 details the scores on the five evaluation criteria. Lastly, Section 4.3.2 discusses additional comments on the summaries.

4.1 Comparison extractive models

Table 14 contains the results of all extractive models on ROUGE-1, ROUGE-2, ROUGE-L, BERTScore, BARTScore, and BLANC. It can be seen that RoBERTa with a dependent ratio scores the highest on ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore. Not using any extractive steps, results in the highest on BARTScore and BLANC.

4.1.1 Effect of number of stages. **The results indicate that models using the dependent ratio type generally achieve higher performance across most metrics.** Notably, the RoBERTa model with the dependent ratio type attains the highest scores in ROUGE-1 (0.4873), ROUGE-2 (0.1974), ROUGE-L (0.2247), and BERTScore (0.8721), suggesting superior performance in these areas. However, the BART model without any extractive steps achieves the best scores in BARTScore (-3.4154) and BLANC (0.1700), indicating a stronger performance in these specific metrics despite not utilizing extraction.

In contrast, models with the fixed and hybrid ratio types show varying results, with generally lower scores than the dependent type but still competitive. For instance, the LexLM model with the dependent ratio type also performs well, achieving high scores close to RoBERTa's best metrics. Overall, the dependent ratio type

Table 14. Evaluation results of all extractive models with all ratio types, fine-tuned on BART. When no extractive model is used, this is showcased with '-'.

Extractive model	Ratio type	Legal LM	Context Length	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BARTScore	BLANC
-	No extraction	✗	-	0.4590	0.1954	0.2174	0.8702	**-3.4154**	**0.1700**
RoBERTa	Fixed	✗	512	0.4670	0.1798	0.2171	0.8692	-3.5654	0.1322
RoBERTa	Dependent	✗	512	**0.4873**	**0.1974**	**0.2247**	**0.8721**	-3.5590	0.1541
RoBERTa	Hybrid	✗	512	0.4809	0.1889	0.2193	0.8700	-3.5781	0.1487
LegalBERT	Fixed	✓	512	0.4390	0.1766	0.2158	0.8700	-3.4893	0.1427
LegalBERT	Dependent	✓	512	0.4619	0.1854	0.2174	0.8713	-3.5143	0.1463
LegalBERT	Hybrid	✓	512	0.4469	0.1774	0.2137	0.8665	-3.5714	0.1423
LexLM	Fixed	✓	512	0.4571	0.1745	0.2123	0.8692	-3.6130	0.1350
LexLM	Dependent	✓	512	0.4859	0.1954	0.2227	0.8713	-3.5441	0.1543
LexLM	Hybrid	✓	512	0.4582	0.1792	0.2135	0.8665	-3.5639	0.1421
Longformer	Fixed	✗	4096	0.4436	0.1686	0.2103	0.8684	-3.5901	0.1270
Longformer	Dependent	✗	4096	0.4613	0.1874	0.2194	0.8712	-3.5835	0.1503
Longformer	Hybrid	✗	4096	0.4778	0.1862	0.2181	0.8703	-3.5697	0.1460
LexLM-Longformer	Fixed	✓	4096	0.4250	0.1584	0.2041	0.8659	-3.6141	0.1258
LexLM-Longformer	Dependent	✓	4096	0.4751	0.1852	0.2164	0.8689	-3.5344	0.1419
LexLM-Longformer	Hybrid	✓	4096	0.4619	0.1819	0.2189	0.8692	-3.5833	0.1417

appears to provide a slight performance advantage across different models, although the differences between the ratio types are often minimal. The visualisation of these results would further highlight the small variations in performance metrics across different model configurations.

To further investigate the difference between the ratio types, all model variants were grouped based on their ratio types: fixed, dependent, and hybrid. The performance metrics for each model variant were then averaged within each group. This grouping allowed for a comparison of the average performance of models using different ratio modes. Table 15 presents the mean and standard deviation of every ratio mode for all evaluation metrics. A visualisation of these results can be found in Appendix D using bar graphs.

Table 15. Average evaluation results for models with the same ratio mode.

Ratio mode	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BARTScore	BLANC
Fixed	0.4463 (± 0.0163)	0.1716 (± 0.0084)	0.2119 (± 0.0051)	0.8686 (± 0.0016)	-3.5744 (± 0.0516)	0.1325 (± 0.0068)
Dependent	**0.4743** (± 0.0125)	**0.1902** (± 0.0058)	**0.2201** (± 0.0035)	**0.8709** (± 0.0012)	**-3.5470** (± 0.0260)	**0.1494** (± 0.0053)
Hybrid	0.4651 (± 0.0141)	0.1827 (± 0.0048)	0.2167 (± 0.0029)	0.8685 (± 0.0019)	-3.5733 (± 0.0076)	0.1442 (± 0.0031)

The grouping was only performed for the comparison of ratio types because, for every model, three ratio types were created, allowing for a fair comparison. Such grouping and comparison were not conducted when comparing general LMs against legal LMs, or models with different context lengths. The reason for this is the inherent variability and imbalance in the groups, which would lead to an unfair comparison. For example, the general LMs group includes models like RoBERTa and Longformer, while the legal LMS group includes models like LegalBERT, LexLM, and LexLM-Longformer. These models have different architectures, training data, and purposes, which introduces significant variability. Comparing these groups directly would not provide a fair

assessment of their performance due to the differences in their design and intended use cases. This same argument can also be made when comparing the models with different context lengths. Furthermore, the findings of this section can be summarized as follows:

- The results indicate that the dependent ratio mode generally achieves higher scores across all evaluation metrics compared to the fixed and hybrid modes.
- The differences in scores between the dependent, fixed, and hybrid modes are very small, suggesting that while the dependent mode performs better, the performance variations across the different modes are minimal.

4.1.2 Effect of Legal Language Models. **The results indicate that general LMs achieve slightly higher scores across all metrics except BARTScore, compared to legal LMs.** Table 14 presents the evaluation results for legal LMS and general LMs. The top five performing models for each metric can be seen in Table 16. Legal LMs outperform the general LMs in BARTScore. It is interesting however that not using any extractive steps achieves the highest BARTScore and BLANC score. Although the general LMs show a marginally better performance overall, the differences in scores between the two model types are minimal. The rest of this section is dedicated to an in-depth comparison of LegalBERT and LexLM against RoBERTa, and LexLM-Longformer against Longformer. This way, the comparison is accommodated for the context length of the models.

Table 16. Top five performing models for each metric. The ratio types are abbreviated where (F) represents the fixed ratio. (D) the dependent ratio and (H) the hybrid ratio. When no extractive model is used, it is shown using '-'.

Ranking	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BARTScore	BLANC
1	RoBERTa (D)	RoBERTa (D)	RoBERTa (D)	RoBERTa (D)	-	-
2	LexLM (D)	LexLM (D)	LexLM (D)	LexLM (D)	LegalBERT (F)	LexLM (D)
3	RoBERTa (H)	-	Longformer (D)	LegalBERT(D)	LegalBERT (D)	RoBERTa (D)
4	Longformer (H)	RoBERTa (H)	RoBERTa (H)	Longformer(D)	LexLM-Longformer (D)	Longformer (D)
5	LexLM-Longformer (D)	Longformer (D)	LexLM-Longformer (H)	Longformer(H)	LexLM (D)	RoBERTa (H)

ROUGE-1. For ROUGE-1, it can be seen that RoBERTa scores higher than both LexLM and LegalBERT when comparing every model with its ratio counterpart. BERT's highest score is 0.4873 using a dependent ratio. In comparison, LexLM's best ROUGE-1 score is 0.4859 with the dependent ratio, while LegalBERT's best score is 0.4619, achieved with the dependent ratio. When comparing LexLM-Longformer against Longformer, Longformer scores higher for the hybrid and fixed ratio types. If Longformer and LexLM-Longformer both use the dependent ratio, LexLM-Longformer however outperforms Longformer. Longformer's highest ROUGE-1 score is 0.4778 with the hybrid ratio, whereas LexLM-Longformer's best score is 0.4751 with the dependent ratio.

ROUGE-2. For ROUGE-2, RoBERTa again leads, with its highest score of 0.1974 in the dependent ratio configuration. LexLM with a dependent ratio follows closely with a top score of 0.1954. LegalBERT has a top score of 0.1854, achieved with the dependent ratio. When comparing Longformer against LexLM-Longformer, Longformer shows a slight edge with the highest ROUGE-2 score of 0.1874 in the dependent ratio. LexLM-Longformer's best ROUGE-2 score is 0.1852, also with the dependent ratio. For the remaining ratio types, Longformer also outperforms LexLM-Longformer.

ROUGE-L. Regarding ROUGE-L, RoBERTa has a top score of 0.2247 using the dependent ratio. LexLM follows with a score of 0.2227 in the dependent ratio, and LegalBERT scores highest at 0.2174 with the same ratio type. For the other ratio types, RoBERTa also seems to outperform its counterparts. Comparing LexLM-Longformer

and Longformer, Longformer again scores slightly higher, with its best ROUGE-L score being 0.2194 using the dependent ratio, while LexLM-Longformer's best score is 0.2189 using the hybrid ratio. When Longformer uses the fixed and dependent ratio it outperforms LexLM-Longformer, for the hybrid ratio LexLM-Longformer outperforms Longformer.

BERTScore. For BERTScore, RoBERTa achieves the highest score of 0.8721, LexLM matches closely with a score of 0.8713 and LegalBERT's top score is also 0.8713. These scores are all attained using the dependent ratio. RoBERTa outperforms LegalBERT and LexLM using the hybrid ratio. However, when using the fixed ratio, LegalBERT has a slightly higher score of 0.8700 than RoBERTa and LexLM, both having a score of 0.8692. When comparing Longformer against LexLM-Longformer, Longformer obtains its highest BERTScore of 0.8712 with the dependent ratio. LexLM-Longformer's highest BERTScore is 0.8692 using both the fixed and hybrid ratios. But Longformer also outperforms LexLM-Longformer when it uses these two ratios.

BARTScore. For BARTScore, the results are different than previous results. RoBERTa's best (least negative) score is -3.5590 with a dependent ratio. LexLM achieves a BARTScore of -3.5441 with the dependent ratio, while LegalBERT's highest BARTScore is -3.4893 in the dependent ratio. RoBERTa is outperformed by both legal LMs in all ratios, except for LexLM with a fixed ratio, showing that the legal LMS might have a better performance on BARTScore. When comparing LexLM-Longformer to Longformer, LexLM-Longformer performs better with a best score of -3.5344 in the dependent ratio. Longformer's top BARTScore is -3.5697, achieved with the hybrid ratio. Longformer outperforms LexLM-Longformer in the fixed and hybrid ratio types, however.

BLANC. Finally, for the BLANC metric, LexLM leads with its best score of 0.1543 in the dependent ratio. LexLM scores highest at 0.1541, also in the dependent ratio, and LegalBERT achieves its best score of 0.1463 with the dependent ratio. Longformer scores slightly higher than LexLM-Longformer with a best BLANC score of 0.1503 in the dependent ratio. LexLM-Longformer's highest score is 0.1419, also in the dependent ratio.
The findings of this section can be summarized as follows:

- General LMs tend to outperform legal LMs across text-similarity metrics such as ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore.
- Legal LMs show better performance than general LMs on BARTScore.
- The differences in the results between the General and legal LMs are small.

4.1.3 Effect of context length for the extractive step. **The results showcase that shorter context models achieve higher scores across all metrics.** In Table 16 it can be seen that for every metric except ROUGE-L, the top three exist out of short context length models. Table 14 presents the evaluation results for models with varying context lengths. The comparison focuses on shorter context models (RoBERTa, LegalBERT, LexLM) with a context length of 512 tokens versus longer context models (Longformer, LexLM-Longformer) with a context length of 4096 tokens. In Table 16 it can be seen that for every metric except ROUGE-L, the top-three ranking exists out of short context length models. The differences in scores between the two context lengths are minimal but consistent. The rest of this section is dedicated to comparing RoBERTa against Longformer and comparing LegalBERT and LexLM against LexLM-Longformer. This way, the comparison is accommodated for the model type by comparing egal LMs against each other and by comparing general LMs against each other.

ROUGE-1. For ROUGE-1, RoBERTa leads with the highest overall score of 0.4873 using the dependent ratio. Longformer, with a longer context length, achieves a top score of 0.4778 using the hybrid ratio. RoBERTa seems to outperform Longformer across all ratio types. Out of LegalBERT, LexLM, and LexLM-Longformer, LegalBERT has a ROUGE-1 score of 0.4619, LexLM achieves the best ROUGE-1 score of 0.4859 with the dependent ratio, slightly outperforming LexLM-Longformer, whose best score is 0.4751 with the same ratio. Also for the fixed

ratio, the shorter context length models show higher ROUGE-1 scores than LexLM-Longformer. For the hybrid ratio, it seems to be the other way around, as LexLM-Longformer outperforms LegalBERT and LexLM.

ROUGE-2. In ROUGE-2, RoBERTa leads with a top score of 0.1974 using the dependent ratio, outperforming Longformer, whose highest score is 0.1874 with the dependent ratio. RoBERTa consistently outperforms Longformer for all ratio types. LegalBERT achieves a score of 0.1854 with the dependent ratio, while LexLM reaches a top score of 0.1954 with the dependent ratio, slightly outperforming LexLM-Longformer, which scores 0.1852 with the same ratio. For ROUGE-2 scores, the same pattern as for ROUGE-1 scores arises; LegalBERT and LexLM achieve better scores using the fixed and dependent ratios, whereas LexLM-Longformer achieves a higher score using the hybrid ratio. It must be said that the differences are minimal.

ROUGE-L. For ROUGE-L, RoBERTa maintains the highest overall score of 0.2247 with the dependent ratio, compared to Longformer's top score of 0.2194 using the dependent ratio. RoBERTa performs better than Longformer across all three ratio types. LegalBERT's highest score is 0.2174 with the dependent ratio, while LexLM scores 0.2227 with the dependent ratio, outperforming LexLM-Longformer's best score of 0.2189 with the hybrid ratio. Interestingly enough, the same pattern arises for ROUGE-1 and ROUGE-2 scores. LegalBERT and LexLM achieve higher scores than LexLM-Longformer using the fixed and dependent ratio, while LexLM-Longformere outperforms the two short context length models using the hybrid ratio.

BERTScore. BERTScore results show RoBERTa achieving the highest score of 0.8721 with the dependent ratio, followed by Longformer's top score of 0.8712 with the dependent ratio. RoBERTa has a higher score than Longformer for the fixed ratio as well, but for the hybrid ratio, Longformer outperforms RoBERTa. LegalBERT and LexLM both achieve a score of 0.8713 with the dependent ratio, surpassing LexLM-Longformer's highest score of 0.8692 using both fixed and hybrid ratios. The same pattern emerges, where the short context length models obtain higher scores for the fixed and dependent ratios, while LexLM-Longformer has a higher score for the hybrid ratio.

BARTScore. For BARTScore, RoBERTa's best (least negative) score is -3.5590 with the dependent ratio, while Longformer's top score is -3.5697 with the dependent ratio. For the fixed and dependent ratio, RoBERTa seems to perform better than Longformer. But for the hybrid ratio, Longformer obtains a higher BARTScore than RoBERTa. LegalBERT achieves its highest BARTScore of -3.4893 with the fixed ratio, outperforming LexLM's best score of -3.5441 with the dependent ratio and LexLM-Longformer's top score of -3.5344 with the same ratio. For all other ratio types, LegalBERT outperforms LexLM-Longformer. LexLM obtains lower scores than LexLM-Longformer for all ratio types except the fixed ratio.

BLANC. Finally, for the BLANC metric, RoBERTa achieves its best score of 0.1541 using the dependent ratio, while Longformer's highest score is 0.1503 with the dependent ratio. RoBERTa also outperforms Longformer on the two remaining ratio types. LegalBERT achieves a top score of 0.1463 with the dependent ratio, whereas LexLM scores highest at 0.1543 with the dependent ratio, outperforming LexLM-Longformer, whose best score is 0.1419 with the same ratio. LegalBERT and LexLM also obtain higher BLANC scores for the fixed and hybrid ratios.
The results of this section can be summarized as follows:

- Shorter context models generally outperform longer context models on all metrics.
- For every metric except ROUGE-L, the top three performing models are entirely made up of short context length models.
- The best version of a short context length model will always outperform the best version of a long context model when comparing General and legal LMs.

4.1.4 Optimal extractive model. **Based on the evaluation results presented in Table 14, RoBERTa with a dependent ratio will be chosen as the optimal extractive model.** This model consistently performs the highest across various metrics, including ROUGE-1, ROUGE-2, ROUGE-L, and BERTScore.

It is important to note that the differences in performance metrics among the different models and configurations are very small. This minimal variation suggests that the extractive summarization step might not play a significant role in the overall performance. Additionally, the baseline mode without any extraction step performs remarkably well, outperforming all models in BARTScore and BLANC. This result could indicate that critical information might often be located at the beginning of the source documents, making extraction less impactful. Therefore, for every abstractive model, two versions will be created; one leveraging RoBERTa with a dependent ratio, and one without using any extractive step at all. This will also answer the RQ and showcase what the effect of an extractive step is.

4.2 Comparison abstractive models

The results for all abstractive models and their variants can be seen in Table 17. In general, BART and Llama3 score higher on ROUGE compared to the other abstractive models. All models score high on BERTScore, considering that it ranges from 0 to 1. T5, LongT5 Pegasus, and PegasusX obtain more competitive BARTScores than BART and Llama3. For BLANC, the scores differ per model.

Llama3, T5, LongT5, Pegasus, and PegasusX are prone to generate repeating n-grams in their summaries. This phenomenon is particularly pronounced in T5, LongT5, Pegasus, and PegasusX, where entire sentences are often repeated throughout the generated summary. This repetitive characteristic has implications for their performance across some metrics. The repetitive summaries likely contribute to lower performance on ROUGE scores, especially ROUGE-2 and ROUGE-L. This can be attributed to a decrease in the recall component of the ROUGE F-score (see Equation 4). When sentences are repeated, less unique content from the golden reference summary is likely to be included, leading to lower recall scores. Interestingly, the repetitive nature of these summaries may contribute to their relatively high BARTScores. This can be explained by the precision component (Equation 2) of the F-score. Summaries with many repetitions are likely to score higher on precision, as the repeated content is more likely to match parts of the reference summary. This increased precision can lead to a higher overall F-score, resulting in better BARTScores.

4.2.1 Effect of extractive step. **The impact of the extractive step varies across different models, but for most models, the inclusion of an extractive step harms their performance.** This is evident in the results for T5, LongT5, Pegasus, and PegasusX, where the versions without extraction tend to outperform their counterparts with an extractive step. BART presents a more varied picture as it differs per metric in which variant scores higher. Llama3 seems to benefit from an extractive step because the version with an extractive step outperforms the version without an extractive step on all metrics.

For every abstractive model, there are two fine-tuned versions available; one using an extractive step and one not using an extractive step. The results of these models can be found in Table 17.

The rest of this section compares the results of both versions of each model in-depth per evaluation metric. For clarity, models that incorporate an extractive step will be referred to by the name of the abstractive model. Models that do not use an extractive step will be denoted by appending "-NE" to the name of the abstractive model, where "NE" signifies "No Extraction."

BART. BART outperforms BART-NE on the ROUGE scores and BERTScore. However, BART-NE has a higher BARTScore and BLANC score than BART. The differences in performance between both models are subtle, however.

Table 17. Evaluation results of all abstractive models with and without an extractive step.

Abstractive model	Ratio type	Context length	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	BARTScore	BLANC
BART	No extraction	1024	0.4590	0.1954	0.2174	0.8702	-3.4154	0.1700
BART	Dependent	1024	**0.4873**	**0.1974**	0.2247	**0.8721**	-3.5590	0.1541
T5	No extraction	512	0.3033	0.1241	0.1994	0.8443	-2.1585	0.1426
T5	Dependent	512	0.2934	0.0926	0.1857	0.8404	-2.2234	0.1112
LongT5	No extraction	16384	0.3261	0.1309	0.2192	0.8497	-2.2195	**0.1745**
LongT5	Dependent	16384	0.2854	0.0969	0.0969	0.8444	-2.0423	0.1493
Pegasus	No extraction	1024	0.3305	0.1293	0.2260	0.8499	**-1.8067**	0.1592
Pegasus	Dependent	1024	0.3067	0.0911	0.2021	0.8435	-1.8940	0.1250
PegasusX	No extraction	16384	0.3673	0.1622	**0.2304**	0.8523	-2.4528	0.1666
PegasusX	Dependent	16384	0.3052	0.1162	0.1960	0.8413	-2.4305	0.1326
Llama3	No extraction	8192	0.4088	0.1816	0.2107	0.7854	-3.3424	0.1545
Llama3	Dependent	8192	0.4474	0.1885	0.2284	0.8687	-3.1268	0.1563

T5. T5-NE consistently outperforms T5 on all metrics. For ROUGE-1 and BERTScore, the differences are small. But when comparing ROUGE-2, ROUGE-L, BARTScore, and BLANC, it can be seen that T5-NE achieves higher scores. For ROUGE-2 and BLANC, the differences are by a larger margin.

LongT5. LongT5-NE outperforms LongT5 on all metrics except for BARTScore. Both LongT5 and LongT5-NE score fairly low on ROUGE-2 with respective scores of 0.0969 and 0.1309. LongT5 also underperforms heavily on ROUGE-L compared to other models, at its score is 0.0969, causing LongT5-NE to have a significantly higher score on ROUGE-L.

Pegasus. Pegasus-NE performs better than Pegasus on all metrics. Both versions of Pegasus score low on ROUGE-2; Pegasus-NE has a score of 0.1293 and Pegasus a score of 0.0911. Both versions perform well on BARTScore, with Pegasus versions achieving the overall highest BARTScores of all models. Pegasus-NE achieves a BARTScore of -1.8067 and Pegasus achieves a BARTScore of -1.8940.

PegasusX. PegasusX-NE surpasses PegasusX on all metrics except BARTScore. On the ROUGE metrics and BLANC, PegasusX-NE scores notably higher. PegasusX-NE attains a ROUGE-1 score of 0.3673, while PegasusX scores 0.3052. For ROUGE-2, PegasusX-NE has a score of 0.1622, in contrast to PegasusX's 0.1162. The ROUGE-L score for PegasusX-NE is 0.2304, exceeding PegasusX's 0.1960. PegasusX only outperforms PegasusX-NE on BARTScore, obtaining a score of -2.4305 compared to PegasusX-NE's BARTScore of -2.4528.

LLama3. Llama3 consistently outperforms Llama3-NE across all metrics. For ROUGE-1 and BERTScore, the differences are more substantial. Llama3 obtains a ROUGE-1 score of 0.4474 against Llama3-NE's ROUGE-1 score of 0.4088. The impact of extraction is especially notable on BERTScore, as Llama3 has a BERTScore of 0.8687 compared to Llama3-NE's BERTScore of 0.7854, which is the lowest BERTScore across all models. In some instances, Llama3-NE failed to generate any output, resulting in a BERTScore of 0 for those particular samples. This behaviour is likely due to the decoder-only architecture's reliance on autoregressive generation, which can sometimes struggle to initiate text generation because it has to generate the entire input sequence first. The extractive step appears to mitigate this issue by providing a starting point for the model, hence the improved performance of Llama3 with extraction across all metrics.
The findings of this section can be summarized as follows:

- The extractive step's effectiveness may depend on the model's specific architecture and pre-training approach.
- T5, LongT5, Pegasus, PegasusX perform better without an extractive step.
- BART shows mixed results from using an extractive step, with improvements in some metrics and decreases in others when using extraction.
- Decoder-only models like Llama3 seem to benefit from an initial extractive step.

4.2.2 Effect of context length for the abstractive step. **Results show that long context models generally outperform their short context counterparts, with some exceptions in specific metrics.** In this section T5 and Pegasus will be compared against their long context counterpart; LongT5 and PegasusX. This will be done by first comparing the short and long context variants with an extractive step against each other, and then by comparing the variants without an extractive step against each other. Finally, this section will also compare the results of using T5 and Pegasus with an extractive step against their long context counterpart without an extractive step.

T5 vs. LongT5. When comparing T5-NE against LongT5-NE, LongT5-NE outperforms T5-NE on all metrics except BARTScore. T5-NE obtains a BARTScore of -2.1585, whereas LongT5-NE obtains a BARTScore of -2.2195. In the comparison between T5 and LongT5, the results are varied. T5 performs better in ROUGE-1 and ROUGE-L, while LongT5 shows superior performance in other metrics. Both models score relatively low on ROUGE-2, indicating difficulties in capturing bi-gram overlaps. LongT5 shows a mentionable better performance in BLANC with a score of 0.1493 against a score of 0.1112. LongT5-NE consistently outperforms T5 with an extractive step across most metrics, showing substantial improvements in ROUGE scores and BLANC. Differences between the BERTScore and BARTScore are minimal.

Pegasus vs. PegasusX. PegasusX-NE surpasses Pegasus-NE across all metrics, except BARTScore where Pegasus-NE maintains a significant advantage. Pegasus-NE has a BARTScore of -1.8067 whereas PegasusX-NE obtains a BARTScore of -2.4528. When comparing Pegasus and PegasusX with an extractive step against each other, Pegasus shows slight improvements in ROUGE-1, ROUGE-L, BERTScore, and BARTScore. PegasusX, however, demonstrates a notable in ROUGE-2. The difference in BLANC scores is minimal, with PegasusX having a slight edge. PegasusX-NE outperforms Pegasus in most metrics, except for BARTScore. Pegasus outperforms PegasusX-NE with a BARTScore of -2.4528 compared to a BARTScore of -1.8940.
The findings of this section can be summarized as follows:

- Long context models without an extractive step outperform short context models without an extractive step on all metrics, except BARTScore.
- When an extractive step is used, results vary as short context models show advantages in specific metrics, implying that shot context models benefit more from an extractive step long context models do.
- Long context models without an extractive step generally outperform short context models with an extractive step across all metrics.

4.3 Human evaluation

The averaged human evaluation scores of six different summarization architectures can be found in Table 18. All individual scores and comments can be found in Appendix E. The human evaluation was performed after selecting the optimal extractive model and fine-tuning all abstractive models. The selected summarization architectures were:

(1) RoBERTa combined with BART with a dependent ratio, as RoBERTa with a dependent step is the optimal performing extractive model.
(2) RoBERTa combined with BART without an extractive step to function as a baseline.

(3) LexLM combined with BART as it is the highest performing extractive legal lm.
(4) Longformer combined with BART as it is the highest performing long context extractive model.
(5) PegasusX without an extractive step, as it is the highest performing long context abstractive model.
(6) RoBERTa combined with Llama3 as Llama3 is the single decoder-only model.

4.3.1 Scores on criteria. **Architecture 3, which uses LexLM for the extractive step and BART for the abstractive step, scores the highest across all criteria.** Overall, the scores on the criteria were somewhat different when compared to the results of the automated evaluation metrics. Table 18 shows the averaged results of the human evaluation for the chosen summarization architectures. The generated summary was rated on a scale of 1 to 5, where 1 indicates poor performance and 5 indicates excellent performance. Additional information about individual model characteristics such as context length can be found in Table 8 and Table 9. The definitions of the criteria used can be found in Table 13.

Table 18. Average human evaluation results.

Architecture #	Extractive model	Ratio type	Abstractive model	Factual correctness	Usability	Accuracy	Fluency	Coherence
1	RoBERTa	Dependent	BART	2.0	2.0	1.5	1.5	2.0
2	-	No extraction	BART	3.5	1.0	2.0	**3.0**	1.5
3	LexLM	Dependent	BART	**4.0**	**3.5**	**3.0**	**3.0**	**3.0**
4	Longformer	Dependent	BART	2.0	2.0	2.5	1.5	2.0
5	-	No extraction	PegasusX	3.5	1.0	2.5	**3.0**	1.0
6	RoBERTa	Dependent	Llama3	3.0	2.5	2.5	2.5	2.0

Factual correctness. Architecture 3 scores the highest on factual correctness. This suggests that LexLM improves how factually correct the final summary is relative to the source document. In contrast, Architectures 1 and 4, which incorporate an extractive step using RoBERTa and Longformer, score the lowest on this criterion.

Usability. Architecture 3 scores the highest on usability, with a rating of 3.5. This indicates that the combination of LexLM for the extractive step and BART for the abstractive step yields the most practical and user-friendly summaries according to human evaluators. In contrast, Architectures 2 and 5, which do not use any extractive model, score the lowest on usability with a rating of 1.0.

Accuracy. Regarding accuracy, which assesses the precision and correctness of the information presented in the summaries, LexLM achieves the highest score of 3.0. This is followed by Architecture 4, 5 and 6 which all have an accuracy score of 2.5. The architecture leveraging BART with an extractive step has the lowest accuracy score of 1.5. There is no definitive trend for model architecture on the precision and correctness of the information in the summary.

Fluency. The impact of model architecture on the summary's smoothness and ease of reading, in terms of form, content, and grammar, appears inconsistent. When evaluating fluency, Architectures 2, 3, and 5 score the highest, each with a rating of 3.0. In contrast, Architectures 1 and 4, which use RoBERTa and Longformer for the extractive step with a dependent ratio, score the lowest on fluency with ratings of 1.5.

Coherence. Finally, for coherence, which measures how logical the summary is to its linguistic context, LexLM again leads with a score of 3.0. Architecture 2 and 5, which both do not use an extractive step, but differ in their abstractive model, score the lowest on coherence with scores of 1.5 and 1.0 respectively.
The results of this section can be summarized as follows:

- Architecture 3, which uses LexLM as the extractive model, outperforms the other architectures across all criteria.
- Architecture 4, which uses Longformer as the extractive model, performs better than Architecture 1, which uses RoBERTa as the extractive model.
- The performance difference between Architecture 1, which uses an extractive step, and Architecture 2, which does not use an extractive step, is minimal.
- Architecture 6, using Llama3 as the abstractive model, scores better or similarly compared to both Architecture 1 and Architecture 2.
- The performance difference between Architecture 5 and Architecture 6 is small and inconsistent.

4.3.2 Additional comments. **The evaluators note that Architecture 3 performs well despite its shortcomings, furthermore they provide critical comments on several architectures and they point out the excessive repetitions of some architectures, which severely decrease the quality of the produced summary.** The additional comments can be found in Appendix E. This section gives short summaries of additional comments that were given besides the criteria scores.

Architecture 1. The evaluators indicated that the summary is not usable for readers without prior knowledge of the topic due to its incompleteness, factual mistakes, and inaccuracies. While it does touch upon the main principle of CBAM, some of the procedures and rules are described incorrectly.

Architecture 2. The evaluators indicated that the summary is not usable for readers as it places information in the wrong place, describing background details in the 'key points' section instead of the main content of the regulation. Additionally, one evaluator mentions that the summary completely misses the main point of what CBAM is, despite the state information being mostly correct with only a few mistakes.

Architecture 3. One evaluator indicates that the summary correctly grasps the key points of the regulation, making it quite useful. However, the evaluator noted that it is not fully complete and that the fluency and coherence of the sentences could be improved. Despite these shortcomings, the summary is considered a good starting point.

Architecture 4. One evaluator noted that this summary is less flawed than Summary 1 but is still unusable due to containing a significant amount of false information and incorrect words.

Architecture 5. Both mentioned that the summary contains excessive repetitions. Although the summary starts well, its usability degrades as more repetitions are encountered.

Architecture 6. One evaluator states that the summary contains quite some useful information. However because the summary contains a lot of repetition, it becomes unusable.
This section can be summarized as follows:

- Architecture 3, which incorporates LexLM, receives the most favourable comments. While it shows some shortcomings in coherence and fluency, the generated summary is of notable quality.
- The summaries of Architecture 5 and 6 contain useful information, especially in the beginning, but the amount of repetitions causes the quality to degrade severely.
- Some summaries may appear well-structured and readable but fail to capture the essential points of the regulation or contain factual errors.

5 DISCUSSION

Section 5.1 will answer the research questions that were composed at the beginning of this research. Hereafter, Section 5.2 will discuss the ethical implications of this research, and Section 5.3 describes the limitations found in this research. In Section 5.4, suggestions for future work are provided.

5.1 Research questions

This section provides answers to the research questions. The conclusions are based on the automated evaluation metrics as they were the main focus of the entire evaluation, as stated in Section 3.4. The research questions are as follows:

RQ Does a two- or multi-step extractive-abstractive architecture summarize long, regulatory documents better than not using any extractive steps?
RSQ1 Does a two-step or multi-step architecture perform better?
RSQ2 Does the use of a domain-specific legal language model for the extractive summarization step produce better results compared to when no domain-specific language model is used?
RSQ3 Does a long context length in the extractive summarization step provide better results?
RSQ4 Does a long context length in the abstractive summarization step provide better results?

5.1.1 Main Research question. Overall, there is no definitive answer to the main research question as the results of using a two-step architecture compared to not using any extractive steps differ per model and model architecture. RoBERTa with a two-step architecture was identified as the optimal model for extractive summarization. Subsequently, numerous abstractive summarization models were fine-tuned and tested using the two-step architecture. These fine-tuned models were then compared against models that did not use any extractive steps. The following conclusions can however be drawn:

- The performance of encoder-decoder abstractive summarization models generally worsens when using one extractive step, but this differs per model.
- Decoder-only models seem to benefit from an additional extractive step.

Encoder-decoder models generate a condensed representation of the text. Introducing an intermediary summary created by an encoder-only model can confuse, as this intermediary summary consists of disjointed sentences. This could explain the observed performance decline in encoder-decoder models. In contrast, the additional extractive step seems to benefit decoder-only models, likely because it effectively guides them toward producing coherent and relevant summaries.

5.1.2 Research sub-question 1. Based on the results, it can be concluded that utilizing a dependent ratio type achieves higher performance across most evaluation metrics compared to using fixed and hybrid ratios. Therefore, it can be inferred that a two-step architecture performs better than a multi-step architecture.

By using a multi-step architecture, sentences are combined from different chunks during the summarization process. Because K-means clustering is used for the selection of sentences in the concatenated chunk, the clustering may not accurately reflect the overall significance of the information within the context of the whole document. Consequently, the multi-step architecture could introduce noise and fail to capture the most relevant and coherent information, resulting in lower performance. It seems that using one extractive step is more efficient at capturing the most important sentences out of a chunk, relative to the context of the global document.

5.1.3 Research sub-question 2. The use of domain-specific legal language models for the extractive summarization step does not produce better results compared to when no domain-specific language model is used. General LMs outperform legal LMs across most metrics. The differences in performance between legal LMs and general LMs are minimal overall.

A possible explanation for this is that general LMs perform better because they are less focused on too specific legal text and thus provide a broader perspective. Legal LMs, on the other hand, tend to select very specific articles within the chunks, which results in missing out on the global context required for an effective summary. This focus on highly specific information can lead to a lack of coherence and comprehensiveness in the summaries. This finding is insightful because it suggests that using a general LM is effective across a domain-specific task.

5.1.4 Research sub-question 3. It can be concluded that a shorter context length in the extractive summarization step provides better results. Based on the evaluation results, it is evident that shorter context extractive summarization models, with a context length of 512 tokens, consistently achieve higher scores across almost all metrics compared to longer context models, with a context length of 4096 tokens. The top-performing models predominantly feature models with a short context length. The differences in performance between short and long context models are minimal but consistently favour the short context models.

This finding is particularly interesting because one would assume that longer context models would perform better by capturing more global context. However, it appears that the sequences may be too long for the models to properly encode them into representative embeddings. When sequences are excessively long, the models might struggle to maintain and encode all relevant information, leading to a loss of global context and coherence. This could explain why shorter context models, which deal with more manageable chunks of information, consistently perform better.

5.1.5 Research sub-question 4. Employing a long context length in the abstractive summarization step leads to improved results. The results indicate that the performance of long context abstractive summarization models worsens when using an extractive step, whereas short context abstractive models show varied results on using an extractive step. It seems that using a long context abstractive summarization model without a two- or multi-step architecture is preferred over using a short context abstractive summarization model with an extractive step.

Long context models, especially those with encoder-decoder architectures, can process more global information from the entire document. These models can capture broader context and nuances that are essential for generating coherent and comprehensive summaries. However, introducing an extractive step might degrade their performance. This could be because the extractive step breaks the text into concatenated summaries, which might lack the necessary context and continuity that is required for summarization.

5.1.6 Results of human evaluation. Some conclusions from human evaluation contrast with conclusions from human evaluation. Specifically, automated metrics scored summaries produced by general LMs higher, while human evaluators found a legal LM-based architecture to be superior. This is in contrast to the findings of RSQ2. In the human evaluation, the summarization architecture leveraging LexLM consistently outperformed the other architectures across all criteria. Additionally, an architecture employing Longformer as its extractive model performed better than an architecture using RoBERTa, which contrasts with the findings of RSQ3. However, some findings were consistent between both evaluation types, such as the minor performance difference between an architecture using no extraction combined with PegasusX and an architecture using Llama3 with a dependent extractive step. Human evaluation noted that some summaries may have good readability but contain inaccuracies, false information, or misplaced content. Additionally, evaluators highlighted the issue of heavy repetitions, which significantly decreased the overall quality of the summaries.

5.2 Ethical implications

5.2.1 Usage of a summarization model. The usage of summarization models for long regulatory documents generally does not pose significant ethical implications. These tools are designed to assist people by efficiently condensing large volumes of text. However, it is crucial to acknowledge that these models can occasionally produce errors or inaccuracies. Consequently, users should be aware that utilizing such tools carries a certain

level of risk. Although these models can provide valuable assistance, the responsibility for the final content remains with the user, and this consideration should be taken into account in professional settings.

5.2.2 Energy usage. The training and deployment of neural models require significant computational resources, leading to substantial energy consumption. The computational requirements raise concerns about the environmental impact of such models. To address this, researchers and practitioners should strive to optimize model efficiency, reducing the computational load and energy usage. Furthermore, exploring renewable energy sources for powering data centres can mitigate the environmental footprint. Awareness and proactive measures in managing energy consumption are important for the sustainable development and deployment of summarization models.

5.3 Limitations

5.3.1 Dataset. The dataset used in this research consists of 1442 instances, which, although sizable, is relatively small for training robust neural models. Additionally, the dataset is quite unbalanced, with some references and summaries being disproportionately long or short. This variability reflects the nature of regulatory documents. Another challenge with the dataset is the high level of repetition within the documents. There are many occasions of repetitive n-grams due to bullet point lists, referrals to other documents, and dates. Repetitive text in the reference texts and golden reference summaries can be challenging for models to handle, as it may cause the models to become focused on these repetitions. This can lead to repetitions being reproduced in the generated summaries. Besides this, summarizing long regulatory texts is an inherently difficult task that requires a deep understanding of the subject matter, background context of the text, and legal expertise. These factors underscore the complexity of the task.

5.3.2 Extractive summarization sentence selection. K-means clustering was used to select sentences during the extractive summarization process. This means that the quality of the extractive summaries in this research is heavily dependent on the K-means selection process. Based on prior research, K-means seems to perform well. But still, no alternative selection methods were employed for comparison. It is also important to note that an evaluation of the intermediary extractive summaries was not conducted. This lack of evaluation means that the quality of the extracted content, which serves as the basis for the subsequent abstractive summarization, remains unverified.

Moreover, there is an inherent overhead in the extractive step due to the way sentences are counted and selected. For very large texts, maintaining a specific ratio, such as 0.17, becomes increasingly difficult because the process involves selecting entire sentences rather than individual tokens. This chunking of text can lead to inconsistencies, especially when the desired ratio is small. As a result, the cumulative error can become significant, affecting the overall length of the intermediary summary.

Given these challenges, there is a clear need for methods that can also fine-tune the added extractive step. Such methods would help to ensure that the extracted content is optimally suited for the subsequent abstractive summarization.

5.3.3 Token discrepancy. In this study, extractive tokens were used to calculate the amount of extractive steps that needed to be taken. This also means that the tokenizers of the extractive models were used to count the amount of tokens. However, abstractive models use different tokenizers, so there could be a discrepancy between the amount of tokens that the extractive summarization model summarizes, and the actual amount of tokens the abstractive model creates. For example, RoBERTa can tokenize a text up to 1024 tokens while for BART, the number of tokens would be lower. This means that the context length of BART is not fully used. The discrepancy could also work the other way around, where the abstractive model will have to truncate the intermediary summary to ensure it fits in the context length. Calculating the number of extractive steps by counting tokens

with the abstractive model is not ideal, as there will again be a discrepancy between the number of tokens an extractive model creates and the abstractive model creates.

5.3.4 Repetition prevention. More could have been done to prevent repetition errors in the generated summaries. Techniques such as penalty mechanisms [67] for repeated phrases could have been used to prevent this issue. The penalty mechanisms punish a model for repeating the same tokens. However, these penalties come at the cost of more hallucinations. Due to the importance of creating factually consistent summaries, repetition mechanisms were not added. Also, prior work showed no usage of any repetition prevention mechanisms. Instead, the focus of this research was on evaluating the overall performance of various models without additional interventions. This approach allowed for a more clear comparison of the models' natural capabilities and limitations in handling regulatory documents. The trade-off between hallucination and repetition is a topic which is intensively researched.

5.3.5 Decoder-only configurations. The training process for decoder-only models, such as Llama3, encountered specific limitations. The feature to load the best model at the end of the training was not functioning correctly, which meant that the final model used was not necessarily the one with the lowest validation loss. Fortunately, the validation loss did not show significant fluctuations, suggesting that the impact of selecting the final model was minimal.

To strengthen the findings, it would be beneficial to include more decoder-only models in the evaluation. This would help determine if the observed results are consistent across different decoder-only models and provide a broader understanding of their performance in summarizing regulatory documents. By expanding the range of models tested, the research could offer more robust conclusions on decoder-only models in this context.

5.3.6 Evaluation metrics.

Automated evaluation metrics. In this research, a diverse set of evaluation metrics is used to create a comprehensive view of model performance. While this approach provides a broad perspective, it also showcases limitations in NLG evaluation techniques. The use of multiple metrics can sometimes lead to seemingly contradictory results, as illustrated by the performance of Llama3 without an extractive step across different metrics (see Section 4.2.1). Llama3 without an extractive step could sometimes create empty summaries. This caused a significant drop in BERTScore compared to its extraction-based counterpart, while other metrics like ROUGE, BARTScore, and BLANC showed less pronounced differences. This discrepancy initially appears puzzling, as one might expect poor-quality or empty summaries to have a consistent impact across all metrics. However, a closer examination of the practical score ranges and calculation methods of each metric explains these inconsistencies. BERTScore, which relies on cosine similarity between contextual embeddings, typically produces high scores, often above 0.8. Consequently, zero scores can significantly lower the overall score. In contrast, ROUGE scores, while theoretically ranging from 0 to 1, often fall between 0 and 0.5 for this research. The compressed range means that occasional zero scores have a less dramatic impact on the average ROUGE score. BARTScore, ranging from -5.5 to -1.0 in this research, is also less affected by poor outputs as they result in very negative scores rather than zeros, thus not pulling down the F-score as severely. BLANC scores, which theoretically range from -1 to 1, typically fall between 0 and 0.3 in practice, as noted in the original BLANC paper. This narrow range makes BLANC less sensitive to occasional poor outputs.

Additionally, BARTScore is dependent on the context length of BART, which is 1024 tokens. This means that in some cases, the summaries cannot be evaluated properly, providing skewed scores. BLANC is also limited to the context of BERT, which is only 512 tokens. These limited context lengths hamper BARTScore and BLANC from providing a proper evaluation as they cannot read the entire summary.

Understanding the shortcomings of different metrics is important for accurately interpreting model performance in summarization tasks. The shortcomings explain the seemingly inconsistent impact of poor-quality summaries

across various evaluation metrics. This analysis highlights a limitation in the current evaluation metrics for NLG tasks. The limitations of automated evaluation metrics underscore the importance of using multiple, diverse evaluation metrics and carefully interpreting their results. It also emphasises the need for research into robust evaluation methods for NLG tasks, as the current metrics may not always provide an accurate picture of model performance. Each evaluation method offers a different perspective, suggesting that while content overlap might be good, it does not necessarily guarantee optimal accuracy and adequacy from a human perspective.

Human evaluation. Due to resource limitations, the human evaluation was conducted on a small scale. While the results provide valuable insights, the limited scope prevents them from being significant. In addition, human evaluation was only performed on a select group of architectures after obtaining all the results. Ideally, a more extensive human evaluation would have been conducted, particularly in the early stages of the research the optimal extractive summarization model was selected, as it was used in subsequent research.

Human evaluation is also inherently subjective because it involves people. In this evaluation, two participants were involved: one with a deep understanding of the document and another with a less detailed understanding. This dual perspective offers diverse insights but also introduces biases. The participant with a deeper understanding was more critical across all criteria, while the other participant tended to give higher scores to summaries that elaborated on certain parts of the text. Additionally, evaluators faced challenges in interpreting the criteria definitions. This phenomenon is widely recognized in the field of NLG, as highlighted by previous research [110] [57].

5.4 Future work

Given the limitations encountered in this research, several areas for future work have been identified to enhance the performance and reliability of summarizing long regulatory documents. Also, several future work recommendations are given based on ideas and theories that came up during this research. By addressing these areas, future work can build upon the findings of this research.

5.4.1 Adjustments to the extractive summarization step. Future work should focus on refining the extractive step to ensure higher quality and more representative extracted content. This could involve experimenting with different techniques to better capture the most relevant sentences. Additionally, incorporating an evaluation of the intermediary extractive summarizations themselves could provide valuable insights and improve the overall summarization process. It would be even more interesting to incorporate the extractive step into the fine-tuning process. Fine-tuning the extractive step together with the abstractive step could also enhance the quality of the final summaries. Future research could adjust the extractive summarization step by using the ratio between extractive and abstractive token lengths. For the extractive and abstractive models, a ratio can be calculated between the number of tokens that are created by the extractive and abstractive models. Then during summarization, the number of tokens of an input is counted by the abstractive model. The number of tokens is converted to the number of tokens that the extractive model makes out of this. Using these tokens, a more accurate extractive compression ratio can be used for the extractive summarization step. Subsequently, the intermediary summary will fit better into the context length of the abstractive summarization model.

5.4.2 Two- or multi-step abstractive-only summarization. In this research, a two- or multi-step abstractive-only summarization approach was not attempted. This could be an interesting research topic for future work, as it addresses some of the limitations observed with the extractive-abstractive method. Specifically, the extractive summaries could contain incoherent sentences, which can confuse some abstractive models during the abstractive summarization step. By adopting a two- or multi-step architecture that solely uses abstractive summarization, it may be possible to generate more coherent and contextually accurate summaries because the intermediary summaries will most likely not consist of incoherent sentences.

5.4.3 Decoder-only model training enhancement. The limitations encountered with decoder-only models, such as Llama3, suggest several avenues for improvement. Future work could experiment with optimal epoch lengths and ensure that features like loading the best model at the end of training are functional. This would ensure that the model with the best performance is used for evaluation. Moreover, using multiple decoder-only models could provide a more comprehensive understanding of their capabilities and limitations.

5.4.4 Exploration of new encoder-decoder architectures. It is interesting to note that decoder-only models experience a performance boost from the extractive step because the extractive step uses encoder-only models. There might be potential for an in-depth comparison of our make-shift 'encoder-decoder' to pure encoder-decoder models.

The findings suggest that an encoder-decoder architecture is beneficial for complex tasks such as the summarization of long, legal texts. In this research, the performance of BART is very competitive as it only has 406 million parameters. Its performance is up to par with Llama3, which boasts over 8 billion parameters. Exploring newer and more modern encoder-decoder models could provide valuable insights. However, this exploration is dependent on the release of such models, as current research trends have been focusing on decoder-only architectures. Relatively new encoder-decoder models such as T5 and Pegasus were outperformed by BART, indicating that BART was created with a strong combination of data, pre-training objectives, and model architecture.

5.4.5 Automated evaluation metrics. Improving the evaluation methodology is another critical area for future research. Developing more nuanced and domain-specific evaluation metrics could provide a better assessment of the model's performance in summarizing regulatory documents. These metrics should capture not only the accuracy and coherence of the summaries but also their relevance and utility.

5.4.6 Human evaluation. In this research, it can be seen that the results of the human evaluation greatly differ from the automated evaluation metrics. The human evaluation is very useful however and highlights the importance of assessing summaries beyond surface-level quality. This was apparent in summarization architectures which scored well on the automated evaluation metrics but failed to capture the essence of a regulation or produced factual inaccuracies. The usage of human evaluation should be included in future work but on a larger scale to provide more robust and statistically significant results. A large-scale human evaluation captures a wider range of perspectives, and better identifies patterns or trends in summary quality.

Currently, human evaluation is only conducted at the end. Future work could experiment with using human evaluation as a guide throughout the development process to determine which models perform best. By using human feedback, future research can more easily identify which model architectures perform best to more accurately identify which extractive summarization models show the best performance. Additionally, methods such as Instruction Tuning [99] and Reinforcement Learning from Human Feedback [67] could also be used to adjust the models after the initial fine-tuning process.

5.4.7 Applicability to other domains. Future research could explore whether different types of texts benefit from the same summarization architecture. The current study focuses on regulatory documents, but it would be valuable to investigate how well the two- and multi-step architectures perform across various domains such as scientific literature, news articles, and technical manuals.

5.4.8 Comparison of computational requirements. In addition to evaluating performance, it is crucial to compare the computational requirements of different models and architectures. For different model combinations, it is interesting to assess the computational expense of training and deploying various summarization models, as some may require significantly more computational resources due to their complexity and the length of texts they process.

When incorporating an extractive step, the computational overhead introduced by this additional layer of processing should be evaluated, as it can improve performance for certain models but also increase the overall computational load.

Furthermore, the size and computational requirements of abstractive and extractive models due to attention mechanisms should be analyzed, as large models with extensive attention mechanisms may require more memory and processing power, which can be a limiting factor for their practical application. By comparing these aspects, future work can provide a clear picture of the trade-offs between model performance and computational efficiency.

6 CONCLUSION

This research explored the field of Automatic Text Summarization (ATS), particularly focusing on summarizing lengthy and complex regulatory documents. The study began with an examination of supervised and unsupervised learning approaches for ATS, deciding on supervised learning due to the availability of a dataset containing golden-reference summaries. A review of related work was conducted, covering early extractive and abstractive techniques, the implementation of neural models for summarization techniques, methods for summarizing long documents, two- or multi-step summarization approaches, research on regulatory and legal text summarization, relevant datasets, and evaluation metrics.

The methods of this research revolved around a two- and multi-step extractive-abstractive architecture designed to improve the summarization of long regulatory texts. This architecture involved chunking documents into manageable sections, applying extractive summarization to generate intermediary summaries, and then using abstractive summarization to produce the final summaries. Various models and approaches were tested and compared, including different types of language models (general and domain-specific legal models), and models with varying context lengths for extractive and abstractive summarization. The study employed both automated metrics and human evaluation to assess the quality of the generated summaries.

The research aimed to answer whether a two- or multi-step extractive-abstractive approach yields better summarization results than a direct summarization approach, with specific attention to the performance differences between different summarization models in both steps. Conclusions were primarily drawn based on automated evaluation metrics. The findings indicated that decoder-only models benefited from the additional extractive step, while encoder-decoder models generally did not. However, this effect for the encoder-decoder architecture differed per model. Some encoder-decoder models performed worse when an intermediary extractive step was introduced, while others showed varied results. Further investigation revealed that a two-step summarization architecture generally outperformed a multi-step architecture. The multi-step approach possibly introduced noise due to the less effective clustering of sentences from different chunks, failing to maintain coherence and relevance. Additionally, the study found that general language models outperformed domain-specific legal language models, suggesting that a broader context provided by general language models is more effective than the specificity of legal language models for summarizing regulatory documents.

In terms of context length, shorter context lengths were found to be more effective for extractive summarization, while longer context lengths benefited abstractive summarization. Long context models were better at capturing global information necessary for generating coherent and comprehensive summaries. However, the introduction of an extractive step to the long context abstractive models appeared to degrade their performance by disrupting the continuity and context required for effective summarization.

Finally, human evaluations provided a different perspective from automated metrics, favouring legal language models over general language models and highlighting the limitations of automated evaluations, such as the tendency to overlook issues of readability, inaccuracies, and repetitions. This discrepancy underscores the importance of including human judgment in the evaluation of summarization models.

By offering insights into the effectiveness of different summarization strategies and model architectures, this study contributes to the broader field of Natural Language Processing and its application in regulatory compliance. The findings provide valuable insights into the design of summarization architectures and the selection of appropriate models based on document characteristics. These findings have implications for improving accessibility and understanding of regulatory texts, which is crucial in dynamic industries such as the green molecule sector. The research also highlights the ongoing challenges in evaluating generated summaries. It showcases the need for improved automated metrics and large-scale human assessments.

REFERENCES

[1] [n. d.]. Llama-2-7B-32K-Instruct — and fine-tuning for Llama-2 models with Together API. https://www.together.ai/blog/llama-2-7b-32k-instruct

[2] Meta AI. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3/ Accessed: 2024-06-04.

[3] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. 2016. Predicting judicial decisions of the European court of human rights: A natural language processing perspective. *PeerJ Computer Science* 2016 (2016). Issue 10. https://doi.org/10.7717/peerj-cs.93

[4] Dennis Aumiller, Ashish Chouhan, and Michael Gertz. 2022. EUR-Lex-Sum: A Multi- and Cross-lingual Dataset for Long-form Summarization in the Legal Domain. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. https://doi.org/10.18653/v1/2022.emnlp-main.519

[5] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

[6] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Proceedings of the Workshop ACL 2005*.

[7] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SCIBERT: A pretrained language model for scientific text. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/d19-1371

[8] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. arXiv:2004.05150 [cs.CL]

[9] Dan Biderman, Jose Gonzalez Ortiz, Jacob Portes, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. 2024. LoRA Learns Less and Forgets Less. arXiv:2405.09673 [cs.LG]

[10] Avi Bleiweiss. 2023. Two-step Text Summarization for Long-form Biographical Narrative Genre. https://doi.org/10.18653/v1/2023.codi-1.20

[11] Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). *WMT 2018 - 3rd Conference on Machine Translation, Proceedings of the Conference* 2. https://doi.org/10.18653/v1/W18-64028

[12] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 2020-December.

[13] Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact-aware neural abstractive summarization. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. https://doi.org/10.1609/aaai.v32i1.11912

[14] Asli Celikyilmaz, Antoine Bosselut, Xiaodong He, and Yejin Choi. 2018. Deep communicating agents for abstractive summarization. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1. https://doi.org/10.18653/v1/n18-1150

[15] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2020. Neural legal judgment prediction in English. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. https://doi.org/10.18653/v1/p19-1424

[16] Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2017. Extracting contract elements. *Proceedings of the International Conference on Artificial Intelligence and Law*. https://doi.org/10.1145/3086512.3086515

[17] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*. https://doi.org/10.18653/v1/2020.findings-emnlp.261

[18] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2020. Large-scale multi-label text classification on EU legislation. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. https://doi.org/10.18653/v1/p19-1636

[19] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2021. Neural Contract Element Extraction Revisited: Letters from Sesame Street. arXiv:2101.04355 [cs.CL]

[20] Ilias Chalkidis, Nicolas Garneau, Anders Søgaard, Cătălină Goantă, and Daniel Martin Katz. 2023. LeXFiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development. *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 1. https://doi.org/10.18653/v1/2023.acl-long.865

[21] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. *Proceedings of the Annual Meeting of the Association for*

Computational Linguistics 1. https://doi.org/10.2139/ssrn.3936759

[22] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. arXiv:1504.00325 [cs.CV]

[23] Yen Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1. https://doi.org/10.18653/v1/p18-1063

[24] Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers* 1. https://doi.org/10.18653/v1/p16-1046

[25] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.3115/v1/d14-1179

[26] Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*. https://doi.org/10.18653/v1/n16-1012

[27] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 2. https://doi.org/10.18653/v1/n18-2097

[28] Text Analysis Conference. 2010. TAC 2010 Guided Summarization Task. https://tac.nist.gov//2010/Summarization/Guided-Summ.2010.guidelines.html

[29] Text Analysis Conference. 2011. TAC 2011 Guided Summarization Task. https://tac.nist.gov/2011/Summarization/Guided-Summ.2011.guidelines.html

[30] John M. Conroy and Dianne P. O'leary. 2001. Text summarization via hidden Markov models. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*. https://doi.org/10.1145/383952.384042

[31] Ht Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 update summarization task. *Tac* (2008).

[32] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. arXiv:2305.14314 [cs.LG]

[33] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1.

[34] Sunil Dhankhar and Mukesh Kumar Gupta. 2022. *Automatic Extractive Summarization for English Text: A Brief Survey.* https://doi.org/10.1007/978-981-16-3346-1_15

[35] Yue Dong, Andrei Mircea, and Jackie C.K. Cheung. 2021. Discourse-aware unsupervised summarization of long scientific documents. *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*. https://doi.org/10.18653/v1/2021.eacl-main.93

[36] Vladimir Eidelman. 2019. BillSum: A Corpus for Automatic Summarization of US Legislation. https://doi.org/10.18653/v1/d19-5406

[37] Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. 2021. Automatic text summarization: A comprehensive survey. https://doi.org/10.1016/j.eswa.2020.113679

[38] Güneş Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research* 22 (2004). https://doi.org/10.1613/jair.1523

[39] European Union. 2023. Regulation (EU) 2023/0956 of the European Parliament and of the Council of 10 May 2023 on Machinery. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32023R0956 Accessed: 2024-06-28.

[40] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021). https://doi.org/10.1162/tacl_a_00373

[41] Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. 2020. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. https://doi.org/10.18653/v1/p19-1102

[42] Yi Feng, Chuanyi Li, and Vincent Ng. 2022. Legal Judgment Prediction: A Survey of the State of the Art. *IJCAI International Joint Conference on Artificial Intelligence*. https://doi.org/10.24963/ijcai.2022/765

[43] Rafael Ferreira, Luciano De Souza Cabral, Rafael Dueire Lins, Gabriel Pereira E Silva, Fred Freitas, George D.C. Cavalcanti, Rinaldo Lima, Steven J. Simske, and Luciano Favaro. 2013. Assessing sentence scoring techniques for extractive text summarization. Issue 14. https://doi.org/10.1016/j.eswa.2013.04.023

[44] Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review* 47 (2017). Issue 1. https://doi.org/10.1007/s10462-016-9475-9

[45] Alexios Gidiotis and Grigorios Tsoumakas. 2020. A Divide-and-Conquer Approach to the Summarization of Long Documents. *IEEE/ACM Transactions on Audio Speech and Language Processing* 28 (2020). https://doi.org/10.1109/TASLP.2020.3037401

[46] Dan Gillick and Yang Liu. 2010. Non-expert evaluation of summarization systems is risky. *Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, Mturk 2010 at the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2010 - Proceedings.*

[47] Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* 2019-September. https://doi.org/10.21437/Interspeech.2019-3079

[48] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia,* 4 (2003). Issue 1.

[49] Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1. https://doi.org/10.18653/v1/n18-1065

[50] Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontañón, Jianmo Ni, Yun Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient Text-To-Text Transformer for Long Sequences. *Findings of the Association for Computational Linguistics: NAACL 2022 - Findings.* https://doi.org/10.18653/v1/2022.findings-naacl.55

[51] Som Gupta and S. K. Gupta. 2019. Abstractive summarization: An overview of the state of the art. https://doi.org/10.1016/j.eswa.2018.12.011

[52] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *Proceedings of the Annual Meeting of the Association for Computational Linguistics.* https://doi.org/10.18653/v1/2020.acl-main.740

[53] Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan Fang Li, Yong Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021.* https://doi.org/10.18653/v1/2021.findings-acl.413

[54] Peter Henderson, Mark S. Krass, Lucia Zheng, Neel Guha Christopher D. Manning, Dan Jurafsky, and Daniel E. Ho. 2022. Pile of Law: Learning Responsible Data Filtering from the Law and a 256GB Open-Source Legal Dataset. *Advances in Neural Information Processing Systems* 35.

[55] Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems* 2015-January.

[56] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9 (1997). Issue 8. https://doi.org/10.1162/neco.1997.9.8.1735

[57] David M. Howcroft, Anya Belz, Miruna Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions. *INLG 2020 - 13th International Conference on Natural Language Generation, Proceedings.* https://doi.org/10.18653/v1/2020.inlg-1.23

[58] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. arXiv:2106.09685

[59] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2020. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. arXiv:1904.05342 [cs.CL]

[60] Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient Attentions for Long Document Summarization. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference.* https://doi.org/10.18653/v1/2021.naacl-main.112

[61] Quzhe Huang, Mingxu Tao, Chen Zhang, Zhenwei An, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer LLaMA Technical Report. arXiv:2305.15062 [cs.CL]

[62] Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. 2021. Summarization of legal documents: Where are we now and the way forward. https://doi.org/10.1016/j.cosrev.2021.100388

[63] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI meeting corpus. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 1. https://doi.org/10.1109/icassp.2003.1198793

[64] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. Issue 12. https://doi.org/10.1145/3571730

[65] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. arXiv:2401.04088 [cs.LG]

[66] Ambedkar Kanapala, Sukomal Pal, and Rajendra Pamula. 2019. Text summarization from legal documents: a survey. *Artificial Intelligence Review* 51 (2019). Issue 3. https://doi.org/10.1007/s10462-017-9566-2

[67] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. arXiv:1909.05858 [cs.CL] https://arxiv.org/abs/1909.05858

[68] Mohammad Khosravani and Amine Trabelsi. 2023. Recent Trends in Unsupervised Summarization. arXiv:2305.11231 [cs.CL]

[69] Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1.

[70] Eunchong Kim, Taewoo Yoo, Gunhee Cho, Suyoung Bae, and Yun-Gyung Cheong. 2022. The CreativeSumm 2022 Shared Task: A Two-Stage Summarization Model using Scene AttributesUtterances. *COLING, Proceedings of The Workshop on Automatic Summarization for Creative Writing*.

[71] Svea Klaus, Ria Van Hecke, Kaweh Djafari Naini, Ismail Sengor Altingovde, Juan Bernabé-Moreno, and Enrique Herrera-Viedma. 2022. Summarizing Legal Regulatory Documents using Transformers. *SIGIR 2022 - Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3477495.3531872

[72] Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305* (2018).

[73] Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/d19-1051

[74] Philippe Laban, Andrew Hsi, John Canny, and Marti A. Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.460

[75] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36 (2020). Issue 4. https://doi.org/10.1093/bioinformatics/btz682

[76] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.703

[77] Sujian Li, Wei Wang, and Yongwei Zhang. 2009. Tac 2009 update summarization of icl. *Tac* (2009).

[78] Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Improving Unsupervised Extractive Summarization with Facet-Aware Modeling. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. https://doi.org/10.18653/v1/2021.findings-acl.147

[79] C Y Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Proceedings of the workshop on text summarization branches out (WAS 2004)* (2004). Issue 1.

[80] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI Open* 3 (2022). https://doi.org/10.1016/j.aiopen.2022.10.001

[81] Jeffrey Ling and Alexander M. Rush. 2017. Coarse-to-fine attention models for document summarization. *EMNLP 2017 - Workshop on New Frontiers in Summarization, NFiS 2017 - Workshop Proceedings*. https://doi.org/10.18653/v1/w17-4505

[82] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Łukasz Kaiser, and Noam Shazeer. 2018. Generating wikipedia by summarizing long sequences. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

[83] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. arXiv:2303.16634 [cs.CL]

[84] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

[85] H. P. Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM JOURNAL* (1958). Issue April.

[86] Qingsong Ma, Johnny Tian Zheng Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. *WMT 2019 - 4th Conference on Machine Translation, Proceedings of the Conference* 2. https://doi.org/10.18653/v1/w19-5302

[87] Dimitris Mamakas, Petros Tsotsi, Ion Androutsopoulos, and Ilias Chalkidis. 2022. Processing Long Legal Documents with Pre-trained Transformers: Modding LegalBERT and Longformer. *NLLP 2022 - Natural Legal Language Processing Workshop 2022, Proceedings of the Workshop*. https://doi.org/10.18653/v1/2022.nllp-1.11

[88] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.173

[89] Iain McCowan, Jean Carletta, W Kraaij, S Ashby, S Bourban, M Flynn, M Guillemot, T Hain, J Kadlec, V Karaiskos, et al. 2005. The AMI meeting corpus. *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research* 88.

[90] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 - A meeting of SIGDAT, a Special Interest Group of the ACL held in conjunction with ACL 2004.*

[91] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings.*

[92] Derek Miller. 2019. Leveraging BERT for Extractive Text Summarization on Lectures. arXiv:1906.04165 [cs.CL]

[93] N. Moratanch and S. Chitrakala. 2016. A survey on abstractive text summarization. *Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2016.* https://doi.org/10.1109/ICCPCT.2016.7530193

[94] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çaglar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings.* https://doi.org/10.18653/v1/k16-1028

[95] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference* 1. https://doi.org/10.18653/v1/n18-1158

[96] Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. *Mining text data* (2012), 43–76.

[97] Joel Larocca Neto, Alex A. Freitas, and Celso A.A. Kaestner. 2002. Automatic text summarization using a machine learning approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 2507. https://doi.org/10.1007/3-540-36127-8_20

[98] OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[99] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL] https://arxiv.org/abs/2203.02155

[100] Nadav Oved and Ran Levy. 2021. PASS: Perturb-and-select summarizer for product reviews. *ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/2021.acl-long.30

[101] Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 2002-July.

[102] Jason Phang, Yao Zhao, and Peter J. Liu. 2022. Investigating Efficiently Extending Transformers for Long Input Summarization. arXiv:2208.04347 [cs.CL]

[103] Jonathan Pilault, Raymond Li, Sandeep Subramanian, and Christopher Pal. 2020. On extractive and abstractive neural document summarization with transformer language models. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/2020.emnlp-main.748

[104] Maja Popović. 2015. Chrf: Character n-gram f-score for automatic mt evaluation. *10th Workshop on Statistical Machine Translation, WMT 2015 at the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015 - Proceedings*. https://doi.org/10.18653/v1/w15-3049

[105] Xi Peng Qiu, Tian Xiang Sun, Yi Ge Xu, Yun Fan Shao, Ning Dai, and Xuan Jing Huang. 2020. Pre-trained models for natural language processing: A survey. Issue 10. https://doi.org/10.1007/s11431-020-1647-3

[106] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. *OpenAI.com* (2018).

[107] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21 (2020).

[108] Sebastian Raschka. 2023. Practical Tips for Finetuning LLMs Using LoRA (Low-Rank Adaptation). https://magazine.sebastianraschka.com/p/practical-tips-for-finetuning-llms Accessed: 2024-06-10.

[109] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/d19-1410

[110] Ehud Reiter. 2018. A structured review of the validity of BLEU. Issue 3. https://doi.org/10.1162/COLI_a_00322

[111] Gaetano Rossiello, Pierpaolo Basile, and Giovanni Semeraro. 2017. Centroid-based Text Summarization through Compositionality of Word Embeddings. *MultiLing 2017 - Workshop on Summarization and Summary Evaluation Across Source Types and Genres, Proceedings of the Workshop*. https://doi.org/10.18653/v1/w17-1003

[112] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. *Proceedings of EMNLP 2015* 1509 (2015). Issue 685.

[113] Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia* 6 (2008). Issue 12.

[114] Thomas Scialom, Paul Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. 2021. QuestEval: Summarization Asks for Fact-based Evaluation. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*. https://doi.org/10.18653/v1/2021.emnlp-main.529

[115] Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2019. Answers unite! Unsupervised metrics for reinforced summarization models. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/d19-1320

[116] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1. https://doi.org/10.18653/v1/P17-1099

[117] Eva Sharma, Chen Li, and Lu Wang. 2020. BigPatent: A large-scale dataset for abstractive and coherent summarization. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. https://doi.org/10.18653/v1/p19-1212

[118] Xin Shen and Wai Lam. 2022. Improved Divide-and-Conquer Approach to Abstractive Summarization of Scientific Papers. *Proceedings - 2022 4th International Conference on Natural Language Processing, ICNLP 2022*. https://doi.org/10.1109/ICNLP55136.2022.00073

[119] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. arXiv:2009.01325 [cs.CL] https://arxiv.org/abs/2009.01325

[120] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems* 4. Issue January.

[121] Ignacio Tampe, Marcelo Mendoza, and Evangelos Milios. 2022. Neural Abstractive Unsupervised Summarization of Online News Discussions. *Lecture Notes in Networks and Systems* 295. https://doi.org/10.1007/978-3-030-82196-8_60

[122] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra,

Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL]

[123] Howard Turtle. 1995. Text retrieval in the legal world. *Artificial Intelligence and Law* 3 (1995). Issue 1-2. https://doi.org/10.1007/BF00877694

[124] Oleg Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: Human-free quality estimation of document summaries. https://doi.org/10.18653/v1/2020.eval4nlp-1.2

[125] Oleg V. Vasilyev, Vedant Dharnidharka, Nicholas Egan, Charlene Chambliss, and John Bohannon. 2020. Sensitivity of BLANC to human-scored qualities of text summaries. *CoRR* abs/2010.06716 (2020). arXiv:2010.06716 https://arxiv.org/abs/2010.06716

[126] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 2017-December.

[127] Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.450

[128] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models Are Zero-Shot Learners. arXiv:2109.01652 [cs.CL] https://arxiv.org/abs/2109.01652

[129] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. arXiv:2206.07682 [cs.CL] https://arxiv.org/abs/2206.07682

[130] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. BloombergGPT: A Large Language Model for Finance. arXiv:2303.17564 [cs.LG]

[131] Xinghao Wu and Moritz Lode. 2020. Language Models are Unsupervised Multitask Learners (Summarization). *OpenAI Blog* 1 (2020). Issue May.

[132] Yuxiang Wu and Baotian Hu. 2018. Learning to extract coherent summary via deep reinforcement learning. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*. https://doi.org/10.1609/aaai.v32i1.11987

[133] Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for Chinese legal long documents. *AI Open* 2 (2021). https://doi.org/10.1016/j.aiopen.2021.06.003

[134] Yumo Xu and Mirella Lapata. 2020. Coarse-to-fine query focused multi-document summarization. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/2020.emnlp-main.296

[135] Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. FinBERT: A Pretrained Language Model for Financial Communications. arXiv:2006.08097 [cs.CL]

[136] Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. BARTSCORE: Evaluating Generated Text as Text Generation. *Advances in Neural Information Processing Systems* 33.

[137] L. A. Zadeh. 1965. Fuzzy sets. *Information and Control* 8 (1965). Issue 3. https://doi.org/10.1016/S0019-9958(65)90241-X

[138] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems* 2020-December.

[139] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTSCORE: EVALUATING TEXT GENERATION WITH BERT. *8th International Conference on Learning Representations, ICLR 2020*.

[140] Yusen Zhang, Ansong Ni, Ziming Mao, Chen Henry Wu, Chenguang Zhu, Budhaditya Deb, Ahmed H. Awadallah, Dragomir Radev, and Rui Zhang. 2022. SUMMN: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents. *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 1.

[141] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*. https://doi.org/10.18653/v1/d19-1053

[142] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. arXiv:2304.11277

[143] Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help?: Assessing self-supervised learning for law and the CaseHOLD dataset of 53,000+ legal holdings. *Proceedings of the 18th International Conference on Artificial Intelligence and Law, ICAIL 2021*. https://doi.org/10.1145/3462757.3466088

[144] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.466

[145] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a Unified Multi-Dimensional Evaluator for Text Generation. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*. https://doi.org/10.18653/v1/2022.emnlp-main.131

[146] Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A New Benchmark for Query-based Multi-domain Meeting Summarization. *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*. https://doi.org/10.18653/v1/2021.naacl-main.472

[147] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-Tuning Language Models from Human Preferences. arXiv:1909.08593 [cs.CL]

A THE ETHICS AND PRIVACY QUICK SCAN

The Ethics and Privacy Quick Scan of the Utrecht University Research Institute of Information and Computing Sciences was conducted (see Annex A). It classified this research as low-risk with no fuller ethics review or privacy assessment required.

B EXAMPLE SUMMARY

Market surveillance and compliance of products
Market surveillance and compliance of products
SUMMARY OF:
Regulation (EU) 2019/1020 — market surveillance and compliance of products
WHAT IS THE AIM OF THE REGULATION?
It aims to improve how the free movement of goods principle works by strengthening market surveillance* of products covered by EU hamonisation legislation. This must ensure a high level of protection of health and safety, in general and in the workplace, and protect consumers, the environment, public security and other public interest.
It lays down rules and procedures for economic operators* and establishes a system for their cooperation with supervisory authorities.
It establishes controls on products imported into the EU.
It deletes and replaces Articles 15 to 29 of Regulation (EC) No 765/2008 (see summary on Accreditation and market surveillance) and amends Directive 2004/42/EC and Regulation (EU) No 305/2011 (see summary on Construction products).
KEY POINTS

The regulation applies to products:
covered by EU harmonisation legislation (and set out in Annex I);
imported into the EU which are not subject to specific legislation.
Certain products may not be offered for sale to EU consumers without an economic operator established in the EU:
who keeps EU conformity and performance declarations and makes these and the technical documentation available to authorities when asked;
who informs the authorities when they consider a product poses a risk;
who cooperates with the authorities, when asked, by taking immediate corrective action — from remedying the fault to recall or destroying the item — if a product is considered non-compliant, and helps to eliminate or mitigate risks;
whose name and contact details are on the product, packaging or accompanying document.
Market surveillance authorities:
carry out effective surveillance of products sold online and offline;
perform appropriate documentary, physical and laboratory checks on products, taking into account possible hazards;
act when a product, properly installed, maintained and used for its intended purpose: could damage users' health and safety,does not conform to EU legislation;
ensure economic operators take corrective action when instructed and act when they fail to do so;
establish procedures to follow up complaints and reports and to verify economic operators have taken corrective action;
apply a high level of transparency and make available to the public any information they consider relevant to protect end users' interests;
provide economic operators with grounds for their decisions to give them an opportunity to respond;
notify the European Commission and the other EU countries immediately of any measures they take if these could have an impact in other EU countries;
may ask colleagues elsewhere in the EU to help with investigations and enforcement and participate in peer reviews to improve the system's overall efficiency;
have powers to: start investigations on their own initiative,require economic operators to provide relevant documents, data and information on supply chains, distribution networks, product models and ownership of websites,carry out unannounced onsite inspections and physical checks,enter any premises, land or means of transport an economic operator uses,acquire product samples without revealing their identity,instruct economic operators to take measures to end non-compliance or eliminate risk,prohibit or restrict availability of a product and order it be withdrawn or recalled,insist in case of serious risk that product content is removed from a website or require it be accompanied by a warning,adopt measures, including penalties, against economic operators that fail to act.
EU countries:
designate one or more authorities with the powers of market surveillance, investigation and enforcement;

appoint a single liaison office to represent the surveillance authorities and communicate the country’s national strategy;
ensure the authorities and office have sufficient budgetary and other resources;
may authorise surveillance authorities to reclaim from an economic operator all the costs they incur when pursuing non-compliance cases;
draw up an overarching national market surveillance strategy every 4 years from 16 July 2022 to promote a consistent, comprehensive and integrated approach to market surveillance — this must include: data on non-compliant products,priority areas for enforcing the EU legislation,enforcement activities to reduce non-compliance,assessment of cooperation between authorities in other EU countries;
provide economic operators, at their request and free of charge, information on national implementation of EU product harmonisation legislation;
introduce effective, proportionate and dissuasive penalties and notify these to the Commission by 16 October 2021.
The Commission:
ensures the Your Europe portal provides users with easy online access to information about the EU’s product requirements, rights, obligations and rules;
adopts implementing acts;
assists the EU Product Compliance Network (see below) in all its activities;
maintains a computer system to store and process all the data collected;
reports to the European Parliament, the Council and the European Economic and Social Committee by 31 December 2026, and every 5 years thereafter, on the regulation’s implementation.
Specific rules apply to imported products:
EU countries designate authorities with the necessary powers to check imports;
market surveillance authorities provide them with information on products and economic operators where a high risk of non-compliance has been identified;
authorities may impound a product if the necessary documentation is absent or there are concerns it presents a serious health, safety, environmental or public interest risk, and only release it when certain conditions are met.
The regulation:
establishes an EU Product Compliance Network — it: develops cooperation between the surveillance authorities and the Commission;contains national and Commission representatives and experts;organises a range of activities to improve market surveillance across the EU.
FROM WHEN DOES THE REGULATION APPLY?

It applies from 16 July 2021. However, Articles 29, 30, 31, 32, 33 (on the EU Product Compliance Network) and 36 (on financing activities) apply from 1 January 2021.
BACKGROUND

For more information, see:
Market surveillance for products (European Commission).
KEY TERMS

Market surveillance: measures to ensure products comply with EU legislation and protect the public interest.
Economic operator: manufacturer, authorised representative, importer, distributor or fulfilment service provider.
MAIN DOCUMENT

Regulation (EU) 2019/1020 of the European Parliament and of the Council of 20 June 2019 on market surveillance and compliance of products and amending Directive 2004/42/EC and Regulations (EC) No 765/2008 and (EU) No 305/2011 (OJ L 169, 25.6.2019, pp. 1-44)
RELATED DOCUMENTS

Regulation (EU) No 305/2011 of the European Parliament and of the Council of 9 March 2011 laying down harmonised conditions for the marketing of construction products and repealing Council Directive

89/106/EEC (OJ L 88, 4.4.2011, pp. 5-43)
Successive amendments to Regulation (EU) No 305/2011 have been incorporated into the original text. This consolidated version is of documentary value only.
Regulation (EC) No 765/2008 of the European Parliament and of the Council of 9 July 2008 setting out the requirements for accreditation and market surveillance relating to the marketing of products and repealing Regulation (EEC) No 339/93 (OJ L 218, 13.8.2008, pp. 30-47)
Directive 2004/42/EC of the European Parliament and of the Council of 21 April 2004 on the limitation of emissions of volatile organic compounds due to the use of organic solvents in certain paints and varnishes and vehicle refinishing products and amending Directive 1999/13/EC (OJ L 143, 30.4.2004, pp. 87-96)
See consolidated version.
last update 30.08.2019

C HUMAN EVALUATION FORM

Consent Form for Participation in Research

Title of Study:
Step Up Your Game: Summarizing Long Regulatory Documents Using a Two- or Multi-Step Method

Researcher:
Mika Sie
University Utrecht, Graduate School of Natural Sciences
AI MSc Thesis
m.s.y.sie@students.uu.nl

Introduction:
You are invited to participate in a research study from the Graduate School of Natural Sciences at University Utrecht. This form provides information about the study to help you decide if you would like to participate.

Purpose of the Study:
The purpose of this study is to research a two-or multi-step summarization technique for long, regulatory documents. This research compares different types of extractive summarization and abstractive summarization models.

Procedures:
If you agree to participate in this study, you will be asked to rate several summaries created by different architectures along four metrics and provide additional feedback on the quality of the summary. Your participation will take approximately one hour.

Voluntary Participation:
Your participation in this study is entirely voluntary. You may choose not to participate or withdraw from the study at any time without any penalty or loss of benefits to which you are otherwise entitled.

Anonymity and Confidentiality:
Your identity will remain anonymous. We will not collect any personal information that can be used to identify you. All data collected will be kept confidential and will be used for research purposes only. Results will be reported in aggregate form, and no individual responses will be identifiable.

Potential Risks:
The risks associated with participating in this study are minimal. However, if you feel uncomfortable at any point, you have the right to withdraw from the study.

Potential Benefits:

While there are no direct benefits to you for participating in this study, your participation will contribute to the understanding of automatic text summarization using Artificial Intelligence models.

Data Protection:
Your data will be stored securely and only accessible to the research team. We will comply with all applicable data protection laws and regulations. Data will be anonymized/pseudonymized and stored on password-protected servers.

Contact Information:
If you have any questions about this study, you may contact the principal researcher at:
Mika Sie
Graduate School of Natural Sciences
m.s.y.sie@students.uu.nl

For questions about your rights as a research participant, you may contact ics-ethics@uu.nl

Consent:
By signing below, you are indicating that you have read and understood the information provided above, and you agree to participate in this study. You understand that your participation is voluntary and that you can withdraw at any time without penalty.

Participant's Signature:

_____

Date: _____

Researcher's Signature:

_____

Date: _____

Evaluation of summaries

Evaluation Instructions

Thank you for agreeing to participate in this evaluation. In this section, you will be presented with several summaries generated by different architectures. Your task is to rate each summary based on the following metrics, using a scale of 1 to 5, where 1 indicates poor performance and 5 indicates excellent performance:

1. **Factual Correctness:**
    a. Evaluation of how factually correct the summary is relative to the source document.
2. **Usability:**
    a. Assessment of how practical and user-friendly the summary is.
3. **Accuracy:**
    a. Assessment of the precision and correctness of the information in the summary.
4. **Fluency:**
    a. Assessment of the summary's smoothness and ease of reading in terms of from, content and grammar.
5. **Coherence:**
    a. Measure of how logical the summary is to its linguistic context.

In addition to rating each summary, please provide any additional feedback or comments you have regarding the summaries. Your insights are valuable and will help improve the evaluation of the summarization architectures.

Summary 1

Carbon capture and accounting system (CBAM) for imports of greenhouse gases

Carbon capturing and accounting systems for imports into the EU

SUMMARY OF:

Regulation (EU) 2019/331 — the Carbon Capture and Accounting System

WHAT IS THE AIM OF THE REGULATION?

It sets out the rules for the establishment of a Carbon Capture & Accounting System (CBRAM) to prevent the risk of greenhouse gas (GHG) emissions being exported from the EU to non-EU countries.

KEY POINTS

Scope

The regulation applies to imports of GHG emission-intensive goods from the European Union (EU). It does not apply to:

goods produced in the EU for export to third countries which do not apply the EU's Emissions Trading System (ETS) or a similar carbon pricing mechanism;

other goods which are imported from outside the EU and which are subject to a carbon price that is equivalent to that applied to EU products.

Implementing acts

Each implementing act must be adopted in accordance with the examination procedure referred to in Article 29(2) of Directive 2003/87/EC (see summary).

The implementing acts must make the information in the CBAM registry available automatically and in real time to customs authorities and competent authorities.

The Commission must communicate the information to the competent authority of the EU country where the authorised CBAM declarant is established and cross-check that information with the data in the registry pursuant to Article 14 of the regulation.

Review of CBAM declarations

The competent authority must ensure that the declared number of certificates is correct and that the declarations are accurate.

If a CBAM declaration is incorrect, the Commission must assess the obligations under this regulation of that authorised declarant on the basis of the information it has at its disposal.

Where an authorised declrant fails to submit a declaration, or where the Commission considers that the number of declared certificates is incorrect or that the declaration of the number declared number is incorrect in relation to the quantity of emissions embedded in the goods, it must review the declaration and take any appropriate action.

FROM WHEN DOES THE RECOMMENDATION APPLY?

It has applied since 19 December 2019.

BACKGROUND

The EU has adopted a strategy to tackle climate change. The strategy aims to transform the EU into a fair and prosperous society, with a modern, resource-efficient and competitive economy, where there are no net emissions (emissions after deduction of removals) of greenhouse greenhouse gases ('greenhouse gas emissions') at the latest by 2050 and where economic growth is decoupled from the use of resources.
For more information, see:
'Climate action and climate action' on the European Commission's website.
MAIN DOCUMENT

Егракanска барка глаксанта бълка
Commission Delegated Regulation (EU, 2019/332 of 19 December 2018 determining transitional Union-wide rules for harmonised carbon capture & accounting systems and implementing implementing acts relating to the carbon capture and storage system for imports from the Member States of the Union for the period 2019-2023 (OJ L 130, 20.5.2019, pp. 1-2)
Successive amendments to Regulation (EEC, 2018/2067 have been incorporated in the original text. This consolidated version is of documentary value only.
RELATED DOCUMENTS
Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions on the implementation of Implementing Regulation (EC) No 596/2013 of the European Central Bank on the inclusion of the greenhouse gas capture and analysis system in the framework of the Green Deal (COM(2018) 2018/841) (COM) 2018-2067)
Communications from the Council and the European Committee of European Parliament and the Council of 30 May 2018 on the addition of greenhouse Gases to the EU Green Deal and on the harmonisation of the rules on greenhouse gas emission capture & analysis systems for the European Regional Development Fund (REFERENCES
Act
Entry into force - Date of expiry
Deadline for transposition into force
Official Journal L 130 of 21.6.2019
-
OJ C 130 of 20.6-20.2023
Amending act(s) No 2109/2014 of the Commission and of the Representatives of the Governments, meeting within the meaning of Article 29 of the Treaty on European Union, amending Implementing Regulations (EC, Euratom) No 652/2013, (EC), (ECB) No 1082/2013 and (EC)(EU) No 472/2013
Regulations (Euratom) 2020/852 and (EUR) No 852/2014 and (O

Metric Ratings

1. Factual Correctness: [1-5]
2. Usability: [1-5]
3. Accuracy: [1-5]
4. Fluency: [1-5]
5. Coherence: [1-5]

Additional Feedback:

[Your comments]

Summary 2

Carbon border adjustment mechanism — EU countries' contributions to the EU's greenhouse gas (GHG) emissions reduction targets

Carbonborder adjustment mechanism— EU countries' contributions towards the EU's GHG reductions targets

SUMMARY OF:

Regulation (EU) 2023/956 — establishing a carbon border adjustment system

WHAT IS THE AIM OF THE REGULATION?

It establishes a system for EU countries to contribute to reducing their greenhouse gas emissions (GHGs) by reducing their national GHG (greenhouse gas) emissions by at least 55 % below 1990 levels by the end of the year 2030.

It sets out the rules for the European Union (EU), the European Economic Area (EEA) and the European Free Trade Association (EFTA) countries.

KEY POINTS

The regulation sets out:

EU countries must reduce their GHG emissions by a minimum of 55 % by 2030 compared to 1990 levels compared to their national levels and by 2050 compared to the level of the 1990 level;

the EU countries must also reduce their emissions by an equivalent amount by 2020 compared with their 1990 levels.

The EU countries are required to submit a report to the European Commission every 5 years on the progress they have made towards this target. The Commission publishes the results of the report on its website.

Each year, the Commission publishes a report on the implementation of the regulation. The report is published in the Official Journal of the EEA and the Commission forwards it to the Council of the EU.

which adopts the conclusions of the European Parliament and the Council.

FROM WHEN DOES THE LEGULATION APPLACE ENTER INTO FORCE?

The legislation entered into force on 10 May 2023.

BACKGROUND

The European Commission adopted the European Green Deal on 11 December 2019. It aims to transform the EU into a fair and prosperous society, with a modern, resource-efficient and competitive economy where there are no net emissions (emissions after deduction of removals) of greenhouse gases ('greenhouse gases') at the latest by 2050 and where economic growth is decoupled from the use of resources.

For more information, see:

European Green Deal (European Commission).

KEY TERMS

* COVID-19 pandemic: a global warming pandemic caused by the COVID 19 pandemic in the early 1990s that had a major impact on the health and economic well-being of EU citizens.
* GHG: a greenhouse gas emitted by burning fossil fuels, such as coal, oil and natural gas, which has the potential to increase the temperature of the planet by up to 2.5 °C above pre-industrial levels. It is also known as the 'carbon dioxide' (CO2) pandemic.

REFERENCES

Act

Entry into force - Date of expiry

Deadline for transposition in the Member States

Official Journal

Decision (EU, Euratom) 2018/852 of the Council on the conclusion of the Framework Convention on Climate Change (UNFCCC) (the 'Glasgow Climate Pact' and of the Agreement on the Paris Agreement (the Paris Agreement) (Official Journal L 130 of 16.5.2018)

Decisions (EU/EU) 2020/853 of the Parliament and (Euratom) 2021/854 of the Commission on the signing, on behalf of the Union, of the Kyoto Protocol (the Kyoto Protocol) on climate change and on the signature of the Protocol on the European Neighbourhood Policy on the Green Deal for the Implementation of the Treaty on the Functioning of the Single European Union by the European Atomic Energy Community (see summary).

Council Decision (EU-EU) 2019/856 of 20 May 2019 on the adoption of the Copenhagen Agreement on climate action by the EU and the conclusion by the Union of the United Nations Framework Convention for the Prohibition of Chemical Weapons (the UNFCCC).

last update 25.01.2022

Metric Ratings

1. Factual Correctness: [1-5]
2. Usability: [1-5]
3. Accuracy: [1-5]
4. Fluency: [1-5]
5. Coherence: [1-5]

Additional Feedback:

[Your comments]

Summary 3

Carbon border adjustment mechanism (CBAM) for greenhouse gas emissions embedded in imported goods

Carbon base adjustment mechanism for greenhouse gases embedded in imported goods

SUMMARY OF:

Regulation (EU) 2018/2067 — carbon border adjustment scheme

WHAT IS THE AIM OF THE REGULATION?

It establishes a system for calculating greenhouse gas (GHG) emissions embedded into imported goods.

It aims to prevent GHG emissions from leakage into the EU's internal market.

KEY POINTS

Scope

The regulation applies to goods listed in Annex I on their importation into the customs territory of the EU.

Annex II only direct emissions are calculated and taken into account.

Importers must submit a certificate to the competent authority of the Member State where the authorised CBAM declarant is established.

The certificate must be verified by a person accredited by a national accreditation body appointed in accordance with Regulation (EC) No 765/2008 of the European Parliament and of the Council (12) or (13) or pursuant to Directive 2003/87/EC (see summary).

The declaration of emissions must be accompanied by a carbon price that is equivalent to that applied to EU products, and the declaration of the actual emissions.

Authorised CBAM Declarant

The authorised declarant must submit the repurchase request by 30 June of each year during which the CBAM certificates were surrendered.

If the Commission finds that the number of CBAM certificate in the account of an authorised declarant does not comply with the obligations pursuant to paragraph 2, it must inform the authority of that Member State.

Penalties

The penalty amount for the failure to surrender CBAM Certificates is identical to the amount for failing to surrender certificates pursuant to Article 16(3) and (4) of Directives 2003/86/EC and 2003/93/EC.

Exemptions

The Regulation does not apply to:

goods covered by the EU Emissions Trading System (ETS) or to goods covered by Regulation (EU, Euratom) No 652/2013 (see overview);

good-goods that are not covered by Directive 2009/43/EC or Regulation (Euratom) no 552/2012 (see description);

products covered by Article 7(2) of Directive 2003(87) of the EC and which are not subject to the EU ETS.

In addition, the Commission may present a report to the European parliament and to the Council that identifies products further down the value chain of the goods it considers to be considered for inclusion within the scope of this regulation.

DATE OF ENTRY INTO FORCE

It has applied since 30 June 2018.

BACKGROUND

For more information, see:

‘Embedded emissions’ on the European Commission's website.

REFERENCES

Act	Entry into force - Date of expiry	Deadline for transposition in the Member States	Official Journal
Commission Implementing Regulation (U.S.A. 2019	30.6.2018	-	OJ L 130 of 16.5.2019

RELATED ACTS

Regulations (E.U. 2015/856 and (E) 2015/966 of the Commission and Council Directives 2009/44/EC, 2009/45/EC of the Parliament and the Council and Directives 2004/34/EC on the harmonisation of the laws, regulations, administrative provisions and administrative provisions of Member States relating to the emission trading system, the European System of Central Banks and the European Customs Union (ECCBAM), the European Convention on the Law of the Sea and the Protocol on the Harmonization of the Laws, Regulations on the Approximation of the Emissions Trade System, the Rules of the Customs Union, the Regulations of the Court of Justice and the High Representative of the Union for Foreign Affairs and Security Policy, the Treaty on the Functioning of the United Kingdom and the Treaty of Lisbon (OJ C 202, 17.6.).

Directive 2009/42/EC establishes a carbon border adjust mechanism (the ‘CBAM’) to address greenhouse gas emission embedded in the goods imported into the Union in order to prevent the risk of carbon leakage, thereby reducing global carbon emissions and supporting the goals of the Paris Agreement, also creating incentives for the reduction of emissions by operators in non-EU countries.

See also:

EU Emissions trading system (European Commission).

last update 04.02.2021

Metric Ratings

1. Factual Correctness: [1-5]
2. Usability: [1-5]
3. Accuracy: [1-5]
4. Fluency: [1-5]
5. Coherence: [1-5]

Additional Feedback:

[Your comments]

Summary 4

EU carbon capture and storage system (CBAM) for imports and exports of goods from non-EU countries (until 2023)

EU carbon capturing and storage systems (BCAM)

SUMMARY OF:

Regulation (EU) 2018/2067 on the establishment of the European Carbon Capture and Storage System

WHAT IS THE AIM OF THE REGULATION?

It establishes a system for importing and exporting carbon-capturing and storing products from third countries (LDCs) to the EU.

It aims to prevent carbon leakage and reduce greenhouse gas emissions in LDCs.

KEY POINTS

The regulation sets out the rules for the establishment and operation of the EU's carbon capture, storage and management system, known as the CBAM.

The system is designed to:

ensure that imported products are subject to a regulatory system that applies carbon costs equivalent to those borne under the EU ETS, resulting in a carbon price that is equivalent for imports* and domestic products;

encourage LDC countries to reduce their emissions of greenhouse gases (GHG) by reducing their GHG emissions by up to 40% by 2050;

ensures that imports of energy-intensive products from non EU countries are subject, at the latest, to a carbon tax that is equal to those applicable to imports of domestic products.

To achieve these objectives, the regulation sets up a system to register the number of CBAM certificates issued by authorised CBAM declarants (certificates issued by the EU countries to the European Commission.

Each CBAM certificate holder is authorised by the competent authorities in each EU country.

each CBAM holder must keep an account in the registry.

the CBAM registry contains information on:

the number of certificates issued,

the certificates issued and

the total number of installations in the EU that have been certified

the estimated number of installed installations in each installation

the actual emissions of complex goods produced in a given installation.

In addition, the registry contains a list of the countries and territories which have been removed from the list in point 2 of Annex III of the regulation.

Annex IV sets out methods for calculating embedded emissions for the purpose of calculating the embedded emissions.

FROM WHEN DOES THE RECOMMENDATION APPLYEAR OF ENTRY INTO FORCE?

It entered into force on 1 January 2019.

BACKGROUND

The EU has adopted a strategy to transform the EU into a fair and prosperous society, with a modern, resource-efficient and competitive economy, where there are no net emissions (emissions after deduction of removals) of GHG greenhouse gases by 2050 and where economic growth is decoupled from the use of resources.

For more information, see:

‘Greenhouse Gas Emissions from Non-EU Countries’ (European Commission).

KEY TERMS

* Carbon capture & storage system: a system designed to capture, store and store carbon emissions from imported and exported goods.

* Embedded emissions: the emissions of the input materials (precursors) consumed in the production process, which are then used in the final product.

REFERENCES

Act	Entry into force - Date of expiry	Deadline for transposition in the Member States	Official Journal
Commission Implementing Regulation(EU) 2019/2065	1.1.2023	-	OJ L 130 of 21.5.2019

RELATED ACTS

Regulations (EU, Euratom) 2019-2023 of the Parliament and of the Council of 27 April 2019 on the harmonisation of the laws, regulations and administrative provisions of Member States relating to the import and export of carbon-absorbing and storing systems (Official Journal L 130, 31.4.2019, pp. 1-8)

Regulating (EU), Euratom, Euratomic, Euronews and Euratom of 27 March 2019 on a harmonised system for the import of carbon dioxide emissions from certain industrial sectors and repealing Directive 95/46/EC (General Data Protection Regulation) and repealing Regulation (EC) No 552/2004 (Protection Regulation) (Official Journal L 119, 4.3.2019).

See consolidated version.

last update 14.02.2021

Metric Ratings

1. Factual Correctness: [1-5]
2. Usability: [1-5]
3. Accuracy: [1-5]
4. Fluency: [1-5]
5. Coherence: [1-5]

Additional Feedback:

[Your comments]

Summary 5

Carbon border adjustment mechanism (CBAM) Carbon border adjustment mechanism (CBAM) SUMMARY OF: Regulation (EU) 2023/956 establishing a carbon border adjustment mechanism WHAT IS THE AIM OF THE REGULATION? It establishes a carbon border adjustment mechanism (CBAM) to address greenhouse gas emissions embedded in the goods imported into the EU in order to prevent the risk of carbon leakage, thereby reducing global carbon emissions and supporting the goals of the Paris Agreement. The CBAM complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the ‘EU ETS’) by applying an equivalent set of rules to imports into the customs territory of the EU of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the ‘EU ETS’) by applying an equivalent set of rules to imports into the customs territory of the EU of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the ‘EU ETS’) by applying an equivalent set of rules to imports into the customs territory of the EU of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the ‘EU ETS’) by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the ‘EU ETS’) by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the ‘EU ETS’) by applying an equivalent set of rules

to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM also complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The

CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The CBAM complements the system for greenhouse gas emission allowance trading within the EU established under Directive 2003/87/EC (the 'EU ETS') by applying an equivalent set of rules to imports into the customs territory of the goods referred to in Article 2 of this regulation. The CBAM is set to in Article 2 of this regulation to in Article 2 of this regulation. The CBAM is set to replace the mechanisms established under Directive 2003/87/EC to prevent the risk of carbon leakage by reflecting the extent to which EU ETS allowances are allocated free of charge in accordance with Article 10a of that directive. The

Metric Ratings

1. Factual Correctness: [1-5]
2. Usability: [1-5]
3. Accuracy: [1-5]
4. Fluency: [1-5]
5. Coherence: [1-5]

Additional Feedback:

[Your comments]

Summary 6:

Carbon Border Adjustment Mechanism (CBAM) — rules for the inclusion of goods in its scope

Carbon Border Adjustment Mechanism (CBAM) — rules for the inclusion of goods in its scope

The European Union (EU) is taking action to tackle climate change. The Carbon Border Adjustment Mechanism (CBAM) is one of the measures to be implemented in the EU in order to combat climate change.

WHAT DOES THE REGULATION DO?

It establishes the Carbon Border Adjustment Mechanism (CBAM), which is an EU-wide scheme that aims to prevent carbon leakage, that is, the risk that some industries in the EU continue to emit greenhouse gases while their competitors in other countries do not.

The regulation applies from 2023.

KEY POINTS

Scope

The CBAM applies to the following goods:

cement, iron and steel, aluminium, fertilisers, nitrogen oxides, and wet distillates (fuels).

It also applies to electricity imported from third countries (i.e. non-EU countries).

The regulation provides for the possibility to extend the scope of the CBAM to additional goods and services, such as plastic, textiles, glass and ceramics, as well as to the indirect emissions embedded in the goods listed in Annex II to the regulation.

Procedure for imports

Goods may be imported only by an authorised CBAM declarant who must:

ensure that the goods are imported into the customs territory of the EU;

keep records of the information needed to calculate the embedded emissions;

calculate the embedded emissions;

verify the embedded emissions;

pay the carbon price in the third country of origin;

report the embedded emissions in a CBAM declaration;

surrender CBAM certificates.

Goods may be imported only by an authorised CBAM declarant. The authorised CBAM declarant must be an importer established in a Member State. The authorised CBAM declarant must apply for the status of authorised CBAM declarant before importing the goods into the customs territory of the EU.

Embedded emissions

Embedded emissions are emissions from the production and use of goods. The regulation provides for the following rules for calculating embedded emissions:

for goods other than electricity, embedded emissions are determined based on actual emissions, in accordance with the methods set out in Annex IV to the regulation;

for electricity, embedded emissions are determined based on default values set by the Commission.

Carbon price

The regulation provides for the following rules for the carbon price:

the carbon price is paid in the third country of origin;

the carbon price is calculated as the average price of CBAM certificates;

the authorised CBAM declarant pays the carbon price by surrendering CBAM certificates;

the authorised CBAM declarant pays the carbon price by surrendering CBAM certificates.

CBAM certificates

The regulation provides for the following rules for the CBAM certificates:

the authorised CBAM declarant may sell CBAM certificates;

the authorised CBAM declarant must surrender CBAM certificates;

the authorised CBAM declarant may repurchase CBAM certificates;

the authorised CBAM declarant may cancel CBAM certificates;

the Commission calculates the average price of CBAM certificates;

the Commission publishes the average price of CBAM certificates;

the Commission may adopt implementing acts on the application of the methodology for calculating the average price of CBAM certificates and the practical arrangements for the publication of that price;

the Commission may adopt implementing acts on the application of the methodology for calculating the average price of CBAM certificates and the practical arrangements for the publication of that price;

the Commission may adopt implementing acts on the application of the methodology for calculating the average price of CBAM certificates and the practical arrangements for the publication of that price.

Registration of operators and installations in third countries

The regulation provides for the following rules for the registration of operators and installations in third countries:

an operator must register in the CBAM registry;

the Commission may deregister an operator;

the authorised CBAM declarant may disclose information on the verification of embedded emissions to an operator;

the authorised CBAM declarant may withdraw the registration of an operator;

the Commission may deregister information on an operator;

the Commission may deregister information on an operator;

the Commission may deregister information on an operator;

the Commission may deregister information on an operator;

the Commission may deregister information on an operator;

the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;

the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister information on an operator;
the Commission may deregister

Metric Ratings

1. Factual Correctness: [1-5]
2. Usability: [1-5]
3. Accuracy: [1-5]
4. Fluency: [1-5]
5. Coherence: [1-5]

Additional Feedback:

[Your comments]

D VISUALISATION OF RESULTS RATIO TYPES

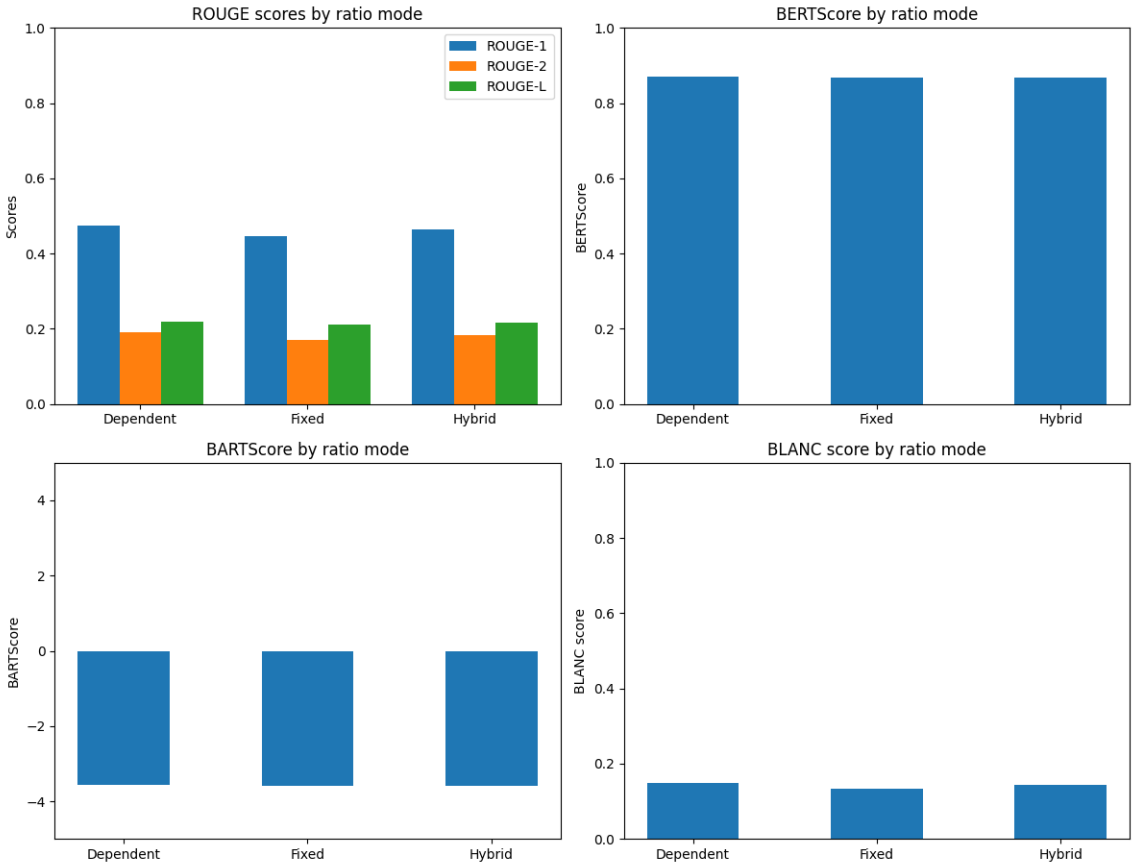


Fig. 10. Evaluation metrics by ratio mode.

E HUMAN EVALUATION RESULTS

Summary 1

Participant 1:

Metric Ratings

1. Factual Correctness: 1
2. Usability: 1
3. Accuracy: 1
4. Fluency: 1
5. Coherence: 1

Additional Feedback:

This summary would not be usable, because if someone that has no knowledge in the topic would read this, they would not understand a thing about the actual regulation, due to the incompleteness, occurrence of factual mistakes and inaccuracy.

Participant 2:

Metric Ratings

1. Factual Correctness: [1-5] → 3
2. Usability: [1-5] → 3
3. Accuracy: [1-5] → 2
4. Fluency: [1-5] → 2
5. Coherence: [1-5] → 3

Additional Feedback:

[Your comments]

Touches upon main principle of CBAM, but some of the procedures/rules are described incorrectly.

Summary 2

Participant 1:

Metric Ratings

1. Factual Correctness: 3
2. Usability: 1
3. Accuracy: 1
4. Fluency: 2
5. Coherence: 1

Additional Feedback:

Summary does not per se contain false information, however it misplaces information (in a false manner). For example: in the 'key points' it describes background information of the regulation instead of the main content of the regulation. This summary would not be usable for readers.

Participant 2:

Metric Ratings

1. Factual Correctness: [1-5] → 4
2. Usability: [1-5] → 1
3. Accuracy: [1-5] → 3
4. Fluency: [1-5] → 4
5. Coherence: [1-5] → 2

Additional Feedback:

[Your comments]
The summary completely misses out on the main point of what CBAM is.
State information appears to be correct (few mistakes).

Summary 3

Participant 1:

Metric Ratings

1. Factual Correctness: 4
2. Usability: 3
3. Accuracy: 3
4. Fluency: 2
5. Coherence: 2

Additional Feedback:

This summary is actually quite useful: it correctly grasps the key points of the regulation. It is not fully complete, and the fluency and coherence of the sentences are a bit lacking, but this summary is a good starting point.

Participant 2:

Metric Ratings

1. Factual Correctness: [1-5] → 4
2. Usability: [1-5] → 4
3. Accuracy: [1-5] → 3
4. Fluency: [1-5] → 4
5. Coherence: [1-5] → 4

Additional Feedback:

[Your comments]

Summary 4

Participant 1:

Metric Ratings

1. Factual Correctness: 1
2. Usability: 1
3. Accuracy: 1
4. Fluency: 1
5. Coherence: 1

Additional Feedback:

This summary is less bad than Summary 1, but still unusable as it contains a lot of false information/incorrect words.

Participant 2:

Metric Ratings

1. Factual Correctness: [1-5] → 3
2. Usability: [1-5] → 3
3. Accuracy: [1-5] → 4
4. Fluency: [1-5] → 2
5. Coherence: [1-5] → 3

Additional Feedback:

[Your comments]

Summary 5:

Participant 1:

Metric Ratings

1. Factual Correctness: 4
2. Usability: 1
3. Accuracy: 2
4. Fluency: 1
5. Coherence: 1

Additional Feedback:

The summary started really well, however it started repeating the same sentence over and over at some point. Therefore not usable.

Participant 2:

Metric Ratings

1. Factual Correctness: [1-5] → 3
2. Usability: [1-5] → 1
3. Accuracy: [1-5] → 3
4. Fluency: [1-5] → 5
5. Coherence: [1-5] → 1

Additional Feedback:

[Your comments]
A lot of repetition

Summary 6:

Participant 1:

Metric Ratings

1. Factual Correctness: 3
2. Usability: 1
3. Accuracy: 2
4. Fluency: 1
5. Coherence: 1

Additional Feedback:

Again unusable due to the sentences that are repeated. However, some of the sentences that are not repeated contain quite some useful information.

Participant 2:

Metric Ratings

1. Factual Correctness: [1-5] → 3
2. Usability: [1-5] → 4
3. Accuracy: [1-5] → 3
4. Fluency: [1-5] → 4
5. Coherence: [1-5] → 3 (malus for repetition at end, otherwise 4)

Additional Feedback:

[Your comments]

Table 19. Human evaluation results participant 1.

Architecture #	Extractive model	Ratio type	Abstractive model	Factual correctness	Usability	Accuracy	Fluency	Coherence
1	RoBERTa	Dependent	BART	1	1	1	1	1
2	-	No extraction	BART	3	1	1	**2**	1
3	LexLM	Dependent	BART	**4**	**3**	**3**	**2**	**2**
4	Longformer	Dependent	BART	1	1	1	1	1
5	-	No extraction	PegasusX	**4**	1	2	1	1
6	RoBERTa	Dependent	Llama3	3	1	2	1	1

Table 20. Human evaluation results participant 2.

Architecture #	Extractive model	Ratio type	Abstractive model	Factual correctness	Usability	Accuracy	Fluency	Coherence
1	RoBERTa	Dependent	BART	3	3	2	2	3
2	-	No extraction	BART	**4**	1	3	4	2
3	LexLM	Dependent	BART	**4**	**4**	3	4	**4**
4	Longformer	Dependent	BART	3	3	**4**	2	3
5	-	No extraction	PegasusX	3	1	3	**5**	1
6	RoBERTa	Dependent	Llama3	3	4	3	4	3