

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
кафедра математики

Кваліфікаційна робота
освітній ступінь – бакалавр
на тему:

**“LLM-BASED SMART CHATBOT DESIGN USING LANGCHAIN
FRAMEWORK”
«ПОБУДОВА РОЗУМНОГО ЧАТ-БОТУ НА ОСНОВІ ВЕЛИКИХ
МОВНИХ МОДЕЛЕЙ ТА ФРЕЙМВОРКУ LANGCHAIN»**

Виконала: студентка 4-го року навчання,
Освітньої програми

«Прикладна математика», 113

Колодяжна Дарія Олександрівна

Керівник Кузьменко Д. О.,

ст. викладач

Рецензент к.ф.м-н, доцент Швай Н.О.

Кваліфікаційна робота захищена

з оцінкою _____

Секретар ЕК _____

« ____ » _____ 2024 р.

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
кафедра математики

ЗАТВЕРДЖУЮ

Зав.кафедри математики,
проф., д.ф.-м.н.

_____ Р. К. Чорней
(підпис)

«__» _____ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на кваліфікаційну роботу

Студентці Колодяжній Дарії Олександрівні факультету інформатики 4 курсу
ТЕМА: LLM-based Smart Chatbot Design Using LangChain Framework/Побудова
розумного чат-боту на основі великих мовних моделей та фреймворку
LangChain

План кваліфікаційної роботи:

1. Introduction
2. Tools and Techniques
3. Experiments and Results
4. Conclusions

Дата видачі „__” _____ 2023 р. Керівник _____ (підпис)

Завдання отримав _____ (підпис)

Календарний план виконання роботи

Тема: LLM-based Smart Chatbot Design Using LangChain Framework/Побудова розумного чат-боту на основі великих мовних моделей та фреймворку LangChain

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на бакалаврську роботу	27.10.2023	
2.	Опрацювання матеріалів	31.01.2024	
3.	Створення програми	01.05.2024	
4.	Написання теоретичної частини	14.05.2024	
5.	Коригування роботи	27.05.2024	

Студент: Колодяжна Д. О

Керівник: Кузьменко Д. О.

“ _____ ”

TABLE OF CONTENTS

TABLE OF CONTENTS	4
1 INTRODUCTION.....	6
2 FUNDAMENTALS OF LARGE LANGUAGE MODELS	9
2.1 EVOLUTION OF LANGUAGE MODELS	9
2.2 TRANSFORMERS	10
2.2.1 <i>Attention Mechanism</i>	10
2.2.2 <i>Positional Encoding</i>	12
2.2.3 <i>Transformer architecture</i>	12
2.3 MISTRAL 7B.....	14
2.3.1 <i>Sliding window attention</i>	14
2.3.2 <i>Rolling buffer cache</i>	15
3 TOOLS AND TECHNIQUES	17
3.1 LANGCHAIN	17
3.2 RETRIEVAL-AUGMENTED GENERATION	17
3.2.1 <i>Concept of RAG</i>	18
3.2.2 <i>Key Copmonents of RAG</i>	19
3.2.3 <i>Advantages of RAG method:</i>	20
4 EXPERIMENTS AND RESULTS	21
4.1 DATASET	21
4.2 RESOURCES	21
4.3 MODEL SELECTION	21
4.4 EXPERIMENTAL FLOW	21
4.4.1 <i>Embedding Creation</i>	21
4.4.2 <i>Retrieval techniques</i>	22
4.4.3 <i>Prompt Engineering</i>	23
4.4.4 <i>Resulting Chain</i>	24

4.4.5 <i>Different Approaches for specified tasks</i>	24
4.4.6 <i>UX\UI</i>	25
4.4.7 <i>Implementation challenges</i>	26
4.5 ACHIEVED RESULTS	26
4.6 ADVANTAGES OF THE WORK:	26
4.7 LIMITATIONS OF THE WORK:	27
5 CONCLUSIONS	28
REFERENCES	29
APPENDIX A	32
APPENDIX B	34

ANNOTATION

This work presents the development of a smart chatbot called "Legislative Assistant" that assists Ukrainian lawyers and citizens in navigating legislative acts. The chatbot utilizes large language models (LLMs), specifically the Mistral-7B-Instruct-Ukrainian model, integrated with the LangChain framework and retrieval-augmented generation (RAG) techniques. By creating embeddings from a corpus of Ukrainian legislative acts and optimizing retrieval using RAG, the chatbot provides accurate and up-to-date responses to queries related to legislation. This work covers theoretical aspects such as the evolution of language models, transformer architectures like Mistral 7B and the concept of retrieval-augmented generation (RAG) for incorporating external data into language model responses.

Key words: chatbots, large language models, Ukrainian legislation, retrieval-augmented generation, LangChain, Mistral 7B, attention mechanism.

1 INTRODUCTION

Large language models (LLMs) are powerful AI models with vast knowledge and language generation capabilities, with potential to revolutionize human-machine interactions. However, harnessing LLMs requires innovative approaches bridging these models to real-world applications.

This work focuses on developing a smart chatbot using LLMs and LangChain framework. The integration of these technologies creates an architecture that addresses the demand for intelligent conversational systems that can understand complex queries and provide accurate responses, especially for languages like Ukrainian where advanced language technologies are limited. Ukrainian is underrepresented in natural language processing (NLP) due to a lack of extensive annotated datasets and limited research compared to more widely spoken languages. Improving language models for Ukrainian is crucial for preserving linguistic diversity and ensuring that speakers of all languages have access to advanced technological solutions.

During this research, we created a chatbot agent to assist Ukrainian lawyers and other citizens in navigating legislative acts. The "Legislative Assistant" chatbot provides a user-friendly interface to explore legal frameworks, promoting accessibility and an informed citizenry.

Our work first explores the theoretical foundations of LLMs. We examine the attention mechanism that lays inside majority of modern LLMs, enabling them to capture global dependencies and facilitate parallelization. First section focuses on Mistral 7B model, a cutting-edge transformer-based architecture that incorporates innovative techniques such as sliding window attention and rolling buffer caching to overcome the computational limitations of traditional attention mechanisms.

We address the limitations of traditional language models by exploring retrieval-augmented generation (RAG). This innovative approach allows LLMs to use external data sources, such as legislative acts, to generate accurate responses. By

incorporating up-to-date information, the chatbot minimizes errors and ensures relevance.

The experimental section describes how the chatbot was developed, including selecting the Mistral-7B-Instruct-Ukrainian model and creating embeddings from legislative acts. It also explains the prompt engineering process, which guides the model's responses for coherence and relevance.

This work combines the power of LLMs, the flexibility of LangChain framework, and the innovative RAG approach. The resulting chatbot, "Legislative Assistant," shows how these technologies can make complex legal information more accessible and understandable.

2 FUNDAMENTALS OF LARGE LANGUAGE MODELS

Language Models (LMs) are computational models designed to understand and generate human language text. At their core, LMs capture the statistical and structural properties of language, enabling them to predict the likelihood of a sequence of words occurring in a given context. The primary goal of LMs is to learn the patterns, relationships, and dependencies within a language corpus, thereby facilitating tasks such as text generation, completion, translation, summarization, and sentiment analysis.

2.1 Evolution of Language Models

The progression of Large Language Models (LLMs) stands as a significant culmination of extensive research [1] and development in the field of language modeling. This advancement can be dissected into four distinct phases:

Statistical Language Models (SLMs): Emerged in the 1950s, statistical language modeling found its roots in Claude Shannon's pioneering work [2] applying information theory to human language. Shannon's approach involved assessing language predictability and compression using n-gram language models [3], notably the Markov chain models.

Neural Language Models (NLMs): Early neural language models [4], [5], [6], [7] surfaced as a response to the limitations observed in statistical models. These models leveraged neural networks to map words to low-dimensional continuous vectors (embedding vectors) and predict subsequent words based on aggregating these vectors.

Pre-trained Language Models (PLMs): Departing from task-specific methodologies, PLMs [8], [9], [10] adopted pre-training and fine-tuning strategies. Models underwent pre-training on extensive, unlabeled text corpora to perform general tasks such as word prediction. Subsequently, they were fine-tuned for specific tasks using labeled data.

Large Language Models (LLMs): Representing the latest evolution in language modeling, LLMs are distinguished by their monumental scale, housing tens to hundreds of billions of parameters [11], [12], [13]. Based mostly on transformer architectures, these models undergo pre-training on extensive text corpora, showcasing superior language understanding and generation capabilities compared to PLMs. They demonstrate emergent abilities like in-context learning, instruction following, and multi-step reasoning, enabling them to adeptly handle complex tasks and engage effectively with users and environments.

2.2 Transformers

In recent years, there has been a significant shift in the field of sequence modeling and transduction towards innovative architectures like transformers [14], [15], [16]. Transformer model is an architecture that entirely relies on attention mechanisms, discarding traditional recurrent or convolutional layers. The key idea behind it is to promote global dependencies between input and output sequences through attention mechanisms, facilitating parallelization and significantly reducing training time without compromising quality.

2.2.1 Attention Mechanism

The attention [17] allows the model to selectively focus on relevant parts of the input sequence when generating the output, mimicking the human ability to attend to important information selectively. The attention mechanism can be formalized mathematically as follows:

Let us consider a sequence of input tokens $x_1, x_2, \dots, x_n \in R$, represented as vectors of dimension n_{model} . The goal is to compute an output vector y_i corresponding to each input vector x_i , while taking into account the entire input sequence.

First, we project the input vectors onto three different vector spaces using learned weight matrices W_q , W_k , and W_v to obtain query vectors q_i , key vectors k_i , and value vectors v_i , respectively:

$$q_i = x_i W_q, \quad k_i = x_i W_k, \quad v_i = x_i W_v$$

The attention weights $\alpha_{i,j}$ are then computed as the scaled dot product between the query vector q_i and the key vectors k_j of all previous input tokens, followed by a softmax activation:

$$\alpha_{i,j} = \text{softmax}\left(\frac{q_i k_j^\top}{\sqrt{d_k}}\right) = \frac{\exp\left(\frac{q_i k_j^\top}{\sqrt{d_k}}\right)}{\sum_{l=1}^i \exp\left(\frac{q_i k_l^\top}{\sqrt{d_k}}\right)}$$

To ensure that the dot-products between query and key do not cause numerical overflow (that leads to small values of softmax gradient), we divide by $\sqrt{d_k}$.

The output vector y_i is then computed as a weighted sum of the value vectors v_j , with the attention weights $\alpha_{i,j}$ as the coefficients:

$$y_i = \sum_{j=1}^i \alpha_{i,j} v_j$$

For efficient computation, these operations can be expressed as matrix multiplications. Let X be the input sequence matrix (we form a matrix by stacking all input vectors on top of each other), and $Q = XW_q$, $K = XW_k$ and $V = XW_v$ be the corresponding query, key, and value matrices. Then, the attention weights can be computed as:

$$A = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) \in R^{n \times n}$$

And the output matrix $Y \in R^{n \times d_v}$ can be computed as:

$$Y = AV$$

The time complexity of the self-attention mechanism in the transformer architecture is quadratic, $O(n^2)$, with respect to the length of the input sequence. This can be attributed to the scaled dot-product attention operation, which involves computing the attention scores between every pair of positions in the input sequence.

2.2.2 Positional Encoding

Since the transformer does not inherently encode the order of tokens, a positional encoding [18] is added to the input embeddings. This encoding uses sinusoidal functions to represent the position of each token, enabling the model to capture sequential information.

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

Here, pos represents the position in the sequence, and i denotes the dimension within the positional encoding vector. The choice of 10,000 as the base for the exponential function and the use of both sine and cosine functions are empirical decisions that have proven effective in practice.

After computing the positional encoding vectors using the formula, they are added element-wise to the input embeddings. This addition integrates the positional information with the semantic information contained in the embeddings, resulting in embeddings that encompass both types of information.

2.2.3 Transformer architecture

Transformer neural networks organize attention layers and other network components in diverse configurations. Multiple attention layers, referred to as attention heads, operate in parallel to create multi-headed attention. Each attention head possesses its own set of independent training parameters. The outputs of these attention heads, represented as matrices with dimensions $n \times d_v$, are concatenated to

form a matrix of dimension $n \times h \cdot d_v$. This concatenated matrix is then reshaped by multiplying it with another trained matrix, $W_O \in \mathbb{R}^{h \cdot d_v \times n_{\text{model}}}$:

$$Y = [Y_1, \dots, Y_h] \cdot W_O \in \mathbb{R}^{n \times n_{\text{model}}}$$

Together with normalization layers, feed-forward layers, and residual connections, multi-headed attention constitutes a transformer block. The input and output of a transformer block share the same shape, allowing them to be connected sequentially (Fig.1).

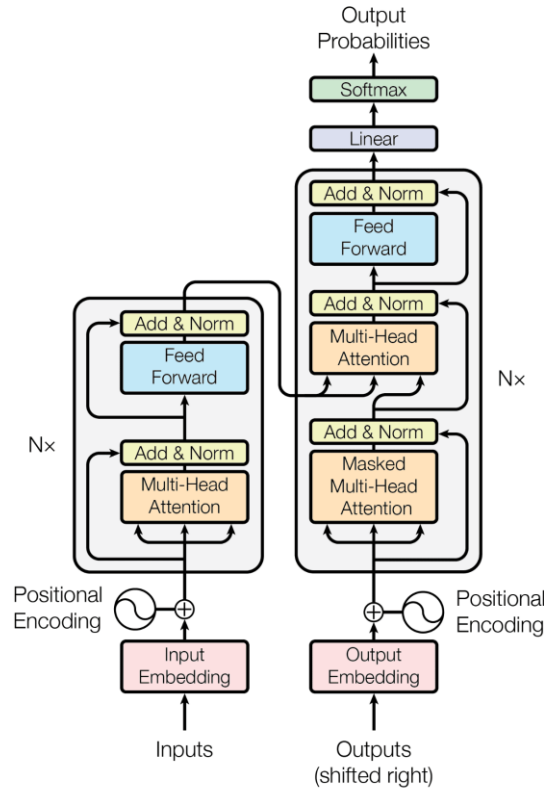


Figure 1. The Transformer architecture.

In the decoder, a masking operation is applied to prevent attending to future tokens during the autoregressive generation process. This ensures that the model generates the output sequence one token at a time, based on the previously generated tokens and the encoder's representation. During the training of a neural network as a language model for predicting the subsequent word, the upper triangular section of the scalar product matrix A signifies "future" information and is not suitable for use.

Consequently, the upper right quadrant of the matrix is masked, where the values are adjusted to $-\infty$. With the convention $\exp(-\infty) = 0$, these modified values do not influence the softmax computation.

2.3 Mistral 7B

Mistral 7B [19] is a state-of-the-art large language model, based on transformer architecture with some additional improvements. From the basic transformer architecture it inherits only decoder part.

Key features of Mistral 7B are considered below.

2.3.1 Sliding window attention

The Mistral 7B model employs a sliding window attention (SWA) [19] mechanism to address the quadratic computational complexity of the standard self-attention mechanism. SWA limits the attention span of each layer to a fixed window size, reducing the computational cost from $O(n^2)$ to faster one, where n is the sequence length (Figure 2).

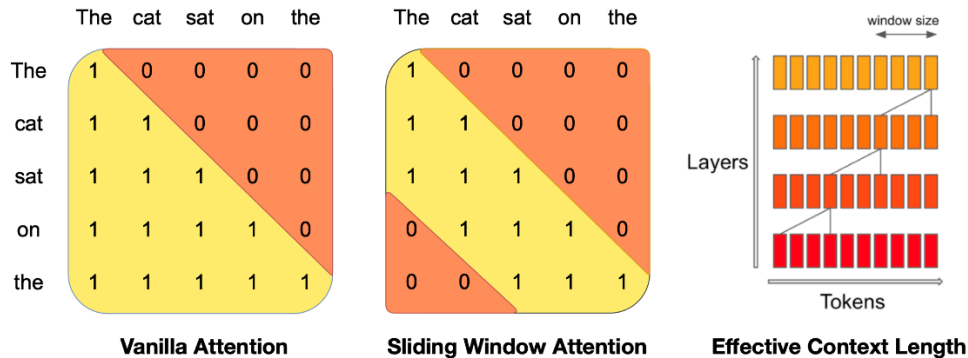


Figure 2: Sliding Window Attention

SWA is an attention mechanism that takes advantage of the transformer's layered design, allowing each layer to access data beyond a set window size, W . Specifically, a hidden state at a particular position within one layer can refer to multiple positions in the preceding layer. This recursive process enables the hidden state to indirectly reach information from the input layer up to $W \times k$ positions away, where k is the number of layers. With a window size of $W = 4096$, the theoretical

attention span extends to about 131K tokens. Due to various optimizations, this method doubles the processing speed for sequences of 16K tokens when compared to the standard attention mechanism.

2.3.2 Rolling buffer cache

A rolling buffer cache [19] is a memory management technique that maintains a fixed-size cache by continually overwriting the oldest data with new data. This ensures that the cache does not grow beyond its specified limit. When new information comes in, it is stored in the cache, and the oldest entries are removed to make room. This method helps manage memory efficiently, as it limits the amount of data stored at any given time while keeping recent and relevant information readily accessible (Figure 3).

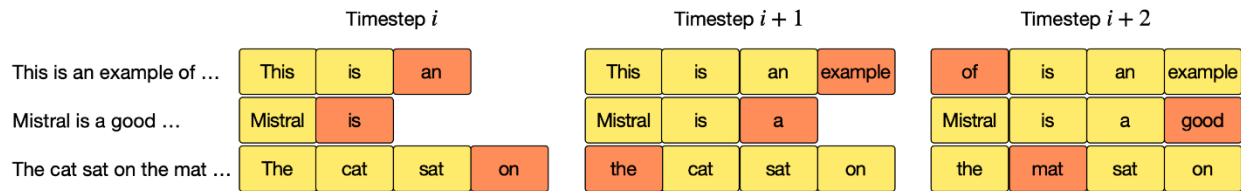


Figure 3: Rolling Buffer Cache

3. Pre-fill and Chunking

Pre-fill and chunking are techniques used to optimize the process of generating text in transformer models. Pre-fill involves initializing the model's cache with the key-value pairs of a predetermined prompt. This preloading allows the model to access the necessary information immediately, facilitating faster and more efficient text generation.

Chunking is a method employed when dealing with exceptionally long prompts. The prompt is divided into smaller segments, which are processed individually. Each chunk is handled within the constraints of the window size, allowing the model to compute attention over both the pre-filled cache and each chunk sequentially. This segmentation ensures efficient memory usage and maintains the quality of the generated text by focusing computational resources on manageable portions of the input data (Figure 4).

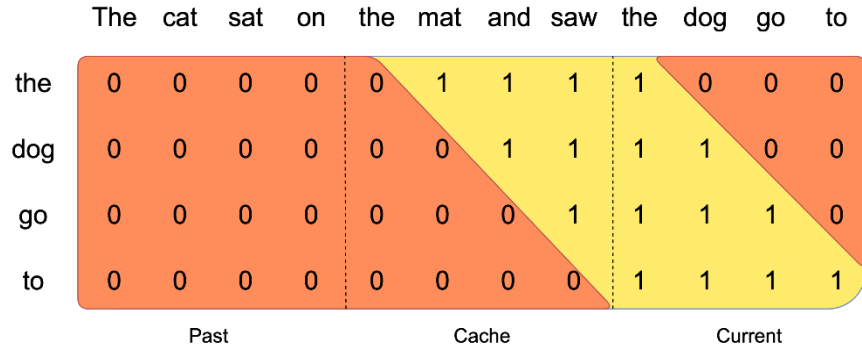


Figure 4: Pre-fill and Chunking

Through these architectural advancements, Mistral 7B effectively overcomes the recognized limitations of large language models, especially in managing long sequences efficiently and ensuring fast inference times.

3 TOOLS AND TECHNIQUES

3.1 LangChain

LangChain [20] is a framework designed for developing applications powered by large language models (LLMs). It acts as a generic interface, helping with the integration of nearly any LLM into various applications.

The framework comprises several components aimed at simplifying LLM application development: abstractions, LLM modules, prompt templates, chains, indexes, text splitters, memory utilities, and agents.

Abstractions simplify LLM application development by representing common steps and concepts required for working with language models. LLM modules offer a standardized interface for integrating various LLMs into LangChain applications. Prompt templates formalize prompt composition without manual coding. Chains combine LLMs with other components to create applications by executing a sequence of functions.

Indexes denote specific external data sources necessary for certain tasks. Document loaders import data from sources like file storage services, web content, and collaboration tools. Text splitters segment text into meaningful chunks for processing. Memory utilities facilitate the retention of conversation history or summarizations. Agents employ LLMs as reasoning engines to determine actions within applications.

LangChain supports various application use cases, including chatbots, summarization, question answering, data augmentation, and virtual agents integrated with robotic process automation (RPA).

3.2 Retrieval-Augmented Generation

The traditional approach to training foundation models involves offline training, which means that after training, any inference data is out-of-distribution for them. Moreover, these models are trained on generic domain corpora, which limits

their effectiveness in handling domain-specific tasks. However, Retrieval Augmented Generation (RAG) [21] offers a solution to this problem by allowing the retrieval of data from external sources and incorporating it into the prompts. By adding relevant retrieved data in context, RAG enhances the performance of foundation models.

3.2.1 Concept of RAG

RAG enables the utilization of external data from various sources, including document repositories, databases, and APIs, to augment prompts. The initial step involves converting the documents and user queries into a compatible format for conducting relevancy searches. This compatibility is achieved by transforming the document collection or knowledge library and user-submitted queries into numerical representations using embedding language models. RAG model architectures compare the embeddings of user queries with the vector representation of the knowledge library. Subsequently, the original user prompt is enriched by appending relevant context from similar documents within the knowledge library. This augmented prompt is then forwarded to the foundation model (Fig. 5). Knowledge libraries and their corresponding embeddings can be updated asynchronously.

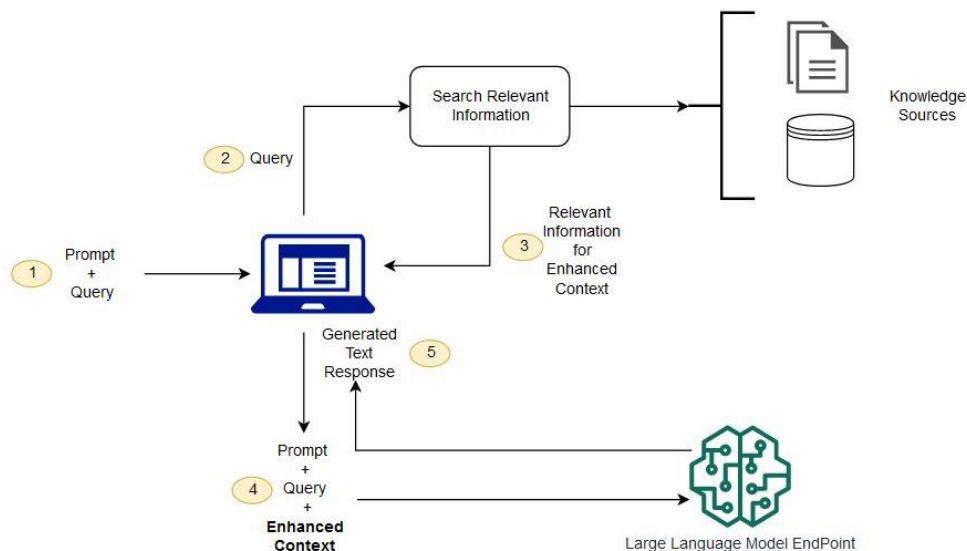


Figure 5: RAG architecture

3.2.2 Key Components of RAG

1. Embedding – numerical representation of text data.

Embeddings convert high-dimensional vectors (text, images) into a more compact form, capturing the semantic relationships between inputs. This process positions semantically similar items close to each other in the embedding space (Fig. 5). By using embeddings to represent queries and documents, RAG applications can use similarity search algorithms like cosine similarity to find the most relevant documents based on their semantic similarity. Embeddings can be generated using specific LMs, like BERT, GPT-3, Word2Vec.

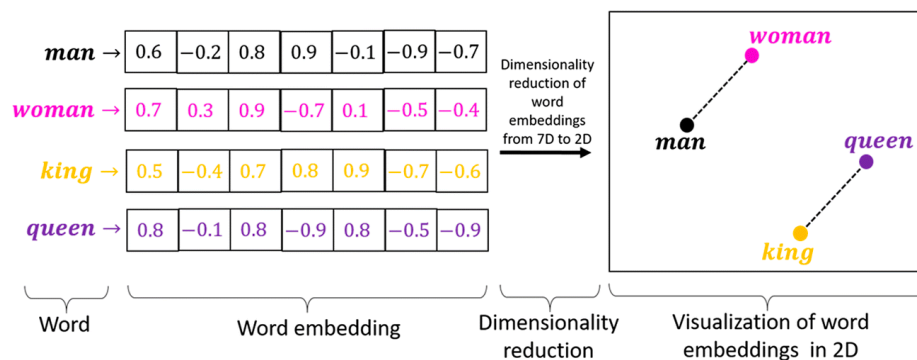


Figure 6: Embedding example

2. Vector store (database) – storage systems for efficiently retrieving embeddings.

The vector database efficiently indexes and stores vector embeddings for quick retrieval and similarity search. It includes features like CRUD operations, metadata filtering, horizontal scaling, and a serverless architecture.

3. Retriever – returns document based on an unstructured query.

Retrieval involves searching for relevant documents within a large collection. The mechanism scans the database to find segments similar to the user's query that is embedded previously). Then the necessary document used to improve the LLM's understanding and response generation.

In LangChain, there are two types of searches available for text retrieval:

"similarity" and "mmr". The "similarity" search uses the retriever object to find

text chunk vectors that closely match the question vector. In contrast, the "mmr" search prioritizes both similarity to the query and diversity among the selected documents using maximum marginal relevance search.

3.2.3 Advantages of RAG method:

Ensuring the use of current and accurate information: The RAG approach guarantees that the responses generated by the LLM are not solely based on outdated or irrelevant training data. Instead, it utilizes up-to-date external data sources to provide accurate and timely responses.

Minimizing the occurrence of incorrect or fabricated responses: RAG aims to reduce the risk of the LLM generating inaccurate or false information, also known as hallucinations. By grounding the model's output in relevant external knowledge, it can include citations of original sources, allowing for human verification.

Providing responses to specific domains: With RAG, the LLM can provide responses that are contextually relevant to an organization's proprietary or domain-specific data. This ensures that the generated responses are specific to the particular field or industry, enhancing their usefulness and applicability.

4 EXPERIMENTS AND RESULTS

4.1 Dataset

For building a chatbot tailored to the Ukrainian legislative system, we selected all legislative acts from the 8th and 9th convocations of the Verkhovna Rada, amounting to 6200 documents. The data was sourced from the official website of the Parliament of Ukraine [22], [23]. This dataset includes supplementary information accompanying the texts of the acts: act number, title, registration date, rubric, subject, and URL.

4.2 Resources

All testing and development were conducted in a virtual environment on Google Colab utilizing a T4 GPU. The resulting chatbot is hosted on the HuggingFace Spaces virtual environment.

4.3 Model Selection

For our chatbot, we chose the model *Mistral-7B-Instruct-Ukrainian* [24], which has been specially fine-tuned for generation in the Ukrainian language using both structured and unstructured datasets. This instruction model is based on Mistral-7B-v0.2 and adopts the architectural features of a transformer model. It was employed for question answering and generating responses using retrieved content.

For embedding generation, we utilized the *multilingual-e5-large model* [25]. This model is a state-of-the-art model designed specifically for embedding generation and supports 94 languages, including Ukrainian.

4.4 Experimental Flow

4.4.1 Embedding Creation

Initially, the loaded data was transformed into embeddings. To achieve this, we split the documents into 59,833 segments using the *RecursiveCharacterTextSplitter*,

with a split size of 1500 tokens and a split overlap of 150. These parameters were found to be the most optimal (Fig. 7).

```
Document(page_content="До 1 липня 2020 року засновники державних і комунальних закладів загальної середньої освіти або уповноважені ними органи зобов'язані припинити безстрокові трудові договори з керівниками таких закладів та одночасно укласти з ними (за їх згодою) трудові договори строком на шість років (з керівниками, які отримують пенсію за віком, - на один рік) без проведення конкурсу. У разі їх незгоди з продовженням трудових відносин на умовах строкового трудового договору - звільнити їх згідно з пунктом 9 частини першої статті 36 Кодексу законів про працю України. Після завершення строку трудового договору такі особи мають право обиратися на посаду керівника того самого закладу освіти на ще один строк відповідно до статті 39 цього Закону;\r\n2) набрання чинності цим Законом є підставою для припинення безстрокового трудового договору з педагогічними працівниками державних і комунальних закладів загальної середньої освіти, яким виплачується пенсія за віком, згідно з пунктом 9 частини першої статті 36 Кодексу законів про працю України.", metadata={'Unnamed: 0': 121, 'act_number': '463-20', 'bill_id': 66333, 'act_date': '2020-01-16', 'act_url': 'https://zakon.rada.gov.ua/go/463-IX', 'type': 'Проект Закону', 'title': 'про повну загальну середню освіту', 'registrationDate': '2019-08-29', 'registrationConvocation': 'IX скликання', 'rubric': 'Галузевий розвиток', 'subject': 'Кабінет Міністрів України'})
```

Figure 7: Split example

The resulting chunks were then converted into embeddings using the *multilingual-e5-large* model, with Chroma DB serving as the vector store. During our experiments, we determined that Chroma DB was the best fit for our purposes. The embedding loading was faster compared to FAISS DB, requiring only 1.6GB of storage, whereas FAISS DB needed more than 100GB. Chroma DB offers a lightweight design and a user-friendly interface, which streamline the storage and querying of vector embeddings.

4.4.2 Retrieval techniques

The most crucial part of the Retrieval-Augmented Generation (RAG) application is text retrieval, as the generated response heavily depends on the obtained splits. In our experiments, we focused on using similarity search and Maximal Marginal Relevance (MMR) methods with different top-k values.

For larger k values (greater than 15), the entire retrieved context did not fit into the context window of our LLM. Consequently, the outputted responses primarily focused on information retrieved at the beginning or the end of the returned context. The use of the MMR method proved to be less effective due to the specific nature of the retrieved content. Splits from different documents, which resulted in more diverse

context due to this method, led to responses that combined different legislative norms into one, often generating summaries with frequent hallucinations.

Through our experiments, we determined that the best and most relevant responses were obtained using the similarity search method with a top-k value of 7. This approach consistently returned the most accurate and contextually appropriate information.

4.4.3 Prompt Engineering:

During prompt engineering, various techniques were employed, including duplicating instructions at the beginning and after the specified context, as well as few-shot prompting. To enhance prompts, generative artificial intelligence models such as Chat-GPT and Gemini were utilized, although this did not yield improved results for our model.

In the prompts, it was essential to specify the role of the assistant, provide context, indicate the desired length of the generated response (3-5 sentences), and maintain a conversational writing style.

```
template = """ [INST] Ти асистент для надання відповідей з законодавства України.
На вхід ти отримав один закон, підсуму його простими словами,
викинь формальності та стандартні фрази.
Додай головні зміни, які згадуються і цьому документі і опиши їх трьома реченнями.
Якщо контекст пустий - пиши "Я не маю релевантної інформації.".
Контекст: {context}
Питання:
{question}
[/INST]
"""
```

Figure 8: Prompt example

The most effective technique proved to be duplication combined with few-shot prompting. An example is provided in Appendix A.

4.4.4 Resulting Chain:

For the final generation, a chain was composed utilizing prompts, models, and parsers for the output. Refer to Figure 9 for visualization.

```
rag_chain_from_docs = (
    RunnablePassthrough.assign(context=(lambda x: format_docs(x["context"])))
    | prompt
    | mistral_llm
    | StrOutputParser()
)
```

Figure 9: General chain for document retrieving

To generate a response, we passed the following hyperparameters to the model (Fig. 10).

```
text_generation_pipeline = pipeline(
    model=model,
    tokenizer=tokenizer,
    task="text-generation",
    temperature=0.01,
    repetition_penalty=1.2,
    return_full_text=True,
    max_new_tokens=750, do_sample=True,
    top_k=50, top_p=0.95,
    stopping_criteria=stopping_criteria
)
```

Figure 10: Pipeline parameters for answer generation

The temperature hyperparameter was set to 0.1 to minimize hallucinations and incorrect interpretations of legislative acts. Additionally, the repetition penalty was set to 1.2 to prevent the model from generating the same context repeatedly, which was a common behavior observed during the processing of legislative acts. This adjustment was made because formal language often contains many repetitions and redundancies, which can hinder the understanding of the main context for the LLM.

4.4.5 Different Approaches for specified tasks

To expedite the results, the base model was quantized. Different chains and prompts were developed for various chatbot functions. A top-k value of 15 was set

for summarizing a single legislative act. Queries to the LLM followed a standard format — "rag_chain_summarize.invoke(`"підсумуй контекст"`)" and were independent of user input.

For filtering relevant documents during search, a chain with dynamic filters in the retriever was developed, incorporating subject, registration date, and rubric metadata. When the filters are applied, similar documents for search will only be sought within documents containing the required metadata fields.

4.4.6 UI/UX

To enhance user experience in document retrieval, the system implemented the following features: metadata filtering (Fig. 11) and the ability to choose between a general inquiry or an act summarization request.

Додаткові фільтри

Хочу отримати підсумок одного документу

Ініціатор
Виберіть ініціатора законопроекту

Кабінет міністрів України

Тематика
Оберіть, яких галузей стосується законопроект

Державне будівництво Економічна політика Галузевий розвиток Соціальна політика
Безпека і оборона Багатосторонні міжнародні угоди

Початкова дата пошуку
Натисніть на значок календаря для вибору дати

30.10.2023

Figure 11: Filtering functionality

The chatbot development process was facilitated by the Gradio library, renowned for its user-friendly interface construction capabilities. Leveraging this tool, we ensured an intuitive and accessible interaction experience for users. Subsequently, deployment and hosting were carried out on the Hugging Face Spaces virtual platform.

4.4.7 Implementation challenges

The implementation of *SelfQueryRetriever* was hindered by the inability of the small model to construct prompts structured in JSON format in the Ukrainian language.

4.5 Results

The developed chatbot, named "Legislative Assistant," can generate responses to both general and specialized queries. This chatbot effectively handles tasks related to summarization, identifies key contexts, and extracts relevant changes in each legislative proposal, presenting them succinctly. While the model does not achieve perfect accuracy for user deployment, it significantly accelerates the preliminary review of documents, aids in understanding their context, and facilitates the selection of pertinent legislative documents in the Ukrainian legal framework. The model provides links to the documents used during generation, allowing users to access the full text of legislative acts (see Appendix B).

Within the model, the addressing off-topic queries feature is implemented using prompt instructions.

During development, various techniques for the retriever and prompt engineering were tested, and the most effective ones, according to human evaluation, were selected.

The model underwent real-time task validation. Analysts assessing the operations of the Verkhovna Rada of Ukraine recognized the model's efficacy. Its deployment notably improves the organization of legislative acts by topics and the creation of concise summaries.

4.6 Advantages of the work:

The work offers several key advantages:

- Ease of reproducibility, facilitated by features like data privacy, offline accessibility, and customization options.

- Effective minimization of hallucinations using RAG.
- Provision of relevant and accurate responses.
- Inclusion of references to the utilized documents.
- Up-to-date legislative acts ensure the timeliness and relevance of the information provided.

4.7 Limitations of the work:

The model, fine-tuned on a small number of parameters and limited datasets, exhibits a high frequency of inaccuracies in formulating lexically and grammatically correct sentences. For instance, it often refers to children as "*дитинчата*" in Ukrainian and constructs logically incorrect sentences. In generating responses related to Ukrainian legislation, inaccuracies arise from merging context from various legislative acts into a single coherent sentence. This can be mitigated by improving the retriever's solution. An important aspect is initial text preprocessing, which would filter out irrelevant and formal additions in each document (as headers or repetitive formal phrases). Another solution lies in creating a corpus with legislative drafts and expert descriptions. Pretraining or fine-tuning the model on such data can significantly improve the quality of generation.

5 CONCLUSIONS

The "Legislative Assistant" chatbot leveraged multiple effective techniques to ensure accurate and relevant responses in the Ukrainian legislative domain.

The integration of retrieval-augmented generation (RAG) techniques, along with the Mistral-7B-Instruct-Ukrainian model and optimized vector storage using Chroma DB, enabled the chatbot to provide up-to-date and grounded responses based on the corpus of legislative acts.

Through experimentation and validation by legal analysts, the chatbot demonstrated its effectiveness in summarizing acts, identifying key contexts, and extracting relevant changes.

The techniques employed in this research contribute to the broader field of conversational AI, showcasing the potential of combining large language models, retrieval-augmented generation, and innovative architectural approaches for developing domain-specific intelligent assistants in Ukrainian language.

REFERENCES

- [1] S. Minaee et al., “Large language models: a survey,” arXiv:2402.06196, arXiv, 2024.
- [2] C. E. Shannon, “Prediction and entropy of printed english,” *Bell system technical journal*, vol. 30, no. 1, pp. 50–64, 1951.
- [3] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.
- [4] Y. Bengio, R. Ducharme, and P. Vincent, “A neural probabilistic language model,” *Advances in neural information processing systems*, vol. 13, 2000.
- [5] H. Schwenk, D. Dechelotte, and J.-L. Gauvain, “Continuous space language models for statistical machine translation,” in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 2006, pp. 723–730.
- [6] T. Mikolov, M. Karafiat, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model.” in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.
- [7] A. Graves, “Generating sequences with recurrent neural networks,” arXiv preprint arXiv:1308.0850, 2013.
- [8] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He et al., “A comprehensive survey on pretrained foundation models: A history from bert to chatgpt,” arXiv preprint arXiv:2302.09419, 2023.
- [9] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang et al., “Pre-trained models: Past, present and future,” *AI Open*, vol. 2, pp. 225–250, 2021.
- [10] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, “Pre-trained models for natural language processing: A survey,” *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.

[11] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann et al., “Palm: Scaling language modeling with pathways,” arXiv preprint arXiv:2204.02311, 2022.

[12] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Roziere, N. Goyal, E. Hambro, F. Azhar ` et al., “Llama: Open and efficient foundation language models,” arXiv preprint arXiv:2302.13971, 2023.

[13] OpenAI, “GPT-4 Technical Report,” <https://arxiv.org/pdf/2303.08774v3.pdf>, 2023.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.

[16] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” arXiv preprint arXiv:2006.03654, 2020.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[18] A. Kazemnejad, “Transformer Architecture: The Positional Encoding,” kazemnejad.com, 2019. [Online]. Available: https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

[19] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier et al., “Mistral 7b,” arXiv preprint arXiv:2310.06825, 2023.

[20] "Introduction to LangChain," python.langchain.com, 2023. [Online]. Available: <https://python.langchain.com/v0.2/docs/introduction/>.

[21] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-

Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv:2005.11401, 2021.

[22] Verkhovna Rada of Ukraine. Available online:

https://data.rada.gov.ua/open/data/en/bills_acts-sk18 . Accessed: April 12, 2024.

[23] Verkhovna Rada of Ukraine. Available online:

https://data.rada.gov.ua/open/data/en/bills_acts-sk19 . Accessed: April 12, 2024.

[24] T. Boros, R. Chivereanu, S. Dumitrescu, and O. Purcaru, "Fine-tuning and Retrieval Augmented Generation for Question Answering using affordable Large Language Models," in *Proceedings of the Third Ukrainian Natural Language Processing Workshop, LREC-COLING*, European Language Resources Association, 2024.

[25] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei,

"Multilingual E5 Text Embeddings: A Technical Report," arXiv:2402.05672, 2024.

APPENDIX A

template ="" [INST] Ти асистент для надання відповідей з законодавства України. Використовуй лише вказаний нижче Context максимально точно. Описуй лише події простими словами без формальностей. Якщо питання не стосується законодавства України - пиши "Я не маю релевантної інформації. Спробуйте ще". Ось приклади використання:

Context:

'9) доручити Міністерству освіти і науки України:\r\nпередбачити у стандартах вищої освіти для першого (бакалаврського) рівня в межах кожної спеціальності наявність компетентності випускника щодо здатності використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя, а також відповідних результатів навчання, зокрема щодо виконання встановлених нормативів фізичної підготовленості;\r\nпрекоментувати ректорам вищих навчальних закладів з метою формування зазначеної компетентності та досягнення відповідних результатів навчання передбачити для студентів заняття з фізичного виховання не менш як протягом двох років навчання в обсязі 4 години на тиждень, враховуючи, що заняття з фізичного виховання мають відновлювальне та рекреаційне значення;\r\nв установленому порядку до 1 вересня 2016 року доповнити технологічні вимоги щодо забезпечення провадження освітньої діяльності у сфері вищої освіти (Ліцензійні умови провадження освітньої діяльності закладів освіти, затверджені постановою Кабінету Міністрів України від 30 грудня 2015 року № 1187) в частині забезпеченості здобувачів вищої освіти спортивними залами, стадіонами та спортивними майданчиками (у квадратних метрах на одного студента);'

Частину другу статті 55 доповнено абзацом дев'ятим згідно із Законом № 2657-VIII від 18.12.2018\r\nвимагати повного та неупередженого розслідування випадків булінгу (цькування) стосовно дитини або будь-якого іншого учасника освітнього процесу.\r\nЧастину другу статті 55 доповнено абзацом десятим згідно із Законом № 2657-VIII від 18.12.2018\r\n3. Батьки здобувачів освіти зобов'язані:\r\nвиховувати у дітей повагу до гідності, прав, свобод і законних інтересів людини, законів та етичних норм, відповідальне ставлення до власного здоров'я, здоров'я оточуючих і довкілля;\r\nсприяти виконанню дитиною освітньої програми та досягненню дитиною передбачених нею

результатів навчання;\r\nповажати гідність, права, свободи і законні інтереси дитини та інших учасників освітнього процесу;\r\nдбати про фізичне і психічне здоров'я дитини, сприяти розвитку її здібностей, формувати навички здорового способу життя;\r\nформувати у дитини культуру діалогу, культуру життя у взаєморозумінні, мирі та злагоді між усіма народами, етнічними, національними, релігійними групами, представниками різних політичних і релігійних поглядів та культурних традицій, різного соціального походження, сімейного та майнового стану;\r\nнастановленням і особистим прикладом утверджувати повагу до суспільної моралі та суспільних цінностей, зокрема правди, справедливості, патріотизму, гуманізму, толерантності, працелюбства;"

ANSWER: регуляції для вищих навчальних закладів передбачають обов'язкове фізичне виховання студентів та забезпечення їх доступом до спортивних залів і майданчиків. Також вони спрямовані на покращення боротьби з булінгом в освітніх установах та підвищення відповідальності батьків за виховання дітей.

Context: {context}

QUESTION:

{question}

Напиши три речення. Описуй отримані документи просто, доступно для підлітка. Вказуй лише найголовніші події. Якщо питання не стосується законів - пиши "Я не маю релевантної інформації. Спробуйте ще". На вхід ти отримав декілька законів, підсумуй їх простими словами, викинь формальності та стандартні фрази. опиши найголовніші зміни у трьох реченнях.

[/INST]

""

APPENDIX B

Додаткові фільтри

 Хочу отримати підсумок одного документу

Номер законодавчого акту

2168-VIII

Щодо законодавства про державні гарантії медичного обслуговування населення: держава гарантує повну оплату медичних послуг та лікарських засобів за програмою медичних гарантій для українців, іноземців, біженців, а також надає екстрену медичну допомогу іноземцям та особам без громадянства. Медичні послуги та лікарські засоби за програмою медичних гарантій можуть фінансуватись місцевими органами влади. Буде створена Рада громадського контролю для контролю за дотриманням законодавства. Не допускається вимога від пацієнтів винагороди за медичні послуги й лікарські засоби, а оплата здійснюється за рахунок державного бюджету. Крім того, держава фінансує програми розвитку комунальних закладів охорони здоров'я, боротьби зі спалахами хвороб, судово-медичну та судово-психіатричну експертизу, а також підтримку некомерційних підприємств охорони здоров'я.

Використані закони:

про державні фінансові гарантії надання медичних послуг та лікарських засобів:

<https://zakon.rada.gov.ua/go/2168-VIII>

Верховна Рада ратифікувала Договір про взаємну правову допомогу у цивільних справах між Україною й Йорданії, підписаний 2019 року. Він набуде чинності через 30 днів після того, як сторони завершать свої внутрішні процедури. Це сприятиме більш ефективному розгляду українцями справ у Йорданії та навпаки. Президент підписав цей документ 2 вересня 2020 року.

Використані закони:

про ратифікацію Договору між Україною та Йорданським Хашимітським Королівством про взаємну правову допомогу у цивільних справах: <https://zakon.rada.gov.ua/go/861-IX> Хочу отримати підсумок одного документу

Номер законодавчого акту

861-20

👤: які регулюються казино та азартні ігри

👤: Ці закони регулюють організацію та проведення азартних ігор у країні, зокрема казино, букмекерські ставки, гральні автомати, а також діяльність у сфері азартних ігор в Інтернеті. Вони передбачають державне регулювання, ліцензування, публічний доступ до деяких реєстрів, заборонені види діяльності, такі як лотереї, ігри з генераторами випадкових чисел, а також окремі галузі, як торгівля цінними паперами та банківська діяльність, які мають власні ліцензування. Також зазначено, що будь-який інший вид діяльності у сфері азартних ігор заборонено, крім тих, що передбачені законом. Відповідно до цих законів, організатори азартних ігор зобов'язані дотримуватися певних правил, таких як запобігання неправдивої інформації, невтручання у процес гри, забезпечення рівних умов для всіх гравців, боротьба з ігровою залежністю, а також дотримання принципів прозорості, стабільності, справедливості та об'єктивності. Також передбачено, що організатори азартних ігор у гральних залах казино повинні утримувати свідоцтва про встановлення готельної категорії та технічні патенти.

Використані закони:

про ліцензування певних видів господарської діяльності: <https://zakon.rada.gov.ua/go/222-VIII>

про державне регулювання діяльності щодо організації та проведення азартних ігор: <https://zakon.rada.gov.ua/go/768-IX>

Я не маю релевантної інформації. Спробуйте ще.

Ввести питання

Спитати

Додаткові фільтри

Хочу отримати підсумок одного документу

Номер законодавчого акту

1234

Який порядок проведення загальних зборів акціонерів?

Загальні збори акціонерів проводяться за два основних способи: очі, електронне голосування й опитування. Вони можуть бути річними або позачерговими. Власники 100% голосуючих акцій мають особливий режим проведення зборів. Звичайні збори мають обговорювати різноманітні питання діяльності компанії, а також приймати рішення щодо важливих змін, таких як зміна структури управління, зміни статуту, припинення бізнесу. Акціонери можуть брати участь дистанційно, обрати представника, а також заздалегідь визначити головного та секретаря зборів. Існують терміни для скликання, завершення та голосування. Крім того, закон передбачає можливість оскарження рішень

Використані закони:

про акціонерні товариства: <https://zakon.rada.gov.ua/go/2465-IX>