UDC 004.8, 519.816

EVALUATION OF QUANTIZED LARGE LANGUAGE MODELS IN THE TEXT SUMMARIZATION PROBLEM

Nedashkovskaya N. I. – Dr. Sc., Associate Professor at the Department of Mathematical Methods of System Analysis, Institute for Applied Systems Analysis at National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Yeremichuk R. I. – Bachelor of Systems Analysis, Kyiv, Ukraine.

ABSTRACT

Context. The problem of increasing the efficiency of deep artificial neural networks in terms of memory and energy consumption, and the multi-criteria evaluation of the quality of the results of large language models (LLM) taking into account the judgments of users in the task of summarizing texts, are considered. The object of the study is the process of automated text summarization based on LLMs.

Objective. The goal of the work is to find a compromise between the complexity of the LLM, its performance and operational efficiency in text summarization problem.

Method. An LLM evaluation algorithm based on multiple criteria is proposed, which allows choosing the most appropriate LLM model for text summarization, finding an acceptable compromise between the complexity of the LLM model, its performance and the quality of text summarization. A significant improvement in the accuracy of results based on neural networks in natural language processing tasks is often achieved by using models that are too deep and over-parameterized, which significantly limits the ability of the models to be used in real-time inference tasks, where high accuracy is required under conditions of limited resources. The proposed algorithm selects an acceptable LLM model based on multiple criteria, such as accuracy metrics BLEU, Rouge-1, 2, Rouge-L, BERT-scores, speed of text generalization, or other criteria defined by the user in a specific practical task of intellectual analysis. The algorithm includes analysis and improvement of consistency of user judgments, evaluation of LLM models in terms of each criterion.

Results. Software is developed for automatically extracting texts from online articles and summarizing these texts. Nineteen quantized and non-quantized LLM models of various sizes were evaluated, including LLaMa-3-8B-4bit, Gemma-2B-4bit, Gemma-1.1-7B-4bit, Qwen-1.5-4B-4bit, Stable LM-2-1.6B-4bit, Phi-2-4bit, Mistal-7B-4bit, GPT-3.5 Turbo and other LLMs in terms of BLEU, Rouge-1, Rouge-2, Rouge-L and BERT-scores on two different datasets: XSum and CNN/ Daily Mail 3.0.0.

Conclusions. The conducted experiments have confirmed the functionality of the proposed software, and allow to recommend it for practical use for solving the problems of text summarizing. Prospects for further research may include deeper analysis of metrics and criteria for evaluating quality of generated texts, experimental research of the proposed algorithm on a larger number of practical tasks of natural language processing.

KEYWORDS: limited resources, natural language processing, text summarization, large language models, quantization, multi-criteria analysis.

ABBREVIATIONS

NN is a neural network;
LLM is a large language model;
LLaMA is a large language model by Meta AI;
NLP is a natural language processing;
PLM is a pretrained transformer language model;
BERT is a bidirectional encoder representations by transformer;
BNN is a binary neural network;
STE is a straight-through estimator;
QAT is a quantization aware training;
PTQ is a post-training quantization;
ROUGE is a recall-oriented understudy for gisting evaluation – a set of metrics;
BLEU is a bilingual evaluation understudy algorithm;
PCM is a pairwise comparison matrix.

NOMENCLATURE

r is a real-valued input;
$[\alpha, \beta]$ is a cutoff range;
b is a quantization bit width;
r^q is a result of quantization of r;
S is a real-valued scaling coefficient;
Z is a integer zero point;
$f(r)$ is a transformation operation for quantization;
int is a rounding operation;
clip is a clipping function;
$\hat{r}$ is a result of dequantization;
Δ_i is a quantization threshold;
y_i is a quantization level;
$Q(\beta, b)$ is a set of quantization levels;
$\prod(\cdot)$ is an operation of projecting;
w is a weight;
w^b is a binarized weight;
$\sigma(w)$ is a "hard sigmoid" function;
a, b, c are weighting coefficients;
λ is a coefficient of regularization;
Q_a is a learnable quantization function;
L_{CE} is a traditional cross-entropy loss function;
L_{DL} is a distribution loss;
$H(\cdot,\cdot)$ is a loss function between the teacher model and the apprentice model;
w^{FP} is a full-precision weight of the teacher model;

DOI 10.15588/1607-3274-2025-2-12

p^T, p^S are predictions based on the teacher and student models.

INTRODUCTION

The importance of text summarization has increased with the information explosion in the digital age. Today, a huge amount of data is generated every second from various sources such as news, scientific reports, emails and social media posts. For both private individuals and businesses, it is almost impossible to consume available information without spending significant time. Text summarization tools offer a practical solution, quickly represent the essence of voluminous documents, and thus allow efficient information consumption.

Large language models (LLMs) have fundamentally changed the process of text summarization, providing opportunities that surpass traditional statistical methods [1, 2]. Trained on vast amounts of textual data from various sources, the LLMs develop a comprehensive understanding of linguistic nuances, idiomatic expressions and complex sentence structures. As a result, they can create summaries that not only capture the essential information from the texts, but also preserve the style and tone of original text [1, 3].

One of the most significant effects of LLMs is their ability to perform text summarization at extremely large scales and at extremely high speeds, quickly extracting key points from large volumes of text. This capability allows businesses, researchers, and policymakers to stay informed and make data-driven decisions without having to manually sift through extensive documents.

LLMs can work with various types and formats of texts: books, news articles, blog posts, technical reports and other [1]. These models support many languages and offer the possibility of summarization in different languages [4, 5].

LLMs are now very accessible to users, not even requiring authorization to use the latest updated version of the GPT 3.5 Turbo model [4]. In addition, LLMs offer a recommendation service and a personalized summarization experience: the user, for example, may indicate his/her interests or the most relevant text's aspects, and the models adjust their summarization strategies accordingly. Such personalization allows to create individual resumes which are more relevant and useful for individual users, increasing their interest and satisfaction.

However, summarization using LLMs also faces a number of problems [6]. The first and the most important of them is to ensure a high level of accuracy of the generated summaries and a contextual understanding of input text documents. While LLMs can create coherent resumes, the complexity of human language and the subtleties of textual nuance often present challenges. The problem is that the LLMs sometimes try to capture irony, sarcasm, and implicit meaning in text documents, which can lead to summaries that distort the original content. In addition, LLMs may omit important information or emphasize less important details, especially in texts of high information density or complex structure.

The second challenge is the bias of summaries generated by LLMs. The reason for this problem lies in the fact that the extremely large training data sets are usually created by humans and, as a result, are already characterized by a certain level of bias. Texts generated by LLMs can further reinforce these biases, leading to a distorted or partial representation of the original texts. This problem is particularly relevant in such sensitive areas as news distribution, legal document processing and educational content, where impartiality and fairness are of paramount importance.

At last, the use of LLMs requires powerful processors and large amounts of memory. To run, for example, a model of the GPT-4 level, which has at least 70 billion parameters, you need at least 48 GB of video memory [5].

In the end of 2023 and the beginning of 2024, the focus of the LLM community has shifted to the release of open-source and quantized models. In April 2024, Meta AI introduced LLaMA-3 70B and quantized LLaMA-3 8B models, which are improved versions of the LLaMA-2 [7]. The release of LLaMA-3 has once again raised the bar of quality for models of their size. For instance, the LLaMa-3 8B model aims to perform better than the LLaMa-3 70B in some tasks, while having 8.75 times fewer parameters.

The object of study is the process of automated text summarization using LLMs.

The subject of study is the analysis of quantization techniques and several different LLMs, which were proposed in 2023 and the beginning of 2024, depending on a set of multiple criteria.

The purpose of the work is to develop an algorithm of LLMs' evaluation in terms of multiple quantitative and qualitative criteria.

1 PROBLEM STATEMENT

Suppose $a_1, a_2, ..., a_n$ are alternative LLMs, such as LLaMa-3, Gemma, Qwen, Stable LM-2, Phi-2, Mistal, and their quantized versions, GPT-3.5 Turbo and other LLMs (decision alternatives), and $c_1, c_2, ..., c_m$ are the following decision criteria:

– metrics ROUGE, BLEU and BERT-score, which are used for evaluation of generated texts;

– speed of text summarization;

– convenience of using the LLM.

It is necessary:

– to evaluate decision alternatives (modern LLMs of various sizes) in a text summarization problem in terms of above decision criteria, using different data sets: CNN/ Daily Mail 3.0.0, and Extreme Summarization (XSum);

– to integrate modern LLMs of various sizes: LLaMa-3, Gemma, Qwen, Stable LM-2, Phi-2, Mistal, GPT-3.5 Turbo into a summarization service.

DOI 10.15588/1607-3274-2025-2-12

2 REVIEW OF THE LITERATURE

Several approaches are considered to improve the efficiency of NN models in terms of memory size, power consumption and others, while simultaneously providing an acceptable compromise between accuracy and generalization property of the models:

– designing efficient architectures for NN models; adaptation and co-design of NN architectures for a specific target hardware;

– quantization;

– pruning;

– model distillation.

The issue of quantization of NNs is partly related to works in the field of neuroscience [8–10], according to which the human brain stores information in discrete and quantized rather than continuous form.

One reason for the need for quantization is that the information, which is stored in continuous format, is exposed to noise (external, thermal, synaptic and other), and such noise is always present in small quantities in the physical environment, including the human brain [11]. Signals in discrete form may be more robust to such low-level noise. In addition, discrete representations have a higher generalization ability [12] and higher efficiency in resource-constrained applications [13].

Model distillation consists of first training a large model, and then using it as a teacher to train a more compact model [14 – 16]. The main challenge is to obtain accurate results with a high degree of data compression as a result of distillation. Strong compression for knowledge distillation methods usually leads to a significant decrease in the accuracy of the results. Accuracy can be improved by combining knowledge distillation with quantization and pruning techniques [16].

In recent years, there has been a trend to use pre-trained language representations in natural language processing systems, which are applied more flexibly and independently of the task. Single-layer representations based on word-to-vector models were first explored and transferred to task-specific architectures [17, 18]. After that, the recurrent neural networks with contextual state, multiple representation layers [19–22] and sequence-to-sequence model with copy mechanism [23] were used to form stronger representations.

Recently, pretrained transformer language models (PLMs) have developed [24], which are directly fine-tuned, completely eliminating the need for task-specific architectures [25–27].

The Bidirectional Encoder Representations by Transformer (BERT) model [25] marked a significant advance in NLP tasks. BERT is extended to the sequence generation task in [26], where a two-stage decoding process is designed for efficient usage of BERT's context modeling ability. Firstly, the summary is generated using a left context-only-decoder. After that, each word of the summary is masked, the refined word is predicted, and the reinforcement objective is cooperated with the refined decoder for further improvement of the naturalness of the generated sequence [26]. A self-supervised pre-training objective for abstractive summarization, a gap-sentences generation and study strategies for selecting those sentences are proposed in PEGASUS model [27]. The PEGASUS is able to be adapted very quickly, and fine-tune with small numbers of supervised pairs.

The T5 (Text-to-Text Transfer Transformer) model had 11 billion parameters and used a transfer learning approach, outperforming its predecessors BERT and GPT2 in a wide range of NLP applications, including text classification, question answering, and especially text summarization [28]. The GPT-3 model proposed in 2020 already had 175 billion parameters and could be zero-short transferred to downstream tasks without fine-tuning [1]. In 2020, a new learning method known as replaced token detection was introduced in the ELECTRA model. In a pre-training task, this model learns to distinguish real input tokens from plausible but synthetically generated replacements [29]. ELECTRA is effective in extractive note-taking, where identifying and summarizing the most important sentences in a text is critical.

The DeBERTa (Decoding-enhanced BERT with Disentangled Attention) model, released in 2020 as an improvement to BERT, separated the representation of words from their positions in the text, which allowed the model to more clearly understand contextual relationships, create contextually deeper and more coherent summarizations, generalize different types of texts [30].

GPT-3.5, also known as ChatGPT, introduced in 2022 provides enhanced interactivity, allowing users to interact with the model directly by refining the summarization results [4]. Dynamically responding to user input and adjusting its responses, GPT-3.5 became not just a tool for passive data processing, but also an active participant in information analysis and decision-making processes.

Further advances in this area have been made by fine-tuning LLMs for a set of tasks formulated as instructions, allowing the models to better respond to instructions and reducing the need for labeled data. It is emphasized in [31] that fine-tuning on instructions can improve performance across a range of models, prompting setups, and evaluation tasks. As a result, Flan-T5 model with 11 billion parameters [31] achieves strong few-shot performance compared to much larger models, such as PaLM 62B.

An open-source LLaMA (Large Language Model Meta AI) model was proposed by Meta AI in February 2023 and marked a significant evolution in NLP for deploying advanced NLP tools in resource-constrained environments by optimizing performance in various computing environments [32]. The LLaMA model has an ability to produce high output quality with less training data, which optimizes use of this model for real-time text summarization problems, where fast and accurate compression of information is very important. Also, the release of the LLaMA model greatly improved the position of open-source models, since there were and still are a lot of big players hiding detailed information about the architecture, number of parameters, training

DOI 10.15588/1607-3274-2025-2-12

configurations, data sets, etc. This has prompted many of the big players in AI to release even more open-source models of the GPT-level from OpenAI.

In the Gemini model proposed by Google in May 2023, two-context processing of texts and integration of information from different sources were introduced [33]. In this regard, the Gemini model is effective for dynamic content such as news streams.

The Mistral open-source model [34], developed by an independent research group from France and released in July 2023, aimed to solve the problem of generating resume texts of better quality in many different languages that were previously unavailable. The Mistral model became more powerful than the similar LLaMa model by 7 billion parameters, actually taking first place among open-source models at that time according to expert evaluations [34]. Anyone can download the Mistral model weights for free and run it locally having the appropriate computing resources, unlike GPT-4 and GPT-3.5 models, which are only available as an application programming interface service.

In 2023 and 2024, there is also a trend to reduce the dimensions of LLMs, so that they can effectively work on devices with limited computing resources. Examples of such models are Qwen1.5 – 0.5B, 1.8B, 4B, Stable LM – 1.6B, Phi-2 – 2.7B, Phi-3 – 3.8B and TinyLlama - 1.1B. Recently, 8-bit and 4-bit quantization opens up an opportunity of running LLMs on consumer hardware [35 – 37].

Non-uniform quantization, binarized weights and activations, extreme and mixed precision quantization, quantization aware training (QAT) and post-training quantization (PTQ) are used in modern LLMs [37].

In order to choose the best models for text summarization based on user preferences and multiple quality criteria, and to increase the speed and quality of text summarization based on LLMs, it is necessary to evaluate modern quantized LLMs of different sizes in a text summarization problem using several data sets and metrics BLEU, Rouge-n, Rouge-L and BERT-score.

3 MATERIALS AND METHODS

Uniform quantization. The basic quantization operation performs uniform quantization using the following steps [38, 39]:

1. Specify the range of a real-valued quantity to be quantized, and to clip values outside this range.

2. Map real values to integer values which are represented by the required bit-width of the quantized representation. This is often performed by rounding each real value to the nearest integer.

Let r be a real-valued input, [α, β] be the range of r chosen for quantization (the cutoff range), and b is the quantization bit-width. Uniform quantization represents the full-precision input value $r \in [\alpha, \beta]$ as the low-precision integer within the range $[-2^{b-1}, 2^{b-1}-1]$. Inputs outside the range are cut to the nearest boundary.

Asymmetric uniform or affine quantization represents a real value $r \in \mathbb{R}$ as a signed b-bit integer $r^q \in \{-2^{b-1}, -2^{b-1}+1, ..., 2^{b-1}-1\}$. The following transformation operation is defined by

$$f(r) = S \cdot r + Z,$$

$$S = \frac{2^b - 1}{\beta - \alpha},\ Z = -\text{int}(\alpha \cdot S) - 2^{b-1},$$

where S is a real-valued scaling coefficient, Z is the zero point – the integer, to which the real-valued zero is mapped, and int is a rounding operation, which displays a real value to an integer. The scaling coefficient S divides the range of the real-valued r into several partitions. In the 8-bit case, $S = \frac{255}{\beta - \alpha}$ and $Z = -\text{int}(\alpha \cdot S) - 128$.

The uniform quantization operation, also called asymmetric, is defined as follows [39]:

$$r^q = \text{quantize}(r, b, S, Z) =$$
$$= \text{clip}(\text{int}(S \cdot r + Z), -2^{b-1}, 2^{b-1} - 1),$$

where $\text{clip}(r,l,u) = l$ if $r < l$, $\text{clip}(r,l,u) = u$ if $r > u$ and $\text{clip}(r,l,u) = r$ if $l \le r \le u$.

The corresponding dequantization operation, which computes an approximation of the original real-valued input $\hat{r} \approx r$, is defined by

$$\hat{r} = \text{dequantize}(r^q, S, Z) = \frac{1}{S}(r^q - Z).$$

In the symmetric uniform quantization, the cutoff range and integer range are symmetric around zero, that is $\alpha = -\beta$, and the zero-point $Z=0$. For example, the integer range [−127, 127] is used for 8-bit quantization, and we do not use the −128 value in favor of symmetry. For int8, the loss of one out of 256 representable values is minor. However, for lower bit quantization we have to re-evaluate the trade-off between representable values and symmetry of quantization.

Symmetric uniform quantization represents a real value $r \in \mathbb{R}$ as a signed b-bit integer $r^q \in \{-2^{b-1}+1, -2^{b-1}+2, ..., 2^{b-1}-1\}$. The following transformation operation is defined by

$$f(r) = S \cdot r,$$
$$S = \frac{2^{b-1} - 1}{\beta},$$

and the result of quantization is as follows:

$$r^q = \text{quantize}(r, b, S) =$$
$$= \text{clip}(\text{int}(S \cdot r), -2^{b-1} + 1, 2^{b-1} - 1).$$

Uniform quantization operations are shown on Fig. 1.

DOI 10.15588/1607-3274-2025-2-12

Calibration is the process of selecting the cutoff range [α, β] [38, 39]. A popular method is to set $\alpha = r_{\min}$ and $\beta = r_{\max}$ for asymmetric uniform quantization. In this case, S is specified as $S = \frac{\max(|r|)}{2^{b-1}-1}$. In a case of symmetric quantization method, the maximum of absolute values is used: $\beta = \max(|r_{\min}|, |r_{\max}|)$. Then S is given as $S = \frac{2\max(|r|)}{2^b - 1}$.

A percentile of the distribution of absolute values observed during calibration also can be set [40]. For example, the 99% percentile would cut off 1% of the largest values. The Kullback-Leibler divergence can be used for calibration, which minimizes the loss of information between the quantized values and the original floating-point values.

Asymmetric uniform quantization is often applied in practice, as it results in a wider and therefore more accurate range, however, leading to more computationally expensive inference compared to symmetric quantization.

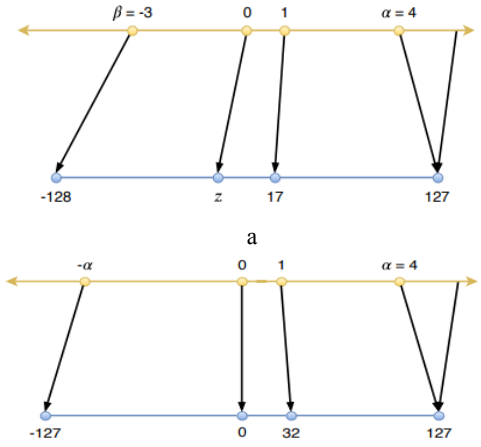

Figure 1 – Quantization of real values to int8: a – asymmetric, b – symmetric [39]

Non-uniform quantization methods provide higher accuracy for a fixed bit width, and these methods allow more attention to be focused on important regions of weight or activation values, such as in the case of bell-shaped distributions with long tails, and also support dynamic definition of cutoff ranges [41–43].

The non-uniform quantization operation is specified as

$$f(r) = y_i \text{ if } r \in [\Delta_i, \Delta_{i+1}),$$

where Δ_i are quantization thresholds, and y_i are quantization levels.

The quantization of a real-valued input r can be presented as

$$r^q = \prod_{Q(\beta,b)} \text{clip}(r, \beta),$$

where $Q(\beta,b)$ is a set of quantization levels, β is the cutoff threshold, the cutting function clip(·,β) clips values of r into range [– β, β], and b is the bit-width. Operation $\prod(\cdot)$ projects clipped r value onto the quantization level.

In a case of uniform quantization, the quantization levels are defined as

$$Q(\beta,b) = \beta \times \left\{0, \frac{\pm 1}{2^{b-1}-1}, \frac{\pm 2}{2^{b-1}-1}, \frac{\pm 3}{2^{b-1}-1}, ..., \pm 1\right\}.$$

For non-uniform "powers-of-two" quantization method, the quantization levels are constrained to be powers-of-two values or zero (Fig. 2) [42]:

$$Q^{PoT}(\beta,b) = \beta \times \left\{0, \pm 2^{-2^{b-1}+1}, \pm 2^{-2^{b-1}+2}, ..., \pm 2^{-1}, \pm 1\right\}.$$

Multiplication between a number 2^x, that is a power of two, and other number q can be implemented by bitwise shifting as follows:

$$2^x \cdot u = \begin{cases} u, & \text{if } x = 0 \\ u << x, & \text{if } x > 0 \\ u >> x, & \text{if } x < 0 \end{cases},$$

where ≫ is the right shift operation, which accelerates the computation and takes only one clock cycle in modern CPU architectures [42].

Quantization layers can also be trained along with model parameters using gradient descent methods [44].

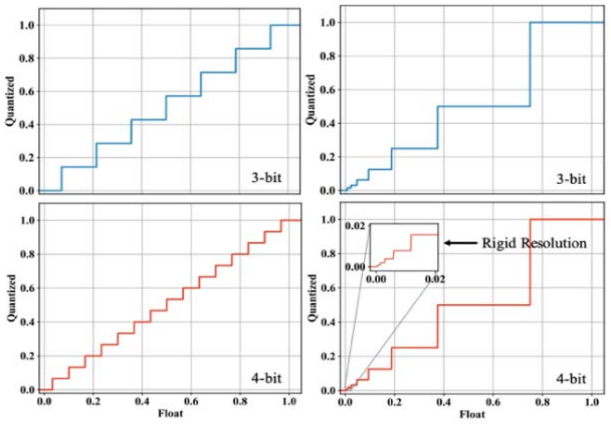

Figure 2 – Quantization of unsigned data to 3-bit or 4-bit: a – uniform, b – Powers-of-Two (PoT) quantization [42]

In binary networks (BNNs) – networks with binary weights and activations – most arithmetic operations are replaced with bit-wise operations, which potentially lead to a substantial increase in power-efficiency. BNNs and a method for their training are proposed in [45]. Experiments in [45] show that binarization cardinally reduces memory consumption, namely number of accesses and memory size during the forward pass at run-time and train-time.

DOI 10.15588/1607-3274-2025-2-12

When training a BNN, the weights and the activations are both constrained to +1 or −1. The binarization function can be either deterministic:

$$w^b = \text{sign}(w) = \begin{cases} +1, & \text{if } w \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

or stochastic:

$$w^b = \text{sign}(w) = \begin{cases} +1, & \text{with probability } p = \sigma(w) \\ -1, & \text{with probability } 1-p \end{cases},$$

where σ is the "hard sigmoid" function:

$$\sigma(x) = \text{clip}(\tfrac{w+1}{2}, 0, 1) = \max\left(0, \min\left(1, \tfrac{w+1}{2}\right)\right).$$

A significant benefit of joint binarization of weights and activations in BNNs is that the floating-point matrix multiplication is replaced by lightweight operations

$$\text{XnorDotProduct}\left(a_{k-1}^b W_k^b\right), \quad k = 1, \ldots, L,$$

followed by bit counting.

This operation is based on a following trick: it is relatively easy to handle continuous-valued inputs as fixed-point numbers, with m bits of precision [45]. For example, in the common case of 8-bit fixed point inputs:

$$sum = x \cdot w^b, \; sum = \sum_{n=1}^{8} 2^{n-1}(x^n \cdot w^b),$$

where x is a vector of 1024 8-bit inputs, w^b is a vector of 1024 1-bit weights, and *sum* is the resulting weighted sum.

Binarization, which limits quantized values to a 1-bit representation is considered as the most extreme quantization method. Binary operations can be computed efficiently using bitwise arithmetic and achieve significant speedup compared to higher precisions such as FP32 and INT8. Peak binary arithmetic performance on NVIDIA V100 GPUs is 8 times faster than INT8 [46].

Also, binarization can radically reduce memory requirements by 32 times. For many complex problems, however, simple binarization methods usually result in a serious decrease in accuracy.

Several methods were proposed to reduce decrease in accuracy in extreme quantization [47]:

1. Minimize the quantization error. The floating-point parameters are approximated by introducing a scaling factor $a \in \mathbb{R}$ for the binary parameter. Then, the quantization of weight w is formulated as $w \approx a \cdot w^b$, where w^b is the binarized weight. Optimal scaling factor and binary weights are found minimizing the quantization error

$$\min_{a, w^b} \| w - a \cdot w^b \|^2.$$

Thus, $w^b \in \{-a, a\}$ and lead to less quantization error than directly using values $\{-1, 1\}$. This method increases the inference accuracy of the network, and still have the benefits of fast computation.

A two-step quantization method is proposed to overcome shortcomings of the previous method [48]:

– All activations are low-bit quantized using a learnable quantization function Q_a. All weights are considered as full-precision values.

– Q_a is fixed, and scaling factor $a \in \mathbb{R}$ and the low-bit quantized weight vector w^b are learned as follows:

$$\min_{a, w^b} \| z - Q_a(a(x \odot w^b)) \|_2^2,$$

where the minimization problem can be solved iteratively.

2. Improve the network loss function. Additional quantization-aware loss item is proved to be practical and is introduced as regularizer [49]:

$$L = L_{CE} + \lambda \cdot L_{DL},$$

where L_{CE} is the traditional cross-entropy loss function, L_{DL} is the distribution loss to learn the binarization property, and λ is the coefficient of regularization.

Another approach uses the distillation technique, training a low-precision student network using a full-precision, well-trained, and large-scale teacher network. The loss function in this approach is as follows [15]

$$L(x; w^{FP}, w^b) = a \cdot H(y, p^T) + b \cdot H(y, p^S) + c \cdot H(p^T, p^S),$$

where $H(\cdot,\cdot)$ is the loss function between the teacher model and the apprentice model; w^{FP} is the full-precision weights of the teacher model, w^b is the binary weights of the apprentice (student) model; p^T, p^S are predictions based on the teacher and student models; y is the label for sample x; a, b, c are weighting coefficients.

3. Improved Training Method. The training method of BNNs, proposed in [45], uses the shift-based AdaMax algorithm and is a variant of the dropout method, but instead of randomly setting half of the activations to zero while computing the gradients, the binarization of activation and weight values is performed. A version of the straight-through estimator (STE) is applied with additional saturation effect to propagate gradients through a non-differentiable signed function while using the standard back-propagation algorithm.

When using a pre-trained model, quantization can lead to distortion of parameters of the trained model and, as a result, to convergence to a non-optimal value of the loss function. To deal with this problem, a NN model may be retrained using the quantized parameters to minimize the decrease in accuracy after quantization. The method is called QAT and consists of the following steps (Fig.3, a) [47, 50]:

DOI 10.15588/1607-3274-2025-2-12

1. Pretraining the base NN model without taking into account quantization. Model accuracy assessment.

2. Apply quantization to all layers of the pre-trained model. The resulting model supports quantization, but is not quantized. For example, the weights are float32 instead of int8. Only individual layers of the pre-trained model can be quantized to increase the accuracy of the model.

3. Fine-tuning (retraining) the model obtained in the previous step, which is quantization aware, on a subset of training data. Assessing the accuracy of the model and comparing it with the accuracy of the base model.

4. Building an actually quantized model with int8 weights and uint8 activations. Evaluating the accuracy of this model and comparing it with the accuracy of the base model.

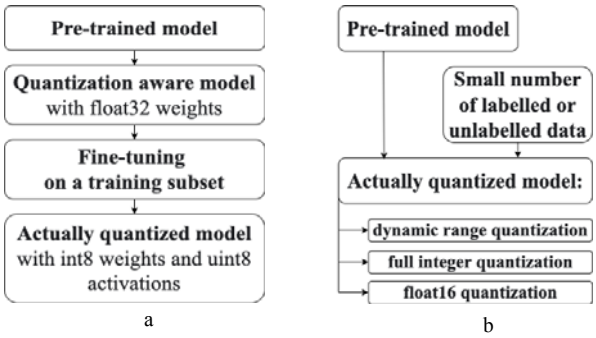

Figure 3 – QAT (a) and PTQ (b) methods of quantization

The process of fine-tuning the quantization aware model in above step 3 is as follows:

– the usual forward and backward pass, as well as the gradient step for updating the weight, are performed with floating point,

– model parameters are quantized after gradient update,

– the non-differentiable quantization operator is approximated by the identity function called the STE [51]. Later, instead of the rounding operation, a *W*-shaped non-smooth regularization function was proposed [52].

The QAT method helps to minimize the decrease of accuracy after quantization, despite the use of a rough approximation STE. The main limitation of QAT is the computational cost of retraining the NN. For example, low-bit precision quantization models may require several hundred epochs of the retraining. Also, the QAT method requires sufficient training data to retrain.

Post-training quantization (PTQ) performs quantization of weights and activations of a pre-trained model without additional fine-tuning (Fig.3b) [47, 50, 53]. Thus, PTQ is very fast method for quantizing NN models.

The evaluation of modern LLMs aims to answer the question whether model's size, architecture, quantization, and features of architecture significantly affect the summarization efficiency. Efficiency of texts generated in a process of summarization is estimated in terms of metrics ROUGE [54], BLEU [55] and BERT-score [56]. Additionally, speed of text summarization and convenience of using the LLM are analyzed. Qualitative decision criterion "convenience of using" requires, in its turn, expert or user evaluation. Weights of decision criteria are also calculated based on expert (user) preferences or judgements about relative importance of the criteria in a given problem.

The proposed algorithm for multiple-criteria evaluation of LLMs has several stages (Fig. 4).

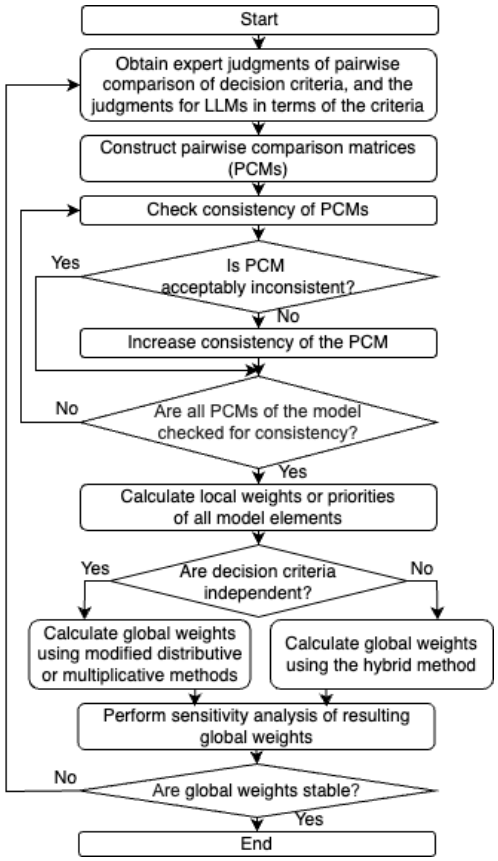

Figure 4 – An algorithm for LLMs multiple-criteria evaluation

At the first stage, expert compare importance of decision criteria using special scale, and his/her judgements are presented as pairwise comparison matrices (PCMs) [57]. Expert also estimates LLMs in terms of the criteria based on previously calculated metrics ROUGE, BLEU and BERT-score. Consistency of expert judgments is analyzed, using method of assessment and increasing of consistency [57]. This method is based on the property of weak inconsistency of PCM, finds undesirable cycles in a PCM and the most inconsistent elements of PCM. The method can be applied to various types of PCMs, such as multiplicative, additive, fuzzy and other [58]. As a result, we obtain PCMs of acceptable quality (inconsistency).

At next stage, the fuzzy preference method [59] is used for calculating local weights or priorities of model elements (LLMs and decision criteria). After that, local weights of model elements are aggregated using modified distributive, multiplicative or proposed hybrid

DOI 10.15588/1607-3274-2025-2-12

aggregation methods depending on the mutual dependence of the criteria. At the last stage, a sensitivity analysis of results is performed, and stability of results is assessed.

4 EXPERIMENTS

Experiments to evaluate the quality of text summarization by various LLMs were conducted on two different data sets: CNN/Daily Mail 3.0.0 [60] and Extreme Summarization (XSum) [61].

The CNN/Daily Mail 3.0.0 dataset includes over 300,000 unique English language news articles written by CNN and Daily Mail journalists. Initially, the set was developed for tasks of machine reading and understanding of texts, and subsequently for tasks of extractive and abstract summarization. Each entry in this set is represented by the following three key fields: the "id" field contains the SHA1 hash of the URL in hexadecimal format from which the text was retrieved; the "article" field is the text of the news article itself; and "highlights" of the article written by the author.

The XSum dataset was designed specifically to address complex summarization problems. The set entries are represented by the following fields: "id", "document", which is the text of the news article itself, "summary", which contains a one-sentence summary of the article.

Using two different data sets helps to increase the validity of obtained results for evaluation of the quality of text summarization by various LLMs.

Experiments were conducted using a temperature value of 0.1 and a maximum token length of 100 for each LLM, as proposed in [3]. The summation of 25 test samples of each data set was carried out.

5 RESULTS

Various LLMs in both standard and quantized form were tested on the ROUGE, BERT-score and BLEU metrics, in order to assess the impact of quantization on the performance of the models (Tables 1 and 2).

LLMs of different configurations and sizes were compared with each other, and it was analyzed how the number of model parameters affects the quality of summarization and processing speed.

Values of performance metrics for different LLMs depending on the size and quantization level of the models are shown in Tables 1 and 2 for the CNN/Daily Mail 3.0.0 and the XSUM datasets, respectively.

An example of expert pairwise comparison judgements of decision criteria made in fundamental scale and the corresponding PCM are shown in Figs. 5 and 6. These judgements (and PCM) have no cycles, are acceptably inconsistent and can be used for calculation of reliable local weights, shown on the right parts of Figs. 5 and 6.

An example of unacceptable PCM is shown in Fig. 7. In this case, the system finds the most inconsistent element of PCM and offers a new value for it, which ensures an increase of consistency level of the entire PCM (Fig. 7).

Table 1 – Metric values for different LLMs on the CNN/Daily Mail 3.0.0 dataset

LL model	Number of parameters (in billions)	Rouge-1	Rouge-2	ROUGE-L	BLEU	BERT-precision	BERT-recall	BERT-F1
LLaMa-3-8B-4bit	8	0.288	0.094	0.261	0.044719	0.858	**0.881**	**0.869**
Gemma-2B	2	0.263	0.08	0.245	0.039777	0.858	0.871	0.864
Gemma-2B-4bit	2	0.269	0.078	0.247	0.036853	**0.861**	0.873	0.867
Gemma-7B-4bit	7	0.271	0.082	0.245	0.036121	0.857	0.875	0.866
Gemma-1.1-2B	2	0.256	0.069	0.223	0.032376	0.858	0.874	0.866
Gemma-1.1-2B-4bit	2	0.251	0.067	0.227	0.031926	0.856	0.874	0.865
Gemma-1.1-7B-4bit	7	0.259	0.082	0.238	0.035257	0.858	0.876	0.867
Qwen-1.5-0.5B	0.5	0.286	0.097	0.248	0.045120	0.84	0.867	0.853
Qwen-1.5-0.5B-4bit	0.5	0.268	0.08	0.237	0.039331	0.844	0.867	0.855
Qwen-1.5-1.8B	1.8	0.283	0.083	0.253	0.040930	0.852	0.873	0.862
Qwen-1.5-4B	4	**0.295**	0.109	0.266	0.057609	0.85	0.876	0.863
Qwen-1.5-4B-4bit	4	0.294	**0.111**	**0.267**	**0.066049**	0.848	0.874	0.861
Qwen-1.5-7B-4bit	7	0.284	0.079	0.245	0.039231	0.855	0.88	0.868
Stable LM-2-1.6B	1.6	0.27	0.072	0.241	0.034851	0.853	0.872	0.862
Stable LM-2-1.6B-4bit	1.6	0.271	0.086	0.241	0.044366	0.852	0.877	0.864
Phi-2	2.7	0.283	0.097	0.259	0.051176	0.857	0.878	0.867
Phi-2-4bit	2.7	0.277	0.09	0.253	0.045895	0.858	0.876	0.867
Mistal-7B-4bit	7	0.26	0.071	0.229	0.028532	0.853	0.873	0.863
GPT-3.5 Turbo	175	0.275	0.078	0.25	0.035723	0.858	0.878	0.868

DOI 10.15588/1607-3274-2025-2-12

Table 2 – Metric values for different LLMs on the XSum dataset

LL model	Number of parameters (in billions)	Rouge-1	Rouge-2	ROUGE-L	BLEU	BERT-precision	BERT-recall	BERT-F1
LLaMa-3-8B-4bit	8	0.179	0.031	0.137	0.012043	0.842	0.886	0.864
Gemma-2B	2	0.175	0.031	0.141	0.007783	0.842	0.881	0.861
Gemma-2B-4bit	2	0.182	0.032	0.149	0.009392	0.843	0.882	0.862
Gemma-7B-4bit	7	0.173	0.03	0.149	0	0.844	0.884	0.863
Gemma-1.1-2B	2	0.167	0.023	0.139	0	0.844	0.881	0.862
Gemma-1.1-2B-4bit	2	0.167	0.025	0.134	0	0.845	0.882	0.863
Gemma-1.1-7B-4bit	7	0.18	0.035	0.153	0.013915	**0.846**	0.887	**0.866**
Qwen-1.5-0.5B	0.5	0.146	0.019	0.116	0.007328	0.819	0.864	0.841
Qwen-1.5-0.5B-4bit	0.5	0.151	0.021	0.115	0.007367	0.825	0.868	0.846
Qwen-1.5-1.8B	1.8	0.181	0.029	0.142	0.007655	0.839	0.881	0.859
Qwen-1.5-4B	4	0.178	0.024	0.146	0.006867	0.837	0.878	0.857
Qwen-1.5-4B-4bit	4	0.175	0.031	0.144	0.011973	0.83	0.873	0.851
Qwen-1.5-7B-4bit	7	0.185	0.035	0.155	**0.014836**	0.843	**0.891**	**0.866**
Stable LM-2-1.6B	1.6	0.174	0.028	0.149	0.008054	0.839	0.878	0.858
Stable LM-2-1.6B-4bit	1.6	0.165	0.026	0.136	0.006938	0.837	0.88	0.858
Phi-2	2.7	0.185	**0.036**	**0.165**	0.011087	0.841	0.883	0.861
Phi-2-4bit	2.7	**0.186**	0.034	0.159	0.010903	0.843	0.882	0.862
Mistal-7B-4bit	7	0.185	0.03	0.159	0.012617	0.843	0.888	0.865
GPT-3.5 Turbo	175	0.179	**0.036**	0.15	0.011358	0.845	0.888	**0.866**

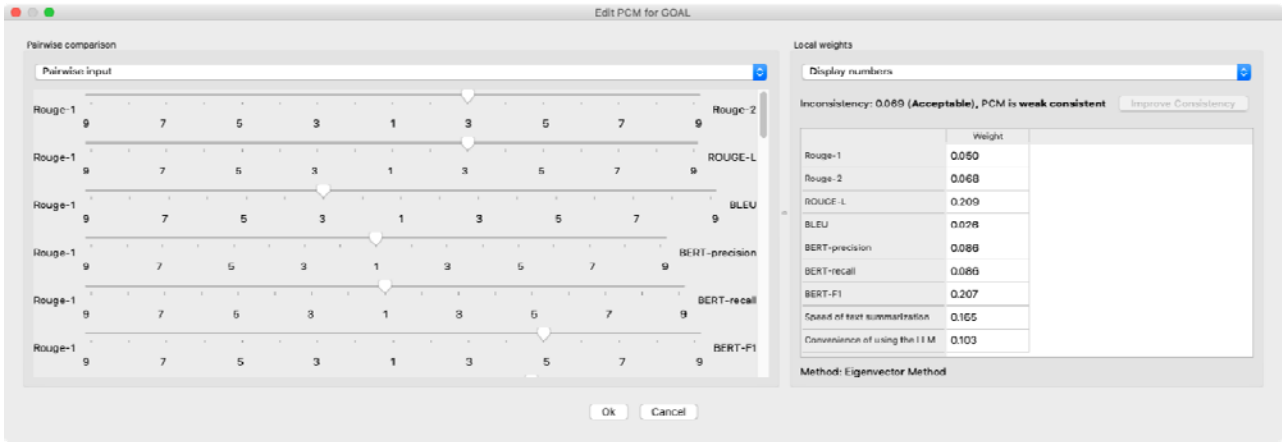

Figure 5 – An example of expert pairwise comparison judgements of decision criteria made in fundamental scale

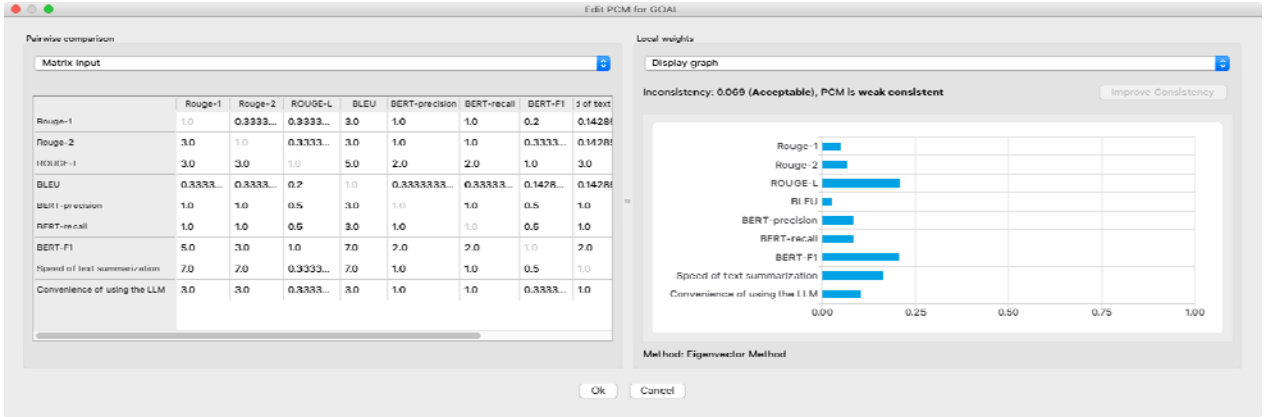

Figure 6 – An example of PCM and calculated weights of decision criteria

DOI 10.15588/1607-3274-2025-2-12

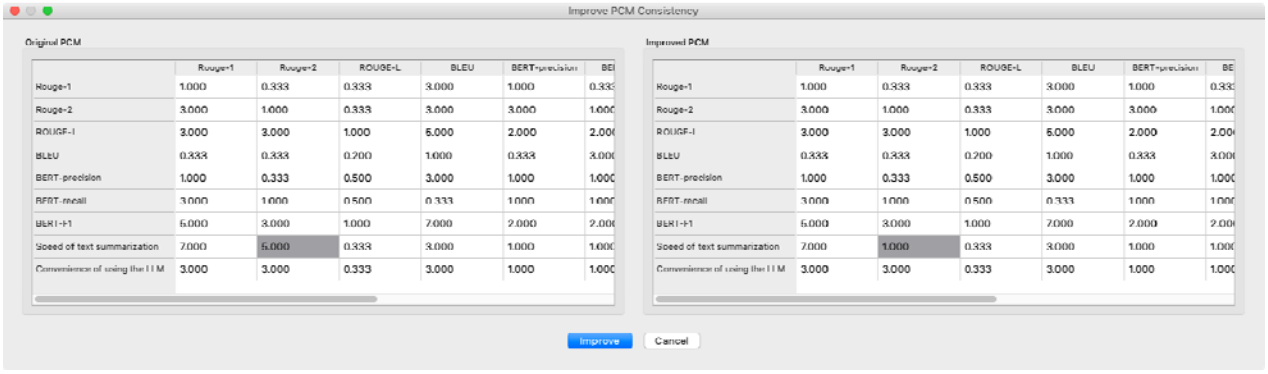

Figure 7 –The most inconsistent element of PCM (marked in gray) and its correction without the participation of an expert

The web interface for the text summarization service is developed using FastAPI, providing fast response to user actions during summarization processes which are computationally intensive.

The interface includes elements necessary for the summarization process (Fig. 8):

1. Drop-down menus to select language, model type and voice for audio feedback.
2. A text field for entering the URL of the article for which you want to generate a summary.
3. Button to start the process.
4. Areas displaying the initial text, summarized text, the area displaying the progress of the operation and the reproduced audio of the summarized text.

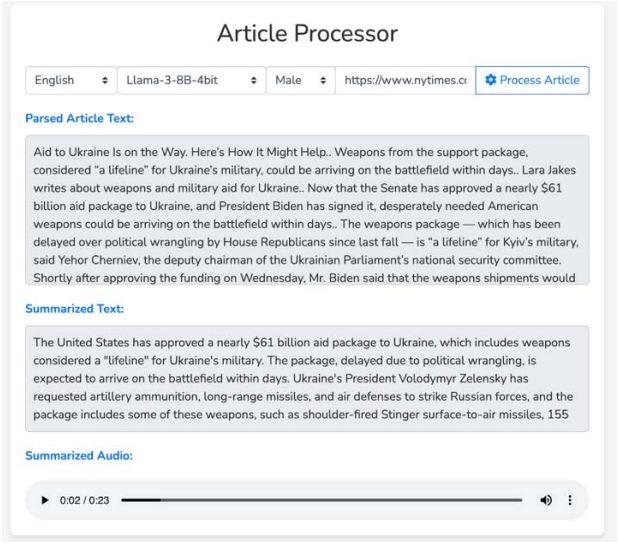

Figure 8 – Visualization of the service Web-interface after completion of all stages of summarization on the example of an arbitrary article from the New York Times

A WebSocket connection is used for real-time communication between the client and the server, allowing dynamically display updates, task progress, intermediate and final results without the need to reload the page.

To generate summarized text, the user first selects desired LLM model, text language, voice type, and specifies the URL of the article. After clicking the "Process Article" button, the request is sent to the backend, where the appropriate model is loaded based on the parameters selected by the user. The article is then downloaded, processed and summarized.

The system in real time informs the user about the status of their request with the help of a progress indicator and a direct display of the analyzed and summarized text.

A separate area of the interface displays the original text so that users can compare it with the summarized one. The summarized text is displayed together with the possibility to listen to the audio of the summarized text. Progress indicators visually show the current state of processing stages, increasing user engagement and ensuring clarity of system operation.

6 DISCUSSION

A comparative analysis of the obtained values of the metrics (Tables 1 and 2) shows that the model's size, architecture, quantization, and features of architecture significantly affect the summarization efficiency. Thus, models with more parameters tend to be capable of better understanding of context and more complex patterns in texts. However, the GPT-3.5 Turbo model with 175 billion parameters did not outperform the other considered in this study models on the ROUGE, BERT, and BLEU metrics on either the CNN/Daily Mail 3.0.0 set or the XSum dataset.

Well-structured LLMs with fewer parameters outperformed larger LLMs by some metrics in this study. Thus, the LLaMa-3 quantized model with 8 billion parameters showed the best performance in BERT metrics, and the Qwen-1.5 quantized model with 4 billion parameters was the best in ROUGE and BLEU metrics on the CNN/Daily Mail 3.0.0 set.

On the XSum set, the quantized models Gemma-1.1 and Qwen-1.5, both with 7 billion parameters, showed the highest performance in BERT metrics, and the quantized model Qwen-1.5-7B-4bit was the best among the considered models in terms of BLEU. The Phi-2 model with 2.7 billion parameters was the best according to the ROUGE metrics, showing the highest values of this metric among the other considered models.

Quantization of all considered models increased the speed of inferences based on these models. In some cases, the performance of quantized models decreased on considered metrics, but the decrease was minor compared to the significant advantages of quantized models in speed and resource utilization.

DOI 10.15588/1607-3274-2025-2-12

The obtained results indicate that quantization is a viable strategy for the LLMs that leads to a significant increase in the speed of the model inferences while maintaining acceptable levels of accuracy and consistency of summarization results.

Asymmetric uniform quantization is often used in practice, as it results in a wider and therefore more accurate range compared to symmetric quantization [39]. Asymmetric quantization, however, leads to more computationally expensive inference in comparison with the symmetric variant.

In discussed quantization methods, we need to know the range of change of the real-valued activation or weight value so that we can determine the correct scaling coefficients. This requires access to all training data. In cases where there is no access to the original training data during the quantization procedure (for example, the training data set is too large), the zero-shot quantization methods should be used.

An additional task is the integration of LLMs into existing information systems at enterprises and optimization of the dynamics of interaction with users. Users need intuitive interfaces and the ability to customize results, which requires continuous improvement in the human-machine interaction aspects of LLM applications. Having a service with LLM, which can host a large number of users at the same time, can be afforded by companies with a large budget, since this requires a large number of servers to run large language models.

CONCLUSIONS

The large size of the NN models significantly limits their ability to be deployed and used by many applications that require real-time output, low power consumption and high accuracy in conditions of limited resources. Quantization is an extremely important technology for further improving the efficiency of NLP models in conditions of limited computing resources.

The scientific novelty of obtained results is that the algorithm for LLMs' evaluation in terms of multiple criteria (metrics) is proposed, and estimates of quality for nineteen different quantized and unquantized LLMs of various sizes, including LLaMa-3-8B-4bit, Gemma-2B-4bit, Qwen-1.5-4B-4bit, Stable LM-2-1.6B-4bit, Phi-2-4bit, Mistal-7B-4bit, GPT-3.5 Turbo are obtained in terms of metrics Rouge-1,2, Rouge-L, BLEU and BERT-scores. To the best of our knowledge, such estimates of quality of the considered open-source LLMs have been obtained for the first time. The proposed algorithm for multi-criteria model's evaluation allows to choose the most appropriate model for summarizing the text, to find a compromise between the complexity of the model, its performance and operational efficiency.

The practical significance of obtained results is that the software for multiple criteria LLMs evaluation and choosing the most appropriate model for text summarization has been developed. Also a service has been developed that automatically receives text from an online article, summarizes and speaks it. The web interface for the text summarization service has been created using FastAPI, providing fast response to user actions during summarization processes which are computationally intensive.

Prospects for further research are to study the proposed algorithm for a broad class of practical problems.

ACKNOWLEDGEMENTS

This study was funded and supported by National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" (NTUU KPI) in Kyiv (Ukraine), and also financed in part of the NTUU KPI Science-Research Work by the Ministry of Education and Science of Ukraine "Development of the theoretical foundations of scenario analysis based on large volumes of semi-structured information" (State Reg. No. 0117U002150).

REFERENCES

1. Brown T., Mann B., Ryder N. et al. Language models are few-shot learners, *Advances in neural information processing systems,* 2020, Vol. 33, pp. 1877–1901. DOI: arXiv:2005.14165
2. Xie, Q. Bishop J. A., Tiwari P. et al. Pre-trained language models with domain knowledge for biomedical extractive summarization, *Knowledge-Based Systems*, 2022, Vol. 252. DOI: 10.1016/j.knosys.2022.109460
3. Basyal L., Sanghvi M. Text summarization using large language models, *ArXiv,* 2023. DOI: 2310.10449
4. OpenAI GPT 3.5 Turbo [Electronic resource]. Access mode: https://platform.openai.com/docs/models/gpt-3-5-turbo
5. OpenAI GPT-4 [Electronic resource]. Access mode: https://openai.com/index/gpt-4
6. Xu J., Ju D., Li M. et al. Recipes for safety in open-domain chatbots, *ArXiv,* 2021. DOI: 2010.07079
7. Meta LLaMa 3 [Electronic resource]. Access mode: https://llama.meta.com/llama3
8. McCulloch W. S., Pitts W. A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, 1943, Vol. 5, № 4, pp. 115– 133. DOI: 10.1007/BF02478259
9. VanRullen R. Is perception discrete or continuous? / R. VanRullen, C. Koch // Trends in cognitive sciences. – 2003. – Vol. 7, № 5. – P. 207–213. DOI: 10.1016/S1364-6613(03)00095-0
10. Tee J., Taylor D. P. Is information in the brain represented in continuous or discrete form?, *IEEE Transactions on Molecular, Biological and Multi-Scale Communications,* 2020, Vol. 6, № 3, pp. 199–209. DOI: 1805.01631
11. Faisal A. A., Selen L. P. J., Wolpert D. M. Noise in the nervous system, *Nature reviews neuroscience*, 2008, Vol. 9, № 4, pp. 292–303. DOI: 10.1038/nrn2258
12. Varshney L. R., Varshney K. R. Decision making with quantized priors leads to discrimination, *Proceedings of the IEEE,* 2016, Vol. 105, № 2, pp. 241–255. DOI:10.1109/JPROC.2016.2608741
13. Varshney L. R., Sjöström P. J., Chklovskii D. B. Optimal information storage in noisy synapses under resource constraints, *Neuron,* 2006, Vol. 52, № 3, pp. 409–423. DOI: 10.1016/j.neuron.2006.10.017
14. Hinton G., Dean J., Vinyals O. Distilling the knowledge in a neural network, *NIPS 2014 Deep Learning Workshop*, 2015, pp. 1–9. DOI: 1503.02531

DOI 10.15588/1607-3274-2025-2-12

15. Mishra A. D. Marr Apprentice: using knowledge distillation techniques to improve low-precision network accuracy, *ArXiv*, 2017. DOI: 1711.05852
16. Polino A., Pascanu R., Alistarh D. Model compression via distillation and quantization, *Proceedings of the Workshop at ICLR*, 2018. DOI: 1802.05668
17. Mikolov T., Chen K., Corrado G. et al. Efficient estimation of word representations in vector space, *Proceedings of the Workshop at ICLR, Scottsdale*, 2013, pp. 1–12. DOI: 1301.3781
18. Pennington J., Socher R., Manning C. GloVe: global vectors for word representation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar. Association for Computational Linguistics*, 2014, pp. 1532–1543.
19. Dai A. M., Le Q. V. Semi-supervised sequence learning, *Advances in neural information processing systems*, 2015. DOI: 1511.01432
20. McCann B., Bradbury J., Xiong C. et al. Learned in translation: contextualized word vectors, *Advances in neural information processing systems.* – 2017. – P. 6297–6308. DOI: 1708.00107
21. Peters M. E., Neumann M., Zettlemoyer L. et al. Dissecting contextual word embeddings: architecture and representation, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*. Brussels, Belgium, 2018, pp. 1499–1509. DOI: 10.18653/v1/D18–1179
22. Gehrmann S., Deng Y., Rush A. M. Bottom-up abstractive summarization, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium, 2018, pp. 4098–4109.
23. See A., Liu P. J., Manning C. D. Get to the point: summarization with pointer-generator networks, P*roceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Canada, 2017, pp. 1073–1083.
24. Vaswani A., Shazeer N., Parmar N. et al. Attention is all you need, *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA, USA, 2017, pp. 6000–6010.
25. Devlin J., Chang M.-W., Lee K. et al. BERT: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minnesota, 2019, pp. 4171–4186.
26. Zhang H., Cai J., Xu J., Wang J. Pretraining-based natural language generation for text summarization, *Computational natural language learning, Hong Kong*. China, 2019, pp. 789–797. DOI: 10.18653/v1/K19–1074
27. Zhang J., Zhao Y., Saleh M. et al. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization, *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 11328–11339.
28. Raffel C., Shazeer N., Roberts A.et al. Exploring the limits of transfer learning with a unified text-to-text transformer, *The Journal of Machine Learning Researc,* 2020, Vol. 21, № 1, pp. 5485–5551.
29. Clark K., Luong M.-T., Le Q. V. et al. ELECTRA: Pre–training Text Encoders as Discriminators Rather Than Generators, *8th International Conference on Learning Representations, 2020 [Electronic resource]*. Access mode: https://iclr.cc/virtual_2020/poster_r1xMH1BtvB.html
30. He P., Liu X., Gao J., Chen W. DeBERTa: decoding-enhanced BERT with disentangled attention, *ArXiv,* 2021, DOI: 2006.03654
31. Chung H. W., Hou L., Longpre S. et al. Scaling instruction-finetuned language models, *Journal of Machine Learning Research*, 2024, Vol. 25, pp. 1–53.
32. Touvron H., Lavril T., Izacard G. et al. LLaMA: open and efficient foundation language models, *ArXiv,* 2023. DOI: 2302.13971
33. Gemini [Electronic resource]. Access mode: https://gemini.google.com/
34. Jiang A. Q., Sablayrolles A., Mensch A. et al. Mistral 7B, *ArXiv*, 2023. DOI: 2310.06825
35. Labonne M. Quantize LLaMa with GGUF and llama.cpp [Electronic resource], 2023. Access mode: https://towardsdatascience.com/quantize-llama-models-with-ggml-and-llama-cpp-3612dfbcc172
36. Zmora N., Wu H., Rodge J. Achieving FP32 accuracy for INT8 inference using quantization aware training with NVIDIA TensorRT, *NVIDIA Technical Blog.* [Electronic resource]. Access mode: https://developer.nvidia.com/blog/achieving-fp32-accuracy-for-int8-inference-using-quantization-aware-training-with-tensorrt/
37. Dettmers T., Lewis M., Belkada Y. et al. LLM.int8(): 8-bit matrix multiplication for transformers at scale, *Proceedings of the 36th International Conference on Neural Information Processing Systems,* 2022, pp. 30318–30332. DOI: 2208.07339
38. Jacob B., Kligys S., Chen B. et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713. DOI: 1712.05877
39. Wu H., Judd P., Zhang X. et al. Integer quantization for deep learning inference: Principles and empirical evaluation, *ArXiv*, 2020. DOI: 2004.09602
40. McKinstry J. L., Esser S. K., Appuswamy R. et al. Discovering low-precision networks close to full-precision networks for efficient embedded inference, *ArXiv*, 2019. DOI: 1809.04191
41. Baskin C., Schwartz E., Zheltonozhskii E. et al. Uniq: Uniform noise injection for non–uniform quantization of neural networks, *ACM Transactions on Computer Systems,* 2021, Vol. 37, № 1–4, pp. 1–15. DOI: 10.1145/3444943
42. Li Y. Dong X., Wang W. Additive powers-of-two quantization: an efficient nonuniform discretization for neural networks, *ArXiv*, 2020. DOI: 1909.13144
43. Fang J., Shafiee A., Abdel-Aziz H. et al. Post-training piecewise linear quantization for deep neural networks, *Proceedings of the European Conference on Computer Vision,* 2020, pp. 69–86. DOI: 2002.00104v2
44. Jung S., Son C., Lee S. et al. Learning to quantize deep networks by optimizing quantization intervals with task loss, *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4350–4359. DOI: 1808.05779
45. Hubara I., Courbariaux M., Soudry D. et al. Binarized neural networks, *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS),* 2016, pp. 4114–4122.
46. NVIDIA V100 TENSOR CORE GPU [Electronic resource]. Access mode: https://www.nvidia.com/en-us/data-center/v100

DOI 10.15588/1607-3274-2025-2-12

47. Qin H., Gong R., Liu X. et al. Binary neural networks: a survey, *Pattern Recognition*, 2020, Vol. 105. DOI: 10.1016/j.patcog.2020.107281
48. Wang P., Hu Q., Zhang Y. et al. Two-step quantization for low-bit neural networks, *IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2018. DOI:10.1109/CVPR.2018.00460
49. Ding R., Chin T.-W., Liu Z. et al. Regularizing activation distribution for training binarized deep networks, *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. – 2019. DOI:10.1109/CVPR.2019.01167
50. Quantization aware training. Post-training quantization [Electronic resource], Access mode: https://www.tensorflow.org/model_optimization/guide/
51. Bengio Y., Léonard N., Courville A. Estimating or propagating gradients through stochastic neurons for conditional computation, *ArXiv,* 2013. DOI: 1308.3432
52. Bai Y., Wang Y.-X., Liberty E. Proxquant: Quantized neural networks via proximal operators, *ArXiv,* 2019. DOI: 1810.00861
53. Hubara I., Nahshan Y., Hanani Y. et al. Improving post training neural quantization: layer-wise calibration and integer programming, *ArXiv*, 2020. DOI: 2006.10518
54. Lin C.-Y. ROUGE: A package for automatic evaluation of summaries, *Proceedings of the Workshop on Text Summarization Branches Out*. Spain, Association for Computational Linguistics, 2004, pp. 74–81.
55. [Papineni K., Roukos S., Ward T. et al. bBleu: a method for automatic evaluation of machine translation, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Pennsylvania,* USA, 2002, pp. 311–318.
56. Zhang T., Kishore V., Wu F. et al. BERTScore: evaluating text generation with BERT, *International Conference on Learning Representations*, 2020, pp. 1–43. DOI: 1904.0967
57. Nedashkovskaya N. I. Investigation of methods for improving consistency of a pairwise comparison matrix, *Journal of the Operational Research Society*, 2018, Vol. 69, № 12, pp. 1947–1956. DOI: 10.1080/01605682.2017.1415640
58. Nedashkovskaya N. I. Estimation of the accuracy of methods for calculating interval weight vectors based on interval multiplicative preference relations, *IEEE 3rd Internatioal Conference on System Analysis & Intelligent Computing (SAIC)*, 2022. DOI: 10.1109/SAIC57818.2022.9922977
59. Nedashkovskaya N. I. Method for weights calculation based on interval multiplicative pairwise comparison matrix in decision-making models, *Radio Electronics, Computer Science, Control,* 2022, №3, pp. 155–167. DOI: 10.15588/1607-3274-2022-3-15
60. CNN/DailyMail 3.0.0 dataset [Electronic resource]. Access mode: https://huggingface.co/datasets/cnn dailymail.
61. XSum dataset [Electronic resource]. https://huggingface.co/datasets/xsum.

Received 22.01.2025.
Accepted 21.04.2025.

УДК 004.8, 519.816

ОЦІНЮВАННЯ КВАНТОВАНИХ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ В ЗАДАЧІ УЗАГАЛЬНЕННЯ ТЕКСТІВ

Недашківська Н. І. – д-р техн. наук, доцент, доцент кафедри математичних методів системного аналізу Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Інститут прикладного системного аналізу, Київ, Україна.

Єремічук Р. І. – бакалавр системного аналізу, Київ, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу підвищення ефективності глибоких штучних нейронних мереж щодо обсягу пам'яті та енергоспоживання, та багатокритеріальне оцінювання якості результатів великих мовних моделей (LLM) з урахуванням суджень користувачів в задачі сумаризації текстів. Об’єктом дослідження є процес автоматизації сумаризації текстів на основі LLM.

Мета роботи – знайти компроміс між складністю моделі LLM, її точністю та ефективністю в задачі сумаризації або узагальнення текстів.

Метод. Запропоновано алгоритм оцінювання моделей LLM за багатьма критеріями (метриками), який дозволяє обрати найбільш підходящу модель LLM для сумаризації тексту, знайти прийнятний компроміс між складністю моделі LLM, її продуктивністю та якістю узагальнення тексту. Значне підвищення точності результатів на основі нейронних мереж у задачах обробки природної мови часто досягається використанням занадто глибоких і надмірно параметризованих моделей, що суттєво обмежує здатність моделей використовуватися у задачах виводу в реальному часі, за потреби високої точності в умовах обмежених ресурсів. Пропонований алгоритм обирає прийнятну модель LLM за багатьма критеріями, такими як показники точності BLEU, Rouge-1, 2, Rouge-L, BERT-оцінки, швидкість сумаризації або іншими критеріями, які визначаються користувачем в конкретній практичній задачі інтелектуального аналізу тексту. Алгоритм включає аналіз і підвищення узгодженості суджень користувачів, оцінювання моделей LLM за кожним критерієм, агрегування локальних ваг моделей, аналіз чутливості отриманих глобальних ваг моделей.

Результати. Розроблено програмне забезпечення для автоматичного отримання текстів з онлайн-статей і сумаризації цих текстів, та для оцінювання якості моделей LLM. Отримано оцінки якості дев’ятнадцяти квантованих і неквантованих моделей LLM різних розмірів, серед яких LLaMa-3-8B-4bit, Gemma-2B-4bit, Gemma-1.1-7B-4bit, Qwen-1.5-4B-4bit, Stable LM-2-1.6B-4bit, Phi-2-4bit, Mistal-7B-4bit, GPT-3.5 Turbo за показниками BLEU, Rouge-1, Rouge-2, Rouge-L і BERT-оцінок на двох різних наборах текстів XSum та CNN/Daily Mail 3.0.0.

Висновки. Проведені експерименти підтвердили працездатність пропонованого математичного забезпечення, дозволяють рекомендувати його для використання при вирішенні задач сумаризації текстів на практиці. Перспективи подальших досліджень можуть полягати у більш глибокому аналізі метрик та критеріїв оцінювання якості сгенерованих текстів, а також експериментальному дослідженні пропонованого алгоритму на більшій кількості практичних задач обробки природної мови.

КЛЮЧОВІ СЛОВА: обмеженість ресурсів, обробка природної мови, сумаризація або узагальнення тексту, великі мовні моделі, квантизація, багатокритеріальний аналіз.

DOI 10.15588/1607-3274-2025-2-12

ЛІТЕРАТУРА

1. Language models are few–shot learners / [T. Brown, B. Mann, N. Ryder et al.] // Advances in neural information processing systems. – 2020. – Vol. 33. – P. 1877–1901. DOI: arXiv:2005.14165
2. Pre-trained language models with domain knowledge for biomedical extractive summarization / [Q. Xie, J. A. Bishop, P. Tiwari et al.] // Knowledge-Based Systems. – 2022. – Vol. 252. DOI: 10.1016/j.knosys.2022.109460
3. Basyal L. Text summarization using large language models / L. Basyal, M. Sanghvi // ArXiv. – 2023. DOI: 2310.10449
4. OpenAI GPT 3.5 Turbo [Electronic resource]. – Access mode: https://platform.openai.com/docs/models/gpt-3-5-turbo
5. OpenAI GPT-4 [Electronic resource]. – Access mode: https://openai.com/index/gpt-4
6. Recipes for safety in open–domain chatbots / [J. Xu, D. Ju, M. Li et al.] // ArXiv. – 2021. DOI: 2010.07079
7. Meta LLaMa 3 [Electronic resource]. – Access mode: https://llama.meta.com/llama3
8. McCulloch W. S. A logical calculus of the ideas immanent in nervous activity / W. S. McCulloch, W. Pitts // Bulletin of Mathematical Biophysics. –1943. –Vol. 5, № 4. – P. 115–133. DOI: 10.1007/BF02478259
9. VanRullen R. Is perception discrete or continuous? / R. VanRullen, C. Koch // Trends in cognitive sciences. – 2003. – Vol. 7, № 5. – P. 207–213. DOI: 10.1016/S1364-6613(03)00095-0
10. Tee J. Is information in the brain represented in continuous or discrete form? / J. Tee, D. P. Taylor // IEEE Transactions on Molecular, Biological and Multi-Scale Communications. – 2020. – Vol. 6, № 3. – P. 199–209. DOI: 1805.01631
11. Faisal A. A. Noise in the nervous system / A. A. Faisal, L. P. J. Selen, D. M. Wolpert // Nature reviews neuroscience. – 2008. – Vol. 9, № 4. – P. 292–303. DOI: 10.1038/nrn2258
12. Varshney L. R. Decision making with quantized priors leads to discrimination / L. R. Varshney, K. R. Varshney // Proceedings of the IEEE. – 2016. – Vol. 105, № 2. – P. 241–255. DOI:10.1109/JPROC.2016.2608741
13. Varshney L. R. Optimal information storage in noisy synapses under resource constraints / L. R. Varshney, P. J. Sjöström, D. B. Chklovskii // Neuron. – 2006. – Vol. 52, № 3. – P. 409–423. DOI: 10.1016/j.neuron.2006.10.017
14. Hinton G. Distilling the knowledge in a neural network / G. Hinton, J. Dean, O. Vinyals // NIPS 2014 Deep Learning Workshop. – 2015. – P. 1–9. DOI: 1503.02531
15. Mishra A. Apprentice: using knowledge distillation techniques to improve low-precision network accuracy / A. Mishra, D. Marr // ArXiv. – 2017. DOI: 1711.05852
16. Polino A. Model compression via distillation and quantization / A. Polino, R. Pascanu, D. Alistarh // Proceedings of the Workshop at ICLR. – 2018. DOI: 1802.05668
17. Efficient estimation of word representations in vector space / [T. Mikolov, K. Chen, G. Corrado et al.] // Proceedings of the Workshop at ICLR, Scottsdale. – 2013. – P. 1–12. DOI: 1301.3781
18. Pennington J. GloVe: global vectors for word representation / J. Pennington, R. Socher, C. Manning // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar. Association for Computational Linguistics. – 2014. – P. 1532–1543.
19. Dai A. M. Semi-supervised sequence learning / A. M. Dai, Q. V. Le // Advances in neural information processing systems. – 2015. DOI: 1511.01432
20. Learned in translation: contextualized word vectors / [B. McCann, J. Bradbury, C. Xiong et al.] // Advances in neural information processing systems. – 2017. – P. 6297–6308. DOI: 1708.00107
21. Dissecting contextual word embeddings: architecture and representation / [M. E. Peters, M. Neumann, L. Zettlemoyer et al.] // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium. – 2018. – P. 1499–1509. DOI: 10.18653/v1/D18–1179
22. Gehrmann S. Bottom-up abstractive summarization / S. Gehrmann, Y. Deng, A. M. Rush // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium. – 2018. – P. 4098–4109.
23. See A. Get to the point: summarization with pointer-generator networks / A. See, P. J. Liu, C. D. Manning // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Canada. – 2017. – P. 1073–1083.
24. Attention is all you need / [A. Vaswani, N. Shazeer, N. Parmar et al.] // 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. – 2017. – P. 6000 – 6010.
25. BERT: Pre-training of deep bidirectional transformers for language understanding / [J. Devlin, M.-W. Chang, K. Lee et al.] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minnesota. 2019. – P. 4171–4186.
26. Pretraining-based natural language generation for text summarization / [H. Zhang, J. Cai, J. Xu, J. Wang] // Computational natural language learning, Hong Kong, China. – 2019. – P. 789–797. DOI: 10.18653/v1/K19–1074
27. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization / [J. Zhang, Y. Zhao, M. Saleh et al.] // Proceedings of the 37th International Conference on Machine Learning. – 2020. – P. 11328–11339.
28. Exploring the limits of transfer learning with a unified text–to–text transformer / [C. Raffel, N. Shazeer, A. Roberts et al.] // The Journal of Machine Learning Research. – 2020. – Vol. 21, № 1. – P. 5485–5551.
29. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators / [K. Clark, M.-T. Luong, Q. V. Le et al.] // 8th International Conference on Learning Representations, 2020 [Electronic resource]. – Access mode: https://iclr.cc/virtual_2020/poster_r1xMH1BtvB.html
30. DeBERTa: decoding-enhanced BERT with disentangled attention / [P. He, X. Liu, J. Gao, W. Chen] // ArXiv. – 2021. DOI: 2006.03654
31. Scaling instruction-finetuned language models / [H. W. Chung, L. Hou. S. Longpre et al.] // Journal of Machine Learning Research. – 2024. – Vol. 25. – P. 1–53.
32. LLaMA: open and efficient foundation language models / [H. Touvron, T. Lavril, G. Izacard et al.] // ArXiv. – 2023. DOI: 2302.13971
33. Gemini [Electronic resource]. – Access mode: https://gemini.google.com/
34. Mistral 7B / [A. Q. Jiang, A. Sablayrolles, A. Mensch et al.] // ArXiv. – 2023. DOI: 2310.06825
35. Labonne M. Quantize LLaMa with GGUF and llama.cpp [Electronic resource] / M. Labonne. – 2023. – Access mode:

DOI 10.15588/1607-3274-2025-2-12

https://towardsdatascience.com/quantize-llama-models-with-ggml-and-llama-cpp-3612dfbcc172
36. Zmora N. Achieving FP32 accuracy for INT8 inference using quantization aware training with NVIDIA TensorRT / N. Zmora, H. Wu, J. Rodge // NVIDIA Technical Blog. [Electronic resource]. – Access mode: https://developer.nvidia.com/blog/achieving-fp32-accuracy-for-int8-inference-using-quantization-aware-training-with-tensorrt/
37. LLM.int8(): 8-bit matrix multiplication for transformers at scale / [T. Dettmers, M. Lewis, Y. Belkada et al.] // Proceedings of the 36th International Conference on Neural Information Processing Systems. – 2022. – P. 30318–30332. DOI: 2208.07339
38. Quantization and training of neural networks for efficient integer-arithmetic-only inference / [B. Jacob, S. Kligys, B. Chen et al.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – P. 2704–2713. DOI: 1712.05877
39. Integer quantization for deep learning inference: Principles and empirical evaluation / [H. Wu, P. Judd, X. Zhang et al.] // ArXiv. – 2020. DOI: 2004.09602
40. Discovering low-precision networks close to full-precision networks for efficient embedded inference / [J. L. McKinstry, S. K. Esser, R. Appuswamy et al.] // ArXiv. – 2019. DOI: 1809.04191
41. Uniq: Uniform noise injection for non–uniform quantization of neural networks / [C. Baskin, E. Schwartz, E. Zheltonozhskii et al.] // ACM Transactions on Computer Systems. – 2021. – Vol. 37, № 1–4. – P. 1–15. DOI: 10.1145/3444943
42. Li Y. Additive powers–of–two quantization: an efficient nonuniform discretization for neural networks / Y. Li, X. Dong, W. Wang // ArXiv. – 2020. DOI: 1909.13144
43. Post-training piecewise linear quantization for deep neural networks / [J. Fang, A. Shafiee, H. Abdel-Aziz et al.] // Proceedings of the European Conference on Computer Vision. – 2020. – P. 69–86. DOI: 2002.00104v2
44. Learning to quantize deep networks by optimizing quantization intervals with task loss / [S. Jung, C. Son, S. Lee et al.] // Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – P. 4350–4359. DOI: 1808.05779
45. Binarized neural networks / [I. Hubara, M. Courbariaux, D. Soudry et al.] // Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS). – 2016. – P. 4114–4122.
46. NVIDIA V100 TENSOR CORE GPU [Electronic resource]. – Access mode: https://www.nvidia.com/en-us/data-center/v100
47. Binary neural networks: a survey / [H. Qin, R. Gong, X. Liu et al.] // Pattern Recognition. – 2020. – Vol. 105. DOI: 10.1016/j.patcog.2020.107281
48. Two-step quantization for low-bit neural networks / [P. Wang, Q. Hu, Y. Zhang et al.] // IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2018. DOI:10.1109/CVPR.2018.00460
49. Regularizing activation distribution for training binarized deep networks / [R. Ding, T.-W. Chin, Z. Liu et al.] // IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. DOI:10.1109/CVPR.2019.01167
50. Quantization aware training. Post-training quantization [Electronic resource]. – Access mode: https://www.tensorflow.org/model_optimization/guide/
51. Bengio Y. Estimating or propagating gradients through stochastic neurons for conditional computation / Y. Bengio, N. Léonard, A. Courville // ArXiv. – 2013. DOI: 1308.3432
52. Bai Y. Proxquant: Quantized neural networks via proximal operators / Y. Bai, Y.-X. Wang, E. Liberty // ArXiv. – 2019. DOI: 1810.00861
53. Improving post training neural quantization: layer-wise calibration and integer programming / [I. Hubara, Y. Nahshan, Y. Hanani et al.] // ArXiv. – 2020. DOI: 2006.10518
54. Lin C.-Y. ROUGE: A package for automatic evaluation of summaries / C.-Y. Lin // Proceedings of the Workshop on Text Summarization Branches Out, Spain. Association for Computational Linguistics. – 2004. – P. 74–81.
55. Bleu: a method for automatic evaluation of machine translation / [K. Papineni, S. Roukos, T. Ward et al.] // Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Pennsylvania, USA. – 2002. – P. 311–318.
56. BERTScore: evaluating text generation with BERT / [T.Zhang, V. Kishore, F. Wu et al.] // International Conference on Learning Representations. – 2020. – P. 1–43. DOI: 1904.0967
57. Nedashkovskaya N. I. Investigation of methods for improving consistency of a pairwise comparison matrix / N. I. Nedashkovskaya // Journal of the Operational Research Society. – 2018. – Vol. 69, № 12, P. 1947–1956. DOI: 10.1080/01605682.2017.1415640
58. Nedashkovskaya N. I. Estimation of the accuracy of methods for calculating interval weight vectors based on interval multiplicative preference relations / N. I. Nedashkovskaya // IEEE 3rd Internatioal Conference on System Analysis & Intelligent Computing (SAIC). – 2022. DOI: 10.1109/SAIC57818.2022.9922977
59. Nedashkovskaya N. I. Method for weights calculation based on interval multiplicative pairwise comparison matrix in decision-making models / N. I. Nedashkovskaya // Radio Electronics, Computer Science, Control. – 2022. – №3. – P. 155–167. DOI: 10.15588/1607-3274-2022-3-15
60. CNN/DailyMail 3.0.0 dataset [Electronic resource]. – Access mode: https://huggingface.co/datasets/cnn dailymail.
61. XSum dataset [Electronic resource]. – https://huggingface.co/datasets/xsum.

DOI 10.15588/1607-3274-2025-2-12