Krzysztof Ociepa
Łukasz Flis
Krzysztof Wróbel
Adrian Gwoździej
Remigiusz Kinas

BIELIK 7B V0.1: POLISH LANGUAGE MODEL – DEVELOPMENT, INSIGHTS, AND EVALUATION

Abstract *We introduce Bielik 7B v0.1 – a seven-billion-parameter generative text model for Polish language processing. Trained on curated Polish corpora, this model addresses key challenges in language model development through innovative techniques; these include Weighted Instruction Cross-Entropy Loss (which balances the learning of different instruction types) and Adaptive Learning Rate (which dynamically adjusts the learning rate based on training progress). To evaluate performance, we created the Open PL LLM Leaderboard and Polish MT-Bench – novel frameworks assessing various NLP tasks and conversational abilities. Bielik 7B v0.1 demonstrates significant improvements, achieving a nine-percentage-point increase in its average score compared to Mistral-7B-v0.1 on the RAG Reader task. It also excels in the Polish MT-Bench – particularly in the Reasoning (6.15/10) and Role-playing (7.83/10) categories. This model represents a substantial advancement in Polish language AI, offering a powerful tool for diverse linguistic applications and setting new benchmarks in the field.*

Keywords Polish language model, natural language processing, transformer architecture, language model evaluation, instruction tuning

Citation Computer Science 26(4) 2025: 131–161

1. Introduction

The rapid advancement in natural language processing (NLP) has led to the development of increasingly sophisticated language models that can understand and generate human-like text. These models have shown remarkable success in various linguistic tasks across multiple languages. However, the development of high-performing models for less-resourced languages remains a significant challenge due to the scarcity of large and diverse data sets and computational resources.

Existing Polish language models such as TRURL 2 [49] and Qra [30] have made important strides in this domain. TRURL 2, a collection of fine-tuned Llama 2 models with 7 billion and 13 billion parameters, was trained on approximately one million conversational Polish and English samples, with a context size of 4096 tokens. Another series of models is Qra, which comprises continuously pretrained models with 1, 7, and 13 billion parameters. The Qra models were trained on Polish data, totaling 90 billion tokens and also employing a context size of 4096 tokens. While numerous other Polish-focused language models exist, the majority of them are fine-tuned using significantly smaller data sets or fine-tuning approaches, which can limit their performance and versatility.

This paper introduces Bielik 7B v0.1 – a state-of-the-art Polish language model developed as a collaborative effort between the SpeakLeash open-science project and the High Performance Computing (HPC) center: ACK Cyfronet AGH. Bielik 7B v0.1 is an evolution of the Mistral 7B v0.1 model [19] thatwas enhanced to understand and generate Polish text with high accuracy. This model leverages a massive corpus of Polish texts and advanced machine-learning techniques, making it a pioneering tool in the realm of Polish Natural Language Processing (NLP). The development of Bielik 7B v0.1 addresses several challenges, including the adaptation of a model trained primarily on English data to the Polish language; this involved significant linguistic and semantic adjustments.

In the following sections, we detail the architecture of Bielik 7B v0.1, describe the data set preparation, discuss the training process, and evaluate the model's performance on various NLP tasks. Our results demonstrated that Bielik 7B v0.1 not only advances the state of Polish language understanding but also serves as a valuable resource for further research and application in Polish NLP.

2. Model and tokenizer

In this section, we introduce the model design and tokenizer, presenting architectural decisions and configurations.

2.1. Model architecture

The Bielik 7B v0.1 model builds on the Transformer architecture [48] (with its key parameters detailed in Table 1) and incorporates a suite of advanced techniques to enhance its performance.

Table 1
Model architecture

Parameter	Value
Layers	32
Model Dimension	4096
Attention Heads	32
Key/Value Heads	8
Head Size	128
Intermediate Size	14,336
Activation Function	SwiGLU
Vocabulary Size	32,000
Positional Embeddings	RoPE ($\theta = 10,000$)
Context Length	8192
Sliding Window	4096

Self-attention with causal masks [48] allows the model to weigh the importance of different parts of an input sequence. The causal mask ensures that the model only attends to previous tokens, which is crucial for maintaining the autoregressive property in language-modeling tasks.

Grouped-query attention (GQA) [1] reduces computational complexity and memory usage while maintaining model quality. It achieves this by using fewer key-value heads than query heads, allowing for the more efficient processing of long sequences.

Sliding Window Attention [6, 8] limits the attention span to a fixed window size, reducing the computational complexity from quadratic to linear in sequence length. This enables the model to process longer sequences more efficiently while still capturing local context effectively.

SwiGLU activation function [12,41] is a combination of the Swish activation function and Gated Linear Units (GLU), offering improved performance and trainability compared to traditional activation functions like ReLU.

Rotary Positional Embeddings (RoPE) [45] allow the model to better capture the relative positions of tokens in an input sequence, offering advantages over absolute positional embeddings. It excels in tasks requiring nuanced understanding of token positions, providing better extrapolation to longer sequences and improving overall performance.

Root Mean Square Layer Normalization (RMSNorm) [20] is used for normalizing activations within a network. It offers improved training stability and slightly faster computation compared to traditional Layer Normalization, contributing to more efficient training and inference.

Pre-normalization applies layer normalization before the self-attention and feed-forward layers rather than after, resulting in improved model convergence and overall performance.

The Bielik 7B v0.1 model was adapted from the Mistral 7B v0.1 model and further pretrained. The decision to use an existing model instead of training our own from scratch was due to the lack of access to sufficient high-quality data. Additionally, training from scratch would have required significantly more resources, including GPU power and time. We chose the Mistral 7B v0.1 model because of its strong performance in benchmarks and its permissive Apache 2.0 license.

2.2. Tokenizer

One measure of the effectiveness of the tokenization process is the count of tokens generated for the input text. A lower number of tokens indicates faster and more efficient text generation by a language model. The tokenizer from the Mistral 7B model was not specifically trained for the Polish language; therefore, we conducted a series of experiments aimed at expanding the original tokenizer to include Polish tokens. Our approach to expanding the tokenizer involved incorporating tokens from the Polish APT3 model [32] by extending the model's edge layers (embeddings and language model head) and continuing the training process. We chose the preamble of the Constitution of the Republic of Poland as the benchmark text because it effectively captures the essence of Polish writing and includes official English versions for comparative analysis. Table 2 presents a detailed comparison of various metrics, including the token count, characters per token (CpT), and tokens per word (TpW). These metrics illustrate the performance of different tokenizers when applied to both the Polish and English versions of the preamble.

Table 2

Comparison of token count, characters per token (CpT), and tokens per word (TpW) for preamble of Constitution of Republic of Poland in Polish and English versions processed by various tokenizers: APT3 (dedicated Polish language tokenizer); Llama2 and Mistral v0.1 (multilingual tokenizers with minimal Polish support); and merged tokenizers Llama2 + APT3 and Mistral v0.1 + APT3

	Vocab	Avg	Polish			English		
Tokenizer	size	tokens	Tokens	CpT	TpW	Tokens	CpT	TpW
APT3	31,980	480	344	5.22	1.48	615	3.15	1.93
Llama2	32,000	554	681	2.63	2.94	427	4.53	1.34
Mistral v0.1	32,000	578	747	2.40	3.22	408	4.75	1.28
Llama2 + APT3	57,362	442	441	4.07	1.90	442	4.38	1.39
Mistral v0.1 + APT3	58,690	450	493	3.64	2.12	407	4.76	1.28

Despite achieving good results on benchmarks with the trained models, we observed issues in text generation; these occasionally manifested as incorrect token combinations for Polish words. This problem arose partly due to the ambiguity

that occurs when merging pairs of tokens during the tokenization process [18]. This process utilizes the byte pair encoding (BPE) algorithm [40], which is implemented through SentencePiece [24]. Since the tokens from the APT3 model tokenizer and the Mistral 7B model tokenizer are not mutually exclusive (their vocabularies overlap), ambiguity arises during the merging of token pairs, making it impossible to directly combine both tokenizers.

In light of these issues, we decided to retain the original tokenizer from the Mistral 7B model (which has a vocabulary size of 32,000 tokens) while continuing to explore potential expansion options for future model versions.

3. Pre-training

The primary objective of the pre-training phase was to enhance the model's Polish language capabilities, focusing on both accuracy and fluency. To accomplish this, we employed a diverse selection of high-quality Polish texts. These materials were subjected to rigorous cleaning procedures and meticulous quality evaluations, ensuring the highest standard of the training data.

3.1. Pre-training data

The pre-training of the Bielik model involved constructing a novel, diverse, and high-quality data set that was primarily made up of Polish language texts. We leveraged resources from the SpeakLeash project [44]. Using metadata assigned to each document (which included information about its topic and various stylometric features), we selected 18 million documents from different data sets that offered high quality and topic diversity. These selected texts underwent thorough cleaning and quality assessment procedures (detailed in Sections 3.1.1 and 3.1.2). Additionally, we removed documents where, although robots.txt did not prohibit scraping, the terms and conditions explicitly forbade using them for training language models. Only documents meeting our stringent quality criteria were retained for training and subsequently tokenized. This meticulous process yielded a training data set comprising 22 billion tokens. To improve the model's adaptation to a new language and mitigate catastrophic forgetting [17, 25, 35], we supplemented our training data set with English texts; these were sourced from the SlimPajama data set [43], which is known for its diverse and high-quality English content. Ultimately, our final training data set consisted of 36 billion tokens.

3.1.1. Data cleanup

The foundation of our pre-training corpus was a broad collection of texts from the Polish web (including processed data from the CulturaX and HPLT data sets) supplemented with digitized library resources and publicly available documents. To improve the quality of this data, we implemented a series of heuristics aimed at removing damaged and unwanted text fragments, anonymizing personal data (such as physical

addresses, email addresses, phone numbers, and URLs), and fixing any encoding or formatting issues. As a result of this process, we obtained higher-quality texts, which were ready for a detailed quality assessment.

3.1.2. Quality evaluation

To create the training data set for text quality evaluation, we manually annotated 9000 documentsand assigned each to one of three quality classes: HIGH, MEDIUM, or LOW. To ensure high internal consistency in the evaluation, the entire labeling process was carried out by a single specialized annotator. The classification criteria were defined as follows:

- **HIGH class** included documents of high substantive value, characterized by clear, logical, and well-formatted text. Minor formatting errors were permissible if they did not affect the readability (e.g., in valuable content from internet forums or official documents).
- **LOW class** was comprised of clearly problematic texts containing encoding errors, broken formatting (e.g., incorrectly converted tables), thematically mixed-up or truncated fragments, and vulgar content or texts consisting mainly of non-linguistic data (e.g., financial reports).
- **MEDIUM class** served as a buffer for documents of a mixed nature that contained both valuable fragments and significant flaws (e.g., an article snippet surrounded by website interface elements like menus or headers). This class identified texts with potential for future recovery f(or instance, through automated correction).

The lower prevalence of the MEDIUM class in the manually annotated data set (as shown in Figure 1) stemmed from both its natural rarity in the source data and the fact that the annotation process prioritized creating a balanced set for the clearly defined HIGH and LOW classes (whose assessments were less time-consuming).

For each document, we calculated 266 stylometric features, including metrics such as the frequency of verbs, nouns, sentences, and punctuation marks. This comprehensive set of features was derived based on the methodology outlined in the StyloMetrix tool [34]. These linguistic and structural attributes provided a multifaceted representation of each text's stylistic properties.

Using these stylometric features as input, we trained an XGBoost classifier model. This machine-learning approach allowed us to leverage the complex interactions among various textual characteristics to predict document quality effectively (as presented in Table 3 and Figure 1).

To determine an appropriate threshold for identifying high-quality documents, we conducted a manual analysis of 1000 documents. Based on this thorough examination, we established a cut-off point for the HIGH category at a probability score exceeding 90%. Documents that did not meet this threshold were excluded from the target training data set of the Bielik model.

Table 3
Validation results for XGBoost classifier model

Precision	Recall	F1
0.8640	0.8285	0.8431

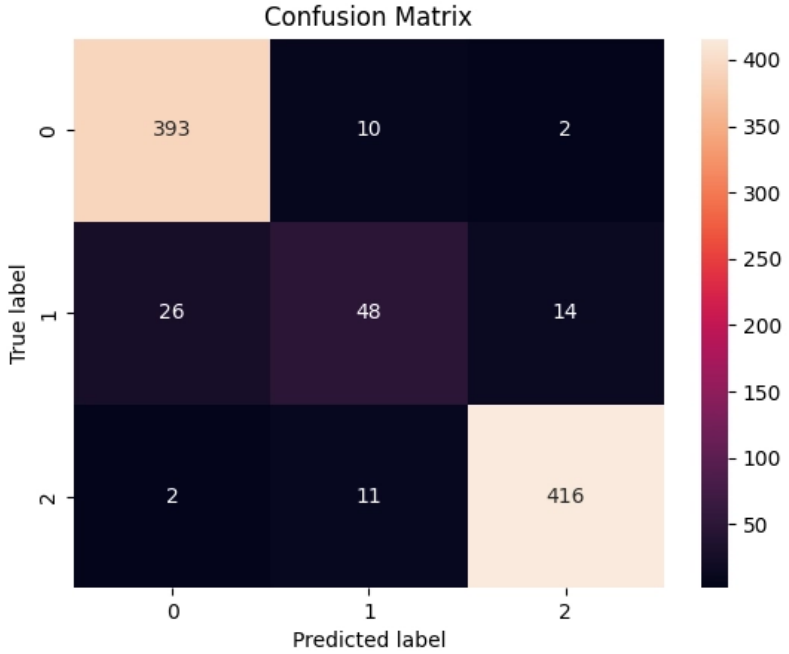

Figure 1. Confusion matrix illustrating validation results for XGBoost classifier model

3.2. Training hyperparameters

We utilized the AdamW optimizer [27] with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay = 0.1. The learning rate followed a cosine decay schedule starting at 3e-05 and decreasing to 2e-05, with a warmup period of 2000 iterations. The training continued for a total of 17,350 iterations. We employed a global batch size of 256, which were composed of local batches of a size of 4. The gradient clipping norm was set to 1.0, and we used mixed precision with bfloat16. The model was trained on 36B tokens over 2 epochs, with a maximum context length of 4096. The training loss and accuracy over the training tokens for the base model are presented in Figures 2 and 3.

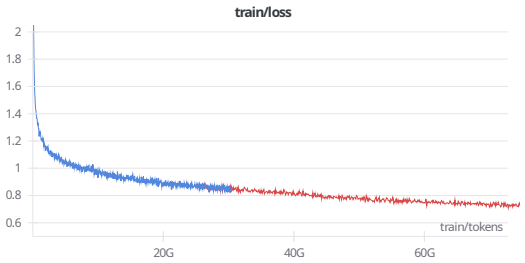

Figure 2. Training loss over training tokens for base model

Figure 3. Training accuracy over training tokens for base model

4. Post-training

After finishing the pre-training phase, we moved on to the post-training phase, which focused on improving the model's capabilities across various areas such as coding, mathematics, logical reasoning, and following instructions.

4.1. Post-training data

A significant challenge in developing high-performing models for the Polish language is the scarcity of large-scale open-source instruction data sets. To address this gap, we initiated the creation of a comprehensive Polish instruction-following data set, which was continuously expanded and refined. Our approach combined manually authored data with a large-scale automated generation pipeline, supplemented by high-quality English data sets.

4.1.1. Polish instruction data set creation

Our Polish data set consisted of two primary components:

Manually Authored Instructions: a high-quality set of instructions was developed by our annotators. This collection focused on tasks that required deep linguistic understanding or could be programmatically verified. Among others, the cate-

gories included text classification, sentiment analysis, and natural language-processing tasks such as part-of-speech tagging (e.g., identifying verbs or nouns using libraries like Morfeusz and spaCy).

Automatically Generated Instruction: to augment the manually curated data, we developed a large-scale automatic generation pipeline. We selected a diverse collection of one million high-quality articles from our pre-training corpus (see Section 3.1). Using the Mixtal 8x22B model, we generated instruction-response pairs for each article, covering tasks such as summarization, question answering based on a provided context, and email composition.

4.1.2. Quality assurance and data composition

To ensure the quality of the automatically generated data, we employed an LLM-as-a-judge methodology for large-scale evaluation. To further validate this automated assessment, approximately 1000 instruction-response pairs were spot-checked manually. These carefully verified examples now serve as a "gold standard" set, which can be used for future evaluation and to guide further data generation.

To further increase the number and diversity of the instructions, we utilized publicly accessible collections of English instructions such as the OpenHermes-2.5 [46] and orca-math-word-problems-200k [29] data sets. These English-language resources accounted for approximately half of the instructions used in the final training mixture.

As a result, we compiled a final training data set containing over 2.3 million instructions, amounting to more than 700 million tokens. To support transparency and enable further research in the Polish language domain, we plan to release a representative portion of our Polish instruction data set in the future.

4.2. Supervised fine-tuning

The varying quality of training instructions negatively impacts a model's benchmark performance (as was demonstrated in previous studies); they found that poor-quality instructions degraded the models' capabilities [56]. These studies showed that smaller, higher-quality instruction data sets often yielded better results than larger, noisier data sets. To address this, we introduced several improvements (summarized below) while still utilizing the previously mentioned data sets.

4.2.1. Masked tokens

We employed a masked token approach, selectively applying loss only to certain parts of the output. Specifically, we masked the loss on user instruction and control tokens [42]. This technique ensured that these tokens did not contribute to the overall loss during training, allowing the model to focus on learning from the content tokens.

4.2.2. Adaptive learning rate

The lengths of instructions can vary significantly, leading to fluctuations in the numbers of tokens used in computing the loss function. To ensure consistent influence

from each instruction, we implemented an adaptive learning rate (ALR). This approach was based on prior research that linked learning rates to batch sizes [15]. In particular, the learning rate (LR) was scaled according to the square root of the ratio between the number of tokens in the batch (T) and the baseline batch size (BS):

$$\mathrm{ALR} = \mathrm{LR} \cdot \sqrt{\frac{\mathrm{T}}{\mathrm{BS}}} \tag{1}$$

4.2.3. Weighted instruction cross-entropy loss

This strategy (inspired by weighted cross-entropy loss [52], offline reinforcement learning [53], and C-RLFT [50]) enabled us to effectively utilize mixed-quality training data annotated with fine-grained weight labels.

Given the $\mathcal{D} = (x_i, y_i)$ SFT conversation data set (where x_i indicates the instruction, y_i is its corresponding response), we assigned a weight $w_i \in (0, 1]$ to each instruction-response pair (x_i, y_i) that represented the quality of the pair. This allowed us to construct a weighted data set, $\mathcal{D}_w$, where the highest-quality pairs were assigned a weight of 1.0 while assigning the lower-quality instructions smaller weights ($w_i < 1.0$). We can express the relationship between the weights and quality as follows:

$$w_i = \begin{cases} 1.0 & \text{highest quality} \\ \alpha & \text{lower quality} \quad (0 < \alpha < 1) \end{cases} \tag{2}$$

This weighting scheme guides the model to favor high-quality responses while still learning from a diverse range of instruction-response pairs. We labeled our data set as described in Section 4.1 and assigned weights to the instruction-response pairs based on predefined rules:

$$w_i = \begin{cases} 1.0 & \text{high quality} \\ 0.7 & \text{medium quality} \\ 0.5 & \text{low quality} \end{cases} \tag{3}$$

where:

high quality – instructions and dialogues manually written by annotators, OpenHermes-2.5 [46], and orca-math-word-problems-200k [29] data sets;

medium quality – generated instructions based on pre-training data that was manually verified and corrected;

low quality – generated instructions based on pre-training data without manual verification.

We include low-quality instructions with reduced weight (0.5) for several reasons. First, they significantly increase training data volume and diversity, exposing the model to a broader range of linguistic patterns and edge cases that may not be well-represented in smaller high-quality data sets. Second, the weighted loss mechanism ensures these samples contribute less to the optimization objective while still

providing useful training signal – effectively allowing the model to learn from imperfect data without compromising performance on high-quality benchmarks. Third, completely discarding these automatically generated samples would waste potentially valuable information, as many contain correct task structures and partially useful content despite the quality limitations. This approach of learning from mixed-quality data with appropriate weighting has been shown to improve model robustness and generalization [7, 56].

Supervised Fine-Tuning (SFT) methods are designed to adapt a pre-trained language model π_0 into a fine-tuned model π_{SFT} using a high-quality $\mathcal{D}$ instruction data set and supervised learning. We use $\pi(y|x)$ to represent the probability of generating response y given instruction x in the $\mathcal{D}$ data set. The objective of SFT can be expressed as a maximum likelihood estimate (MLE):

$$J_{\text{SFT}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\log \pi_{\text{SFT}}(y|x)] \tag{4}$$

To ensure optimal fine-tuning performance, SFT requires the $\mathcal{D}$ instruction data set to be of the highest-possible quality, as it treats all training data uniformly [7, 56]. However, assembling a sufficiently large and high-quality data set can be both time-consuming and financially expensive.

In practice, the quality of available instructions often varies. It is possible that valuable and informative instructions may have lower quality levels than desired. To leverage the potential of such mixed-quality data, we introduce the weighted instruction cross-entropy loss, which guides the learning process to prioritize more preferred answers while still allowing the model to learn valuable insights from lower-quality instructions.

The standard Weighted Cross-Entropy Loss [22], originating from the weighted exogenous sampling maximum-likelihood estimator, is frequently used in multi-class classification problems [52]. It is commonly employed, for instance, to address imbalanced class distributions [37]. We can formulate standard Weighted Cross-Entropy Loss as follows:

$$l(o_i, y_i) = -\sum_{c=1}^{C} w_c \cdot y_{i,c} \cdot \log p_{i,c} \tag{5}$$

where C is the number of classes, $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,C}) \in \{0,1\}^C$ is the one-hot encoding of the ground truth label for sample x_i, and $y_{i,c} = 1$ indicates that x_i belongs to class c. Meanwhile, $\mathbf{p}_i = (p_{i,1}, \ldots, p_{i,C}) \in \mathbb{R}_+^C$ represents the predicted probability vector for sample x_i across C classes. In multi-class classification problems using deep-neural networks, p_i corresponds to the softmax values of the logits for each class produced by the last layer of the network. Specifically, $p_{i,c} = \frac{\exp(o_{i,c})}{\sum_{j=1}^{C} \exp(o_{i,j})}$, where $o_{i,c}$ is the logit for class c for sample x_i.

To integrate fine-grained weights from the $\mathcal{D}_w$ data set, we modified Equation (5) as follows:

$$l(o_i, y_i) = -w_i \cdot \sum_{c=1}^{C} y_{i,c} \cdot \log p_{i,c} \tag{6}$$

where w_i represents the weight assigned to the instruction-response pair (x_i, y_i). This learning objective provides a flexible framework for fine-tuning language models, offering more granular control over the importance of each instruction during training. It can capture subtle differences in data quality while maintaining computational efficiency.

4.3. Training hyperparameters

We applied the AdamW optimizer using $\beta_1 = 0.9$, $\beta_2 = 0.95$ and a weight decay of 0.05. The adaptive learning rate followed a cosine decay – starting at 7e-6, and tapering down to 6e-7 (with 50 warmup iterations). The training process spanned a total of 55,440 iterations. Our setup used a global batch size of 128, which was made up of local batches with a size of 1. Gradient clipping was enforced with a threshold of 1.0, and the model was trained in mixed precision using bfloat16. We trained the model for 3 epochs with a maximum context length of 4096 – processing a total of 2.1 billion tokens. The training loss, accuracy, and adaptive learning rate over the training iterations for the instruction model are presented in Figures 4, 5, and 6.

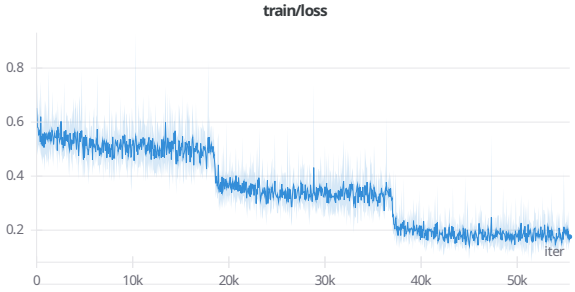

Figure 4. Training loss over training iterations for instruction model

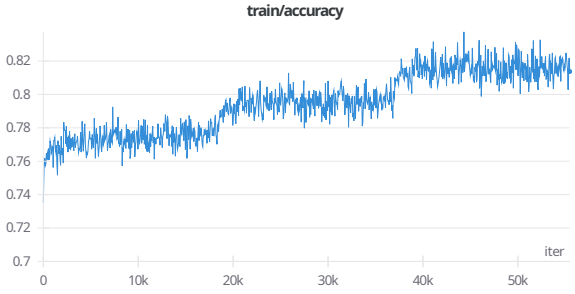

Figure 5. Training accuracy over training iterations for instruction model

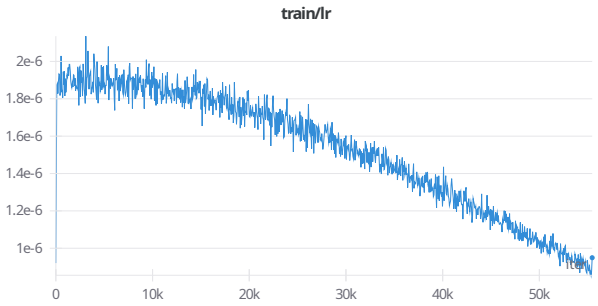

Figure 6. Adaptive learning rate over training iterations for instruction model

4.4. Efficient implementation

For our training needs, we utilized the ALLaMo framework [31] that was developed by a co-author of the Bielik model to optimize the training throughput. This framework allowed us to maximize the computational resources of the supercomputer, enabling faster calculations and reducing the overall training time. ALLaMo achieves high efficiency through numerous optimizations at the dataloader, model, and training process levels, along with a strong reliance on torch.compile in conjunction with an efficient attention implementation using PyTorch SDPA and the PyTorch Fused AdamW optimizer [2]. A significant advantage of ALLaMo is its primary reliance on PyTorch without dependencies on other popular frameworks or libraries; this allows for the better optimization and easier implementation of functionalities not available in other frameworks. For the post-training, we implemented the weighted instruction cross-entropy loss and adaptive learning rate strategies (detailed in Sections 4.2.3 and 4.2.2). These improvements enabled us to efficiently conduct numerous experiments and successfully complete the final model training. During the base model training, we utilized 256 NVIDIA GH200 GPUs, achieving a throughput of over 9200 tokens per GPU per second.

To validate the performance of the ALLaMo framework, we conducted a comparison with the implementation used in training the TinyLlama model [54]. The authors of this model introduced numerous improvements to accelerate the training, including FlashAttention-2 [11], fused LayerNorm, fused SwiGLU, fused Cross-Entropy Loss, and fused Rotary Positional Embeddings. The experiment was carried out on A100 40GB GPUs in 8x and 16x A100 configurations using a model with parameters identical to the TinyLlama 1.1B model. When using ALLaMo, it was possible to increase the local batch size from 8 to 9, which further enhanced the training throughput. Table 4 illustrates the performance differences between the TinyLlama implementation and the ALLaMo framework.

Table 4
Comparison of training performance between TinyLlama implementation and ALLaMo framework

Framework	Configuration	Total Batch Size	Throughput
TinyLlama	8xA100 40 GB	2,097,152 tokens	24,390 tokens/GPU/sec
ALLaMo	8xA100 40 GB	2,097,152 tokens	26,150 tokens/GPU/sec (+7.2%)
ALLaMo	8xA100 40 GB	2,359,296 tokens	26,550 tokens/GPU/sec (+8.8%)
TinyLlama	16xA100 40 GB	2,097,152 tokens	24,000 tokens/GPU/sec [1]
ALLaMo	16xA100 40 GB	2,097,152 tokens	25,850 tokens/GPU/sec (+7.7%)
ALLaMo	16xA100 40 GB	2,359,296 tokens	26,000 tokens/GPU/sec (+8.3%)

[1] Value reported by authors of model

5. Evaluations

5.1. Open PL LLM leaderboard

The Open PL LLM Leaderboard (based on the Open LLM Leaderboard v1 [5]) evaluates models on various NLP tasks, including sentiment analysis, categorization, and text classification; however, it does not test their conversational capabilities [51]. The leaderboard utilizes the lm-evaluation-harness framework for model evaluation [14].

Tasks:

- **polemo2:** Sentiment analysis of online consumer reviews across four domains (medicine, hotels, products, university) with four-class labeling (positive, negative, neutral, ambiguous) [23]; metric: accuracy.
- **klej-ner:** Named entity recognition in sentences containing single-type entities, classifying into six categories (no entity, place, person, organization, time, geographical name) [38]; metric: accuracy.
- **8tags:** Topic classification of social media headlines into eight categories (film, history, food, medicine, motorization, work, sport, technology) [10]; metric: accuracy.
- **belebele:** Machine-reading comprehension for question answering [4]; metric: accuracy.
- **dyk:** Question answering based on human-annotated pairs from Wikipedia's "Did You Know" section [28]; metric: binary F1.
- **ppc:** Text-similarity assessment using manually labeled sentence pairs (exact paraphrases, close paraphrases, non-paraphrases) [9]; metric: accuracy.
- **psc:** Summarization of news articles [33]; metric: binary F1.
- **cbd:** Text classification for cyberbullying and hate-speech detection [36]; metric: macro F1.
- **polqa:** Open-domain question answering from the "Jeden z dziesięciu" TV show, with and without context (abstractive QA/RAG) [39]; metric: accuracy, levenshtein.

- **poquad:** Context-based extractive question answering (QA/RAG) [47]; metric: levenshtein.

Most of the tasks are multiple-choice tests, which means that the model chooses the correct answer from a set of options. These are implemented as two types of tests:

- **Loglikelihood:** we choose highest probability token from given set (e.g., ABCD) – these tests are suitable for base models;
- **Generate:** model generates answer freely.

All tasks are evaluated in both 0-shot and 5-shot settings.

Evaluation scores:

- **All tasks:** average score across all tasks, normalized by baseline scores;
- **Reranking:** score of the polqa_reranking_mc task, which is based on polqa data set – this task evaluates model's ability to determine whether given context is relevant to question (binary relevance classification) (this capability is essential in Retrieval-Augmented Generation [RAG] systems for reranking retrieved documents based on their relevance to query);
- **Reader (Generator):** average score of open-book question-answering tasks (polqa and poquad), which evaluate model's ability to generate or extract answers from provided context – these tasks directly measure performance in reader component of RAG systems, where model must comprehend retrieved documents and produce accurate answers;
- **Perplexity:** bonus metric that does not correlate with other scores and should not be used for direct model comparison (lower is better).

Table 5 presents the current scores of both the pretrained and continuously pretrained models as evaluated on the Open PL LLM Leaderboard in a 5-shot setting as of April 3, 2024.

The Bielik 7B v0.1 base model demonstrated strong performance in RAG-specific tasks. Among the base models, Bielik 7B v0.1 achieved the highest RAG Reader score (88.39), representing a notable improvement of approximately three percentage points over Mistral-7B-v0.1 (85.39). However, it is important to note that this specialized performance came with a trade-off: Bielik's overall average score across all tasks (29.38) was slightly lower than Mistral-7B-v0.1 (30.67). This reflected our base model's optimization focus on Polish language understanding and RAG capabilities, which resulted in particularly strong performance on context-based question answering tasks at the potential expense of broader task coverage.

Similarly, the instruction-tuned Bielik-7B-Instruct-v0.1 achieved exceptional RAG Reader performance (86.00) compared to Mistral-7B-Instruct-v0.1 (73.68), representing nearly a nine-percentage-point improvement, though Bielik maintained a higher overall score (39.28 vs 26.42) across all tasks in this comparison. In our subjective evaluations of conversational abilities, our models outperformed others that had higher average scores. The results presented in Table 5 were obtained without employing instruction templates for the instructional models, treating them instead

as base models. This approach may have skewed the results, as instructional models are specifically optimized to follow particular instructions.

Table 5

Detailed comparison among Bielik 7B v0.1 and other representative open-source models

Model	All tasks	RAG Reranking	RAG Reader	Perplexity
7B parameters models:				
berkeley-nest/Starling-LM-7B-alpha	**47.46**	**75.73**	82.86	1438.04
openchat/openchat-3.5-0106	47.32	74.71	83.60	1106.56
Nexusflow/Starling-LM-7B-beta	45.69	74.58	81.22	1161.54
openchat/openchat-3.5-1210	44.17	71.76	82.15	1923.83
teknium/OpenHermes-2.5-Mistral-7B	42.64	70.63	80.25	1463.00
mistralai/Mistral-7B-Instruct-v0.2	40.29	72.58	79.39	2088.08
Bielik-7B-Instruct-v0.1	39.28	61.89	**86.00**	277.92
internlm/internlm2-chat-7b	37.64	72.29	71.17	3892.50
internlm/internlm2-chat-7b-sft	36.97	73.22	69.96	4269.63
HuggingFaceH4/zephyr-7b-alpha	33.97	71.47	73.35	4464.45
HuggingFaceH4/zephyr-7b-beta	33.15	71.65	71.27	3613.14
szymonrucinski/Curie-7B-v1	26.72	55.58	85.19	389.17
mistralai/Mistral-7B-Instruct-v0.1	26.42	56.35	73.68	6909.94
meta-llama/Llama-2-7b-chat-hf	21.04	54.65	72.93	4018.74
Voicelab/trurl-2-7b	18.85	60.67	77.19	1098.88
Baseline (majority class)	0.00	53.36	–	–
Models with different sizes:				
upstage/SOLAR-10.7B-Instruct-v1.0 (10.7B)	46.07	**76.93**	82.86	789.58
Voicelab/trurl-2-13b-academic (13B)	29.45	68.19	79.88	733.91
Azurro/APT3-1B-Instruct-v1 (1B)	–13.80	52.11	12.23	739.09
7B parameters pretrained and continuously pretrained models:				
alpindale/Mistral-7B-v0.2-hf	**33.05**	60.23	85.21	932.60
internlm/internlm2-7b	33.03	**69.39**	73.63	5498.23
mistralai/Mistral-7B-v0.1	30.67	60.35	85.39	857.32
Bielik-7B-v0.1	29.38	62.13	**88.39**	123.31
internlm/internlm2-base-7b	20.68	52.39	69.85	3110.92
meta-llama/Llama-2-7b-hf	12.73	54.02	77.92	850.45
OPI-PG/Qra-7b	11.13	54.40	75.25	203.36

5.2. Polish MT-Bench

MT-bench [55] is a tool designed to test the ability of language models (LLMs) to conduct two-step conversations and follow instructions. It covers typical use cases and focuses on challenging questions to differentiate the capabilities of various models. Eight main categories of user queries were identified that were used to construct MT-bench:

- writing;
- role-playing;
- information extraction;

- reasoning;
- mathematics;
- coding;
- knowledge/hard sciences/stem;
- knowledge/humanities/social sciences.

For each category, two-step questions were manually developed.

The evaluation of the responses was performed by a metamodel; in the case of MT-Bench, this was the GPT-4 model. By using a metamodel, we could verify responses from the open-ended questions; e.g., write an article about hybrid cars. The model evaluated the content of the response, the quality of facts used, creativity, etc.

The Polish MT-Bench [21] was completely polonized. Each task was first machine-translated and then verified. Additionally, we introduced Polish accents; e.g., instead of describing a vacation in Hawaii, we suggested a location – Masuria. In our language version, many changes were introduced to transfer the test into Polish linguistic realities.

Table 6 presents the results of the Polish MT-Bench evaluation for the various language models. The table shows three key metrics: the Polish score (pl_score), the proportion of responses in Polish (responses_pl), and the average score. The Bielik 7B v0.1 model had a pl_score of 5.40, demonstrating competitive performance among the larger models.

Table 6

Polish MT-Bench results for various language models

Model	pl_score	responses_pl	Average Score
Mixtral-8x7b	**7.64**	1.00	**7.64**
Mistral-Nemo-Instruct-2407	7.37	1.00	7.37
openchat-3.5-0106-gemma	6.51	0.96	6.81
Meta-Llama-3.1-8B-Instruct	6.24	1.00	6.24
Starling-LM-7B-alpha	6.05	0.93	6.49
openchat-3.5-0106	6.03	0.94	6.39
Mistral-7B-Instruct-v0.3	5.75	0.98	5.82
Bielik-7B-Instruct-v0.1	5.40	0.89	6.08
dolphin-2.9.1-llama-3-8b	5.24	0.89	5.86
Polka-Mistral-7B-SFT	4.43	0.98	4.52
trurl-2-7b	2.75	0.99	2.76
Mistral-7B-Instruct-v0.2	2.05	0.31	6.56

Table 7 provides a more detailed breakdown of the Polish MT-Bench results, showing scores across eight different categories for each model. The Bielik 7B v0.1 model showed competitive performance in several categories – notably excelling in Reasoning (6.15) and Role-playing (7.83). These results demonstrated the model's versatility across various tasks despite its smaller size when compared to some of the top-performing models.

Table 7
Polish MT-Bench results for various language models by category

Model	Coding	Extraction	Humanities	Mathematics	Reasoning	Role-playing	Stem	Writing
Mixtral-8x7b	5.20	8.15	9.45	5.65	5.80	**8.95**	8.55	**9.35**
Mistral-Nemo-Instruct-2407	**5.85**	8.95	**9.50**	**6.70**	5.80	7.45	8.30	6.40
openchat-3.5-0106-gemma	5.35	6.90	8.80	4.55	5.40	7.97	8.47	7.05
Meta-Llama-3.1-8B-Instruct	4.60	**9.10**	8.82	5.30	2.50	5.60	6.30	7.70
Starling-LM-7B-alpha	4.75	7.35	8.50	4.15	3.90	6.90	**8.85**	7.55
openchat-3.5-0106	5.05	6.90	9.30	3.80	3.90	6.00	8.40	7.75
Mistral-7B-Instruct-v0.3	4.30	7.30	6.75	2.35	3.80	7.25	7.45	7.35
Bielik-7B-Instruct-v0.1	3.00	4.35	8.47	4.10	**6.15**	7.83	6.90	7.85
dolphin-2.9.1-llama-3-8b	4.60	6.15	8.80	4.80	3.30	7.40	6.35	5.50
Polka-Mistral-7B-SFT	2.95	5.25	5.60	2.95	2.45	4.90	6.80	5.25
trurl-2-7b	1.80	3.50	3.95	1.70	2.05	3.30	2.65	3.15
Mistral-7B-Instruct-v0.2	4.25	7.40	8.40	3.20	5.00	–	–	–

5.3. Bias, toxicity, and misinformation

Language models have been shown to reproduce and amplify biases present in the training data and can generate toxic or offensive content. Since our training data set contained a large proportion of data from the web, Bielik-7B-v0.1 may produce factually incorrect output and should not be relied upon for producing accurate information. Despite significant efforts to clean the training data, it is still possible for this model to generate lewd, false, biased, or otherwise offensive content.

6. Model quantization

In our work on the Bielik 7B v0.1 model, our primary objective was to create quantized versions that could be accessible to users with limited computational resources. This effort was driven by a vision to democratize advanced language models and make them available to those who do not have access to a powerful computing infrastructure. By optimizing our model for low-resource environments, we aimed to facilitate deployment on various devices, including edge devices (such as mobile phones and embedded systems).

To achieve this, we developed and delivered several quantized versions of Bielik 7B v0.1, including GGUF (GPT – Generated Unified Format)[1], HQQ (Half-Quadratic Quantization) [3], AWQ (Activation-aware Weight Quantization) [26], MLX (Apple MLX Framework) [16], EXL2 (ExLlamaV2)[2], GPTQ (Accurate Post-Training Quantization for Generative Pre-trained Transformers) [13], and IQ2_XXS (GGUF IQ)[3]. Each quantization technique offered different trade-offs in terms of performance, memory usage, and computational requirements, allowing for flexibility depending on the intended use case and hardware capabilities. The IQ2_XXS version in particular was specifically designed for edge devices, with a bit-per-weight quantization of 2.06 bpw; this provides an efficient solution for deployments on resource-constrained platforms such as mobile phones.

6.1. Calibration and evaluation of quantized models

In addition to the standard quantization process, we created calibrated versions of the imatrix (Importance Matrix) GGUF model. Calibration plays a crucial role in minimizing performance degradation, which is often a concern during quantization. To support this process, we developed a multilingual (Polish-English) calibration data set with a specific emphasis on the Polish language. This multilingual approach aimed to improve the model's generalization capabilities across the languages while ensuring high fidelity in its Polish-language outputs.

To assess the impact of the calibration, we conducted a thorough comparison between the uncalibrated and calibrated versions of the model for the Polish language.

[1] https://github.com/ggerganov/ggml
[2] https://github.com/turboderp/exllamav2
[3] https://github.com/ggerganov/llama.cpp

Our evaluation metrics focused on both the accuracy of the language understanding and the quality of the generated text. The results showed that the calibration process improved the model's performance – particularly in language-specific contexts where nuances and subtleties are crucial.

Table 8

Comparison of quantization results for Bielik 7B v0.1 model using imatrix: PPL – Perplexity; ΔPPL – change in perplexity; KLD – Kullback-Leibler Divergence; Mean Δp – mean change in correct token probability; RMS Δp – root mean square of change in token probabilities; same top p – percentage of instances where quantized model and FP16 model assign highest probability to same token

Quant.	imatrix	Size [GiB]	PPL	ΔPPL	KLD	Mean Δp	RMS Δp	Same top p [%]
FP16	–	13.49	3.9393	–	–	–	–	–
Q8_0	No	7.17	3.9422	0.0029	0.0010	-0.0070	0.9800	98.6890
Q8_0	Yes	7.17	3.9422	0.0029	0.0010	–0.0070	0.9800	98.6890
Q6_K	No	5.53	3.9450	0.0057	0.0051	-0.0420	2.1850	97.2410
Q6_K	**Yes**	5.53	**3.9406**	0.0013	0.0037	–0.0030	1.8490	97.6130
Q5_K_M	No	4.78	3.9520	0.0127	0.0106	–0.0680	3.1320	96.0510
Q5_K_M	**Yes**	4.78	**3.9473**	0.0080	0.0086	–0.0250	2.8320	96.4670
Q4_K_M	No	4.07	3.9876	0.0483	0.0286	–0.2690	5.1300	93.6550
Q4_K_M	**Yes**	4.07	**3.9727**	0.0333	0.0220	–0.1440	4.4880	94.4700
Q3_K_M	No	3.28	4.0915	0.1522	0.0826	–0.9160	8.6880	89.5780
Q3_K_M	**Yes**	3.28	**4.0458**	0.1065	0.0683	–0.3860	7.8390	90.6290
Q2_K	No	2.53	4.7045	0.7652	0.2852	–3.8050	16.3760	81.1100
Q2_K	**Yes**	2.53	**4.3522**	0.4128	0.1939	–1.8980	13.4190	84.5580

Across all of the quantization schemes examined (Q8_0, Q6_K, Q5_K_M, Q4_K_M, Q3_K_M, Q2_K) (see Table 8), those models quantized with imatrix consistently outperformed their counterparts without imatrix quantization. This was evident through multiple evaluation metrics, indicating that imatrix quantization effectively preserves model quality even at lower bit-widths. The KLD values were consistently lower for the imatrix-quantized models, indicating a closer alignment of the probability distributions between the quantized and the original FP16 models. The imatrix quantization results in the Mean Δp values were closer to zero, indicating less degradation in the model's ability to predict the correct token. At Q3_K_M, the Mean Δp improved from -0.9160 without imatrix to -0.3860 with imatrix. The advantages of imatrix quantization became more pronounced at lower bit-width quantization levels. The reduction in performance metrics such as PPL, KLD, Mean Δp, and RMS Δp was more significant when comparing the imatrix and non-imatrix models at the Q2_K and Q3_K_M levels, demonstrating imatrix's effectiveness in mitigating the adverse effects of aggressive quantization.

The application of imatrix quantization to the Polish language model has led to significant improvements in maintaining model quality across various quantization levels. These findings support the adoption of imatrix quantization as an effective technique for compressing language models without substantially compromising their performance.

7. Conclusion

In this paper, we introduced Bielik 7B v0.1 – a language model specifically trained for the Polish language. We demonstrated that it was possible to significantly enhance the linguistic capabilities of an already trained model by fine-tuning it on texts exclusively in that language. Without changing the tokenizer, we achieved a high quality of the responses generated by the model, which resembled texts written by native Polish speakers. Furthermore, the model performed well in various tasks, opening up intriguing possibilities for its further development.

Acknowledgements

We gratefully acknowledge Poland's high-performance Infrastructure PLGrid ACK Cyfronet AGH for providing computer facilities and support within Computational Grant No. PLG/2024/016951. The model could not have been created without the commitment and work of the entire SpeakLeash team, whose contributions were invaluable. Thanks to the hard work of many individuals, it was possible to gather a large amount of content in Polish and establish collaboration between the open-science SpeakLeash project and the HPC center: ACK Cyfronet AGH. Individuals who contributed to the creation of the model through their commitment to the open-science SpeakLeash project: Sebastian Kondracki, Szymon Mazurek, Maria Filipkowska, Paweł Kiszczak, Igor Ciuciura, Jacek Chwiła, Szymon Baczyński, Grzegorz Urbanowicz, Paweł Cyrta, Jan Maria Kowalski, Karol Jezierski, Kamil Nonckiewicz, Izabela Babis, Nina Babis, Waldemar Boszko, and many other wonderful researchers and enthusiasts of the AI world.

References

[1] Ainslie J., Lee-Thorp J., de Jong M., Zemlyanskiy Y., Lebron F., Sanghai S.: GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. In: H. Bouamor, J. Pino, K. Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, Association for Computational Linguistics, Singapore, 2023. doi: 10.18653/v1/2023.emnlp-main.298.

[2] Ansel J., Yang E., He H., Gimelshein N., Jain A., Voznesensky M., Bao B., *et al.*: PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*, ACM, 2024. doi: 10.1145/3620665.3640366.

[3] Badri H., Shaji A.: Half-Quadratic Quantization of Large Machine Learning Models, 2023. https://mobiusml.github.io/hqq_blog/.

[4] Bandarkar L., Liang D., Muller B., Artetxe M., Shukla S.N., Husa D., Goyal N., *et al.*: The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 749–775, Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 2024. doi: 10.18653/v1/2024.acl-long.44.

[5] Beeching E., Fourrier C., Habib N., Han S., Lambert N., Rajani N., Sanseviero O., Tunstall L., Wolf T.: Open LLM Leaderboard (2023-2024), https://huggingface.co/spaces/open-llm-leaderboard-old/open_llm_leaderboard, 2023.

[6] Beltagy I., Peters M.E., Cohan A.: Longformer: The Long-Document Transformer, *ArXiv preprint*, vol. abs/2004.05150, 2020. https://arxiv.org/abs/2004.05150.

[7] Chen L., Li S., Yan J., Wang H., Gunaratna K., Yadav V., Tang Z., *et al.*: AlpaGasus: Training a Better Alpaca with Fewer Data. In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, OpenReview.net, 2024. https://openreview.net/forum?id=FdVXgSJhvz.

[8] Child R., Gray S., Radford A., Sutskever I.: Generating Long Sequences with Sparse Transformers, *ArXiv preprint*, vol. abs/1904.10509, 2019. https://arxiv.org/abs/1904.10509.

[9] Dadas S.: Training Effective Neural Sentence Encoders from Automatically Mined Paraphrases. In: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 371–378, 2022. doi: 10.1109/SMC53654.2022.9945218.

[10] Dadas S., Perełkiewicz M., Poświata R.: Evaluation of Sentence Representations in Polish. In: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (eds.), *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 1674–1680, European Language Resources Association, Marseille, France, 2020. https://aclanthology.org/2020.lrec-1.207.

[11] Dao T.: FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, OpenReview.net, 2024. https://openreview.net/forum?id=mZn2Xyh9Ec.

[12] Dauphin Y.N., Fan A., Auli M., Grangier D.: Language Modeling with Gated Convolutional Networks. In: D. Precup, Y.W. Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research, vol. 70, pp. 933–941, PMLR, 2017. http://proceedings.mlr.press/v70/dauphin17a.html.

[13] Frantar E., Ashkboos S., Hoefler T., Alistarh D.: GPTQ: Accurate Post-training Compression for Generative Pretrained Transformers, *ArXiv preprint*, vol. abs/2210.17323, 2022. https://arxiv.org/abs/2210.17323.

[14] Gao L., Tow J., Abbasi B., Biderman S., Black S., DiPofi A., Foster C., *et al.*: A framework for few-shot language model evaluation, 2024. doi: 10.5281/zenodo.12608602.

[15] Granziol D., Zohren S., Roberts S.J.: Learning Rates as a Function of Batch Size: A Random Matrix Theory Approach to Neural Network Training, *J Mach Learn Res*, vol. 23, pp. 173:1–173:65, 2020. https://api.semanticscholar.org/CorpusID:226281826.

[16] Hannun A., Digani J., Katharopoulos A., Collobert R.: MLX: Efficient and flexible machine learning on Apple silicon, 2023. https://github.com/ml-explore.

[17] Ibrahim A., Thérien B., Gupta K., Richter M.L., Anthony Q., Lesort T., Belilovsky E., Rish I.: Simple and Scalable Strategies to Continually Pre-train Large Language Models, *ArXiv preprint*, vol. abs/2403.08763, 2024. https://arxiv.org/abs/2403.08763.

[18] Imamura K., Sumita E.: Extending the Subwording Model of Multilingual Pretrained Models for New Languages, *ArXiv preprint*, vol. abs/2211.15965, 2022. https://arxiv.org/abs/2211.15965.

[19] Jiang A.Q., Sablayrolles A., Mensch A., Bamford C., Chaplot D.S., de las Casas D., Bressand F., Lengyel G., Lample G., Saulnier L., Lavaud L.R., Lachaux M.A., Stock P., Scao T.L., Lavril T., Wang T., Lacroix T., Sayed W.E.: Mistral 7B, *ArXiv preprint*, vol. abs/2310.06825, 2023. https://arxiv.org/abs/2310.06825.

[20] Jiang Z., Gu J., Zhu H., Pan D.Z.: Pre-RMSNorm and Pre-CRMSNorm Transformers: Equivalent and Efficient Pre-LN Transformers. In: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. http://papers.nips.cc/paper_files/paper/2023/hash/8f1bacee31caf990a4f08d84f0ccb322-Abstract-Conference.html.

[21] Kinas R., Maria F., SpeakLeash Team, Cyfronet Team: MT-Bench PL, https://huggingface.co/spaces/speakleash/mt-bench-pl, 2024.

[22] King G., Zeng L.: Logistic Regression in Rare Events Data, *Political Analysis*, vol. 9(2), p. 137–163, 2001. doi: 10.1093/oxfordjournals.pan.a004868.

[23] Kocoń J., Miłkowski P., Zaśko-Zielińska M.: Multi-Level Sentiment Analysis of PolEmo 2.0: Extended Corpus of Multi-Domain Consumer Reviews. In: M. Bansal, A. Villavicencio (eds.), *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 980–991, Association for Computational Linguistics, Hong Kong, China, 2019. doi: 10.18653/v1/K19-1092.

[24] Kudo T., Richardson J.: SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In: E. Blanco, W. Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Association for Computational Linguistics, Brussels, Belgium, 2018. doi: 10.18653/v1/D18-2012.

[25] Li D., Chen Z., Cho E., Hao J., Liu X., Xing F., Guo C., Liu Y.: Overcoming Catastrophic Forgetting During Domain Adaptation of Seq2seq Language Generation. In: M. Carpuat, M.C. de Marneffe, I.V. Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5441–5454, Association for Computational Linguistics, Seattle, United States, 2022. doi: 10.18653/v1/2022.naacl-main.398.

[26] Lin J., Tang J., Tang H., Yang S., Chen W.M., Wang W.C., Xiao G., Dang X., Gan C., Han S.: AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration. In: *MLSys*, 2024.

[27] Loshchilov I., Hutter F.: Decoupled Weight Decay Regularization. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

[28] Marcinczuk M., Ptak M., Radziszewski A., Piasecki M.: Open dataset for development of Polish Question Answering systems. In: *Proceedings of the 6th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Wydawnictwo Poznanskie, Fundacja Uniwersytetu im. Adama Mickiewicza*, 2013.

[29] Mitra A., Khanpour H., Rosset C., Awadallah A.: Orca-Math: Unlocking the potential of SLMs in Grade School Math, 2024.

[30] National Information Processing Institute and Gdańsk University of Technology: Qra models, 2024. https://huggingface.co/OPI-PG.

[31] Ociepa K.: ALLaMo: A Simple, Hackable, and Fast Framework for Training Medium-Sized LLMs, https://github.com/chrisociepa/allamo, 2023.

[32] Ociepa K., Azurro Team: Introducing APT3-1B-Base: Polish Language Model, 2024. https://azurro.pl/apt3-1b-base-en. Accessed: 2024-09-30.

[33] Ogrodniczuk M., Kopeć M.: The Polish Summaries Corpus. In: N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pp. 3712–3715, European Language Resources Association (ELRA), Reykjavik, Iceland, 2014. http://www.lrec-conf.org/proceedings/lrec2014/pdf/1211_Paper.pdf.

[34] Okulska I., Stetsenko D., Kołos A., Karlińska A., Głąbińska K., Nowakowski A.: StyloMetrix: An Open-Source Multilingual Tool for Representing Stylometric Vectors, *ArXiv preprint*, vol. abs/2309.12810, 2023. https://arxiv.org/abs/2309.12810.

[35] Ostapenko O., Lesort T., Rodriguez P., Arefin M.R., Douillard A., Rish I., Charlin L.: Continual Learning with Foundation Models: An Empirical Study of Latent Replay. In: S. Chandar, R. Pascanu, D. Precup (eds.), *Proceedings of The 1st Conference on Lifelong Learning Agents*, Proceedings of Machine Learning Research, vol. 199, pp. 60–91, PMLR, 2022. https://proceedings.mlr.press/v199/ostapenko22a.html.

[36] Ptaszynski M., Pieciukiewicz A., Dybala P., Skrzek P., Soliwoda K., Fortuna M., Leliwa G., Wroczynski M.: Expert-Annotated Dataset to Study Cyberbullying in Polish Language, *Data*, vol. 9(1), p. 1, 2023. doi: 10.3390/data9010001.

[37] Rezaei-Dastjerdehei M.R., Mijani A.M., Fatemizadeh E.: Addressing Imbalance in Multi-Label Classification Using Weighted Cross Entropy Loss Function, *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, pp. 333–338, 2020. doi: 10.1109/icbme51989.2020.9319440.

[38] Rybak P., Mroczkowski R., Tracz J., Gawlik I.: KLEJ: Comprehensive Benchmark for Polish Language Understanding. In: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1191–1201, Association for Computational Linguistics, Online, 2020. doi: 10.18653/v1/2020.acl-main.111.

[39] Rybak P., Przybyła P., Ogrodniczuk M.: PolQA: Polish Question Answering Dataset. In: N. Calzolari, M.Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 12846–12855, ELRA and ICCL, Torino, Italia, 2024. https://aclanthology.org/2024.lrec-main.1125.

[40] Sennrich R., Haddow B., Birch A.: Neural Machine Translation of Rare Words with Subword Units. In: K. Erk, N.A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Association for Computational Linguistics, Berlin, Germany, 2016. doi: 10.18653/v1/P16-1162.

[41] Shazeer N.: GLU Variants Improve Transformer, *ArXiv preprint*, vol. abs/2002.05202, 2020. https://arxiv.org/abs/2002.05202.

[42] Shi Z., Yang A.X., Wu B., Aitchison L., Yilmaz E., Lipani A.: Instruction Tuning With Loss Over Instructions. In: A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J.M. Tomczak, C. Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10–15, 2024*, 2024. doi: 10.52202/079017-2210.

[43] Soboleva D., Al-Khateeb F., Myers R., Steeves J.R., Hestness J., Dey N.: SlimPajama: A 627B token cleaned and deduplicated version of RedPajama, https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. https://huggingface.co/datasets/cerebras/SlimPajama-627B.

[44] SpeakLeash Team: SpeakLeash a.k.a Spichlerz!, 2024. https://www.speakleash.org. Accessed: 2024-09-30.

[45] Su J., Ahmed M., Lu Y., Pan S., Bo W., Liu Y.: RoFormer: Enhanced transformer with Rotary Position Embedding, *Neurocomputing*, vol. 568, 127063, 2024. doi: 10.1016/j.neucom.2023.127063.

[46] Teknium: OpenHermes 2.5: An Open Dataset of Synthetic Data for Generalist LLM Assistants, https://huggingface.co/datasets/teknium/OpenHermes-2.5, 2023.

[47] Tuora R., Zwierzchowska A., Zawadzka-Paluektau N., Klamra C., Kobyliński Ł.: PoQuAD-The Polish Question Answering Dataset-Description and Analysis. In: *Proceedings of the 12th Knowledge Capture Conference 2023*, pp. 105–113, 2023. doi: 10.1145/3587259.3627548.

[48] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I.: Attention is All you Need. In: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. doi: 10.65215/pc26a033.

[49] Voicelab: TRURL 2 models), 2023. https://huggingface.co/Voicelab.

[50] Wang G., Cheng S., Zhan X., Li X., Song S., Liu Y.: OpenChat: Advancing Open-source Language Models with Mixed-Quality Data. In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7–11, 2024*, OpenReview.net, 2024. https://openreview.net/forum?id=AOJyfhWYHf.

[51] Wróbel K., SpeakLeash Team, Cyfronet Team: Open PL LLM Leaderboard, https://huggingface.co/spaces/speakleash/open_pl_llm_leaderboard, 2024.

[52] Wu Y., Du K., Wang X.J., Min F.: Misclassification-guided loss under the weighted cross-entropy loss framework, *Knowledge and Information Systems*, vol. 66, pp. 4685–4720, 2024. doi: 10.1007/s10115-024-02123-5.

[53] Xu H., Zhan X., Yin H., Qin H.: Discriminator-Weighted Offline Imitation Learning from Suboptimal Demonstrations. In: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, S. Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17–23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, vol. 162, pp. 24725–24742, PMLR, 2022. https://proceedings.mlr.press/v162/xu22l.html.

[54] Zhang P., Zeng G., Wang T., Lu W.: TinyLlama: An Open-Source Small Language Model, *ArXiv preprint*, vol. abs/2401.02385, 2024. doi: 10.48550/arXiv.2401.02385.

[55] Zheng L., Chiang W., Sheng Y., Zhuang S., Wu Z., Zhuang Y., Lin Z., *et al.*: Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10–16, 2023*, 2023. http://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

[56] Zhou C., Liu P., Xu P., Iyer S., Sun J., Mao Y., Ma X., *et al.*: LIMA: Less Is More for Alignment. In: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10–16, 2023*, 2023. http://papers.nips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

A. Appendix – Examples of tasks

A.1. polemo2

Task: Sentiment analysis of online consumer reviews across four domains (medicine, hotels, products, university) with four-class labeling (positive, negative, neutral, ambiguous).

Example:

Opinia: "Leczyłam się u niej parę lat i nic mi nie pomogła, a jak zmieniłam lekarza po krótkim czasie zobaczyłam już poprawę a w tej chwili jestem już bez leków 6 lat i jest wszystko dobrze. Dr Ciborska leczyła mnie na depresję a potem przez dr Kopystecką miała m rozpoznany zespół maniakalno depresyjny i odtąd zmianę leków i przede wszystkim wysłuchała mnie z zaangażowaniem a nie jak dr Ciborska aby mnie zbyć."

Określ sentyment podanej opinii. Możliwe odpowiedzi:

A – Neutralny; B – Negatywny; C – Pozytywny; D – Niejednoznaczny

Prawidłowa odpowiedź: B

A.2. klej-ner

Task: Named entity recognition in sentences containing single-type entities classified into six categories (no entity, place, person, organization, time, geographical name).

Example:

Zdanie: "Ulewne deszcze nawiedziły także Słupsk."

Pytanie: Jakiego rodzaju jest nazwana jednostka, jeżeli występuje w podanym zdaniu?

Możliwe odpowiedzi: A – Brak nazwanej jednostki; B – Nazwa miejsca; C – Nazwa osoby; D – Nazwa organizacji; E – Czas; F – Nazwa geograficzna

Prawidłowa odpowiedź: B

A.3. 8tags

Task: Topic classification of social media headlines into eight categories (film, history, food, medicine, motorization, work, sport, technology).

Example:

Tytuł: "Czy bateria zrobiona z 1000 cytryn jest w stanie uruchomić silnik?"

Pytanie: jaka kategoria najlepiej pasuje do podanego tytułu?

Możliwe odpowiedzi:

A – film; B – historia; C – jedzenie; D – medycyna; E – motoryzacja; F – praca; G – sport; H – technologie

Prawidłowa odpowiedź: E

A.4. belebele

Task: Machine-reading comprehension for question answering.

Example:

Fragment: "Atom może być uważany za jeden z fundamentalnych elementów budujących całą materię. To bardzo złożona jednostka, która składa się, według uproszczonego modelu Bohra, z centralnego jądra, wokół którego znajdują się elektrony, co nieco przypomina planety krążące wokół Słońca – patrz rysunek 1.1. W skład jądra wchodzą dwa typy cząsteczek: neutrony i protony. Pod względem ładunku elektrycznego protony są dodatnie, elektrony są ujemne, a neutrony nie mają żadnego ładunku."

Pytanie: "Jaki ładunek mają cząstki krążące wokół jądra?"

Możliwe odpowiedzi: A – Ładunek dodatni; B – Bez ładunku; C – Ładunek ujemny; D – Ładunek dodatni i ujemny

Prawidłowa odpowiedź: C

A.5. dyk

Task: Question answering based on human-annotated pairs from Wikipedia's "Did You Know" section.

Example:

Pytanie: "za co Iwanowi Tyszkiwiczowi ucięto dłoń?"

Sugerowana odpowiedź: "Tyszkiewicz był torturowany – wyrwano mu język za "bluźnierstwo przeciw Bogu," a za rzucenie krucyfiksu na ziemię ucięto mu dłoń i nogę."

Czy sugerowana odpowiedź na zadane pytanie jest poprawna? Możliwe opcje: A – brakuje sugerowanej odpowiedzi; B – nie, sugerowana odpowiedź nie jest poprawna; C – tak, sugerowana odpowiedź jest poprawna; D – brakuje pytania

Prawidłowa opcja: C

A.6. ppc

Task: Text-similarity assessment using manually labeled sentence pairs (exact paraphrases, close paraphrases, non-paraphrases).

Example:

Zdanie A: "Piasek nad Chinami."

Zdanie B: "Burza piaskowa w Chinach."

Pytanie: jaka jest zależność między zdaniami A i B?

Możliwe odpowiedzi: A – wszystkie odpowiedzi poprawne; B – znaczą dokładnie to samo; C – mają podobne znaczenie; D – mają różne znaczenie

Prawidłowa odpowiedź: C

A.7. psc

Task: Summarization of news articles.

Example:

Fragment 1: "Zwykle zaczyna się od siąkających nosami kilku osób. Jednak choroba postępuje lawinowo. Wirus grypy przenosi się drogą kropelkową – podczas rozmowy, kaszlu i kichania. Jedna zagrypiona osoba, która pojawi się w towarzystwie, może zakazić wielu ludzi.Lekarz dyżurny kraju Michał Sobolewski uspokaja: w Polsce jeszcze nie ma epidemii grypy. Wybuchnie, kiedy będzie dużo źródeł zakażenia."

Fragment 2: "W niedzielę przychodnie w Warszwie zalała fala pacjentów z objawami grypy. Lekarz dyżurny kraju uspokaja, że w Polsce nie ma jeszcze epidemii grypy. Podkreśla też, że Polacy lekceważą profilaktyczne szczepienia przeciwko tej chorobie, a tylko one zapobiegają rozprzestrzenianiu się schorzeń zakaźnych. W Europie epidemia grypy dociera do kolejnych państw. Odnotowano już przypadki śmiertelne."

Pytanie: jaka jest zależność między fragmentami 1 i 2?

Możliwe odpowiedzi: A – wszystkie odpowiedzi poprawne; B – dotyczą tego samego artykułu; C – dotyczą różnych artykułów; D – brak poprawnej odpowiedzi

Prawidłowa odpowiedź: B

A.8. cbd

Task: Text classification for cyberbullying and hate-speech detection.

Example:

Wypowiedź: "Ty wiesz lepiej. Ja wiem, że nawet wiceprezydentem nie będziesz na 100%"

Pytanie: Jaka kategoria najlepiej pasuje do podanej wypowiedzi?

Możliwe odpowiedzi: A – nieszkodliwa; B – szyderstwo; C – obelga; D – insynuacja; E – groźba; F – molestowanie

Prawidłowa odpowiedź: B

A.9. polqa

Task: Open-domain question answering from the "Jeden z dziesięciu" TV show, with and without context (abstractive QA/RAG).

Example:

Kontekst: Przymiotnik. Przymiotniki, podobnie jak w języku polskim, odmieniały się przez liczby, rodzaje i przypadki. Wyraz określający następował zawsze po wyrazie określanym, tak jak w innych językach semickich, np. "ezzūtu šārū" „porywiste wiatry", dosłownie „wiatry porywiste"

Pytanie: Czy przymiotniki odmienia się przez przypadki?

Czy kontekst jest relewantny dla pytania?

Odpowiedz krótko "Tak" lub "Nie". Prawidłowa odpowiedź: Tak

Kontekst: Alibi (łac. gdzie indziej) – dowód w postępowaniu karnym na okoliczność, że podejrzany albo oskarżony znajdował się w miejscu innym niż miejsce popełnienia zarzucanego mu przestępstwa.

Pytanie: Jak z łaciny nazywa się dowód sądowy polegający na wykazaniu, że osoba oskarżona nie przebywała na miejscu przestępstwa w chwili gdy je popełniono?

Prawidłowa odpowiedź: alibi

A.10. poquad

Task: Context-based extractive question answering (QA/RAG). **Example:**

Tytuł: Miszna

Kontekst: Pisma rabiniczne – w tym Miszna – stanowią kompilację poglądów różnych rabinów na określony temat. Zgodnie z wierzeniami judaizmu Mojżesz otrzymał od Boga całą Torę, ale w dwóch częściach: jedną część w formie pisanej, a drugą część w formie ustnej. Miszna – jako Tora ustna – była traktowana nie tylko jako uzupełnienie Tory spisanej, ale również jako jej interpretacja i wyjaśnienie w konkretnych sytuacjach życiowych. Tym samym Miszna stanowiąca kodeks Prawa religijnego zaczęła równocześnie służyć za jego ustnie przekazywany podręcznik.

Pytanie: Czym są pisma rabiniczne?

Prawidłowa odpowiedź (krótki cytat z Kontekstu): kompilację poglądów różnych rabinów na określony temat

B. Evaluation reproducibility

To reproduce our results, you need to clone the repository:

```
git clone https://github.com/speakleash/lm-evaluation-harness.git -b
    polish3
cd lm-evaluation-harness
pip install -e .
```

and run benchmark for 0-shot and 5-shot:

```
lm_eval --model hf --model_args pretrained=speakleash/Bielik-7B-
    Instruct-v0.1 --tasks polish_generate --num_fewshot 0 --
    output_path results/ --log_samples
```

```
lm_eval --model hf --model_args pretrained=speakleash/Bielik-7B-
    Instruct-v0.1 --tasks polish_mc --num_fewshot 0 --output_path
    results/ --log_samples

lm_eval --model hf --model_args pretrained=speakleash/Bielik-7B-
    Instruct-v0.1 --tasks polish_generate_few --num_fewshot 5 --
    output_path results/ --log_samples

lm_eval --model hf --model_args pretrained=speakleash/Bielik-7B-
    Instruct-v0.1 --tasks polish_mc --num_fewshot 5 --output_path
    results/ --log_samples
```

Affiliations

Krzysztof Ociepa
SpeakLeash, Azurro, krzysztof.ociepa@bielik.ai

Łukasz Flis
SpeakLeash, ACK Cyfronet AGH, lukasz.flis@cyfronet.pl

Krzysztof Wróbel
SpeakLeash, Jagiellonian University, Enelpol, krzysztof.pawel.wrobel@uj.edu.pl

Adrian Gwoździej
SpeakLeash, ACK Cyfronet AGH, adrian.gwozdziej@cyfronet.pl

Remigiusz Kinas
SpeakLeash, remigiusz.kinas@bielik.ai

Received: 16.10.2025
Revised: 03.11.2025
Accepted: 08.11.2025