INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2025, VOL. 71, NO. 3, PP. 1–10
Manuscript received February 7, 2025; revised June, 2025. doi: 10.24425/ijet.2025.153615

Automatic questions generation based on keywords using language models

Tomasz Gniazdowski, Marek Bazan, and Maciej E. Marchwiany

***Abstract*—In this paper, we presented a novel method for question generation, one of the most impactful NLP tasks in contacts of user interfaces, chatbots and intelligent assistants with a user. Our method outperforms commonly used methods in terms of quality and speed of question generation. Additionally, we benchmarked the most used methods of question generation that are based on the usage of Large Language Models in a few-shot approach as well as finetuned to that task. Our work is done for the Polish language, it has one of the most challenging and complex grammars, which makes the task even more difficult.**

***Keywords*—question generation; finetuned Large Language Models; generative AI; Bart; PLT5; Llama; Bielik**

I. Introduction

QUESTION generation is a key component of user support automatization, robotic assistants, and chatbots. Despite the wide range of applications, the task has not yet been fully addressed in the literature and the artificial intelligence community. The question generation as a natural language processing (NLP) task may be approached in two ways. The first is a process of building questions for information extraction from a long text. The second relies on creating a question based on keywords or descriptions of a text.

In this paper, we focus on the latter understanding of the question generation task, i.e. given a description of a datafield, a name of the data field or some other short abbreviation only to construct the most probable question about the data field. Usually, such data fields can be found in graphical user interfaces, and the generated questions are used to interrogate users about the data fields in a sensible way.

The goal of this paper is to investigate the existing State Of The Art open-source Large Language Models (LLMs), such as: Llama with zero/few-shot prompting and finetuning procedures and Bielik - Polish Large Language Model in a zero/few-shot approach, and compare them with our original method. Moreover, knowledge graphs are also used in the given task [1], but this approach does not increase the efficiency of question generation.

In this paper we proposed a novel Rule-Based algorithm enhanced with encoder-decoder Language Model, such as T5 or BART, for question generation. All investigated methods are based on the name of the data fields, keywords, or a text description.

Our results are significant for chatbot development. The presented algorithm may be used for the question construction for mandatory data fields that were missed by LLMs or were absent in the bulk text provided by the user. The Rule-Based method has several advantages over LLMs and knowledge graphs. The first one is the certainty and flexibility of Rule-Based methods. The content of the generated question is determined by the proposed rules and it will always be the same - no inaccuracies or ambiguities will be introduced (e.g. change of verb tense, change of verb mood, change of modal verb, etc.). The second advantage is flexibility and ease of adding new rules or topics' extension - no training of any models is required, which is both time and cost efficient. The third advantage is the speed of the proposed method (Rule-Based methods are the fastest solutions) and resource consumption (these methods do not require large hardware resources). Our conclusion has been tested in real-life examples. The solution was implemented in production in the JT Weston's NEULA software. Furthermore, we observed that the listed advantages of our method are crucial for the success of commercial implementation of question generation methods. Our experience shows that it is especially important for automated systems, where speed and reliability must be on the highest level. In that case, LLM-based methods (even expanded by knowledge graphs) are not good enough.

The complexity of the problem arises from its generality. The solution algorithm should be easy to extend to work on fields concerning various subjects. In addition, the solution should be configurable in the same way as the forms themselves. Moreover, the form of questions should be configurable depending on the needs of the users.

To our knowledge, such a task has not been considered in the literature before for the Polish language. Although many question generation algorithms have been investigated, they are dedicated for English only. One of the examples where some rules to create questions starting from `what` or `who` are

This work was financed with European Union funds from the Smart Growth Operational Programme 2014-2020, Measure 1.1.: R&D projects of enterprises. Sub-measure 1.1.1.: Industrial research and development carried out by enterprises

T. Gniazdowski is with JT Weston sp. z o.o. Warszawa, Poland, (tomasz.gniazdowski@jtweston.pl)

M. Bazan is with Wrocaw University of Science and Technology, Faculty of Information and Communication Technology, Department of Computer Engineering, Poland (e-mail: marek.bazan@pwr.edu.pl).

M. E. Marchwiany is with JT Weston sp. z o.o. Warszawa, Poland and also with Interdisciplinary Centre for Mathematical and Computational Modelling (ICM), University of Warsaw, Poland (maciej.marchwiany@jtweston.pl).

© The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0, https://creativecommons.org/licenses/by/4.0/), which permits use, distribution, and reproduction in any medium, provided that the Article is properly cited.

presented in [2] in the context of semantic role labeling task and name entity recognition.

The problem's difficulty increases because the main investigated language is Polish. It should be emphasized that the Polish language is characterized by the complicated flexion system, which covers both nouns, adjectives, and verbs. This fact generates the need for one additional step along the question formation: grammar correction.

To provide the naturalness of generated questions, the proposed algorithm has to be creative and, at the same time, it cannot hallucinate. The method proposed in this paper meets the requirements. We showed its performance on a real dataset that we published for research purposes [3].

The remainder of the paper is organized as follows. The second section contains related work found in the literature, the third chapter is a description of the investigated methods (LLMs and prompting techniques) followed by the definition of our novel algorithm. In section 4, we presented an evaluation of models with the dataset and experiment descriptions. The article concludes with a summary. In the appendix, we present sets of hyperparameters, comparison of the outputs of different methods and hallucination evaluation.

II. Related Work

Several surveys on question generation problems were published recently. The most common approach is keyword extraction and, based on them, question generation. These solutions use models: LSTMs, bi-LSTMs and sequence-to-sequence, such as those based on attention and softmax pointer mechanisms [4]–[6].

Before transformers, several approaches for question generation were developed. Most of them include part-of-speech labeling and the inversion of the resulting tagging. For instance [7] employed semantic role labeling, slot filling, and Named Entity Recognition with predefined question templates to achieve questions that are not just rearranged sentences.

The first generative deep encoder-decoder neural network model for key phrase extraction was proposed in [8]. The mentioned model was trained for abstractive-extractive keyphrase extraction. The encoder-decoder models using a recurrent neural network (RNN) were first introduced in [9] and then enhanced in terms of quality in [4] with the attention mechanism.

One of the first papers on question generation from keywords is [10]. In that paper, RNN (with LSTM units) was proposed as a decoder model to create questions. The method proposed therein is a baseline for future research. The first to improve the results was [11].

Recently, knowledge graphs were used to generate diversified questions in [12].

Automatic question generation for dialogues carried out by an information chatbot was presented in [13]. The authors used the T5 model [14] where the input consists of keywords retrieved from the general query and the context is retrieved from the predefined answer for a specific query. Then, the diverse generated questions may be used to train intent detection models used in chatbots. The SBERT model was used to measure the similarity of the query and each of the top five generated questions.

An approach similar to the latter one was presented in [15]. However, only keywords were used as input to a Machine Transformation Model to generate a well-formed query. The application of this research is broader since it may also serve as a question generator for question-answering tasks.

The question generation problem was also resolved in the context of the improvement of the recruitment process by chatbots [16]. In this work, a chatbot was used to perform a screening of a candidate using automatically generated questions based on the skills reported on the resume of a candidate. The answers of a candidate were compared with the automatically generated answers for the questions of the corresponding passages that were matched to the generated queries.

Keyword extraction as a tool for question generation mechanism to summarize text passages by using questions in the form of fill-in-the-blank questions and multiple choice questions generation with a distractor set has been proposed in [17]. In this publication, the Latent Semantic Analysis method was used for keyword extraction and was compared with several other methods to extract keywords from text.

One of the question generation applications is the evaluation of the knowledge acquired by reading various educational materials. Such an application was considered in [18] where such types of autogenerated questions as: fill-up with double blank questions, True or False questions, match the following and short answer questions to check the knowledge of the learners were considered to provide the diversity of testing. The type of question to be generated was described by four rules. The method was also based on Latent Semantic Analysis making use of Name Entity Recognition and Part of Speech Recognition. This shows that rule-based approaches are still being developed in practical applications.

Recently, question generation has been considered for constructing the dataset containing questions generated in passages in a set of documents together with the corresponding answers [19]. This data set may be used to assess chatbots using LLMs regarding hallucinations in answering questions concerning medical processes in hospitals. In the dataset, questions with no answers based on documents were also generated, and the task of the language models was to detect such a situation instead of thinking up hallucinated answers.

However, general purpose LLMs cannot be used successfully for question generation. Finetuning for the task is requested to improve correctness and eliminate hallucinations. In the task of finetuning LLMs, the two main methods are: *Adapter* [20] and *Lora* [21].

In the *Adapter* method, the weights of the original model are frozen and trainable layers are added in given places in the finetuning model. In the original version of the method, the added layers were a simple "bottleneck" architecture consisting of: a feedforward layer (down-project), a nonlinearity, and a feedforward layer (up-project). These parameters were added after the multihead attention block and after the feedforward layer in the Transformer's encoder architecture.

The *Lora* method also involves adding trainable parameters to the frozen model's parameters. These trainable parameters (e.g. matrix ΔW) are added to an original, frozen model's parameters (e.g. weights W). Moreover, ΔW parameters are described using Low-Rank Approximation to reduce memory usage. Finetuned model's output for original layer W for the input data x is calculated as $(W + \Delta W)x$.

The most common usage of the LLMs are: zero-shot and few-shot [22] prompting. The same techniques can be used for a question generation task. Zero-shot prompting is based on the generative potential of the model and the knowledge on which it has been trained on. The zero-shot prompt is simply a task description. In few-shot prompting, the basic prompt is expanded with a few examples of expected outcomes.

Another challenge in the question generation task is model evaluation. In general, as emphasized in [23], two approaches to evaluate the quality of automatically generated questions dominate in the literature. The first approach is to generate questions to answers for known datasets or learning test books. Then, the exact match defines precision, recall and finally - F1 score. In addition, in the same spirit, the semantic correctness of the generated questions should be checked in as many affirmative sentences as possible. The second approach is based on a comparison of a generated question with a ground truth question generated by the user using metrics such as n-gram-based metrics: METEOR [24], BLEU [25], ROUGE [26] and semantic similarity-based metrics: BERTScore [27] and BLEURT [28].

We are not aware of any references in the literature specifically addressing the question generation task for the Polish language.

An important aspect of the use of LLMs in our method is their trustworthiness and safety. Recently, the subject of the trustworthiness of LLMs was undertaken in [29] or in [30]. While developing business or mission-critical applications with the methodology for building questions presented in this paper, methods to defend against attacks with malicious keywords or field descriptions [30] may be used. An example of the methods that can be used to prevent uncontrolled input and output to and from the LLM was presented recently in [31] in the context of a critical application.

If we wanted to omit the step of defence against malicious input and output, then the trustworthiness of the questions is inherited from the LLM used for generating a question or for model used for the grammar correction. In the paper [29] Llama 2 7B is one of the models performing well on such trustworthiness dimensions as trustfulness, safety, fairness, robustness and privacy. By performing well, we understand the fact that it is amongst the eight best models investigated in 50% or more of tests performed in each dimension – see Figure 1 therein. Since we use Llama 3 8B we can expect even better results.

III. Proposed methods

A. Rule-based method with custom grammar correction model

Together with JT Weston's business department, a set of rules was developed responsible for creating the initial version of a question based on the input keywords. The input for the developed Rule-Based algorithm was the field type and its description. Then, from the field description, the question object was extracted together with the dependent words by the Spacy framework [32] (root token with its dependencies in the Spacy nomenclature). In Spacy non-monotonic arc-eager transition-system [33] and pseudo-projective dependency transformation [34] algorithms are used in this task. Then, at the beginning of the extracted text, a question pronoun was added. The pronoun depended on the type of field and the question's object (e.g. "choose from the list" if the field type was a selection list). At the end of the created question, additional info was appended (if necessary). The last step of creation was an improvement of Polish grammar by our language model.

It needs to be noted that for the Rule-Based algorithm, it was necessary to prepare specific situations detection that were not handled by the method. For example, detection and correction of a word form when the replacement of the form by the grammar corrector was redundant (for example, from the singular to the plural of a word in the same case, e.g. "Ala has a cat." $\rightarrow$ "Ala has cats."). This case was handled using the sentence morphological analysis framework: Morfeusz 2 [35].

1) Grammar correction model: The Bart [36] pretrained model for the Polish language [37] was chosen as the model for the grammar improver of the text. The choice of this model was dictated by the fact that the Bart model was trained for correcting errors in input texts. Our task was similar to the original one, which made it possible to obtain a smaller loss function value for the validation dataset (in comparison to the PLT5 [38] model, which was pretrained on different tasks).

2) Dataset for grammar corrector: For the Polish Bart model finetuning in the task of improving Polish grammar, a database has been created. Data preparation was carried out in a similar way to the solution proposed for the English language [39]. Declarative sentences and questions containing from three to fourteen words were extracted from two text corpora for the Polish language: OpenSubtitles 2018 [40] and ParaCrawl 5 [41]. Then, all the texts were cleaned: punctuation marks were removed (only full stops and question marks were left). Then, all the texts were shuffled and subsampled. In this way $3 * 10^5$ announcement sentences and $3 * 10^5$ questions were obtained.

In the dataset, random nouns, verbs, adjectives and adverbs were reduced to their lemmas with given probabilities. Another perturbation was the removal of random auxiliary verbs from the input sentences. The probabilities values were selected to obtain the data as balanced as possible. The probabilities are presented in Table IV. Polish Verb Conjugator [42] was used for improving our model verb conjugation. Random verbs (in lemma forms) changed their forms to different, random ones. About 2% of all the texts in our dataset do not require grammar correction. This was intended to teach the model, that not all the texts should be modified. Finally, the dataset had more than $4 * 10^5$ texts divided into training and validation sets (in ratio $80 : 20$ respectively).

B. One-step method (Polish dataset, PLT5 and Bart)

1) Models: The second proposed solution to the problem was a one-step method using language models: T5 and Bart pretrained on the Polish language. These models were finetuned on our dataset containing pairs (keywords, output sentence) in the Polish language.

2) Dataset: The dataset of pairs (keywords, output sentences) was created using two corpora for the Polish language: Polish OpenSubtitles 2018 and ParaCrawl 5. Declarative, imperative and question sentences containing between three to fourteen words were extracted from these corpora. The data was cleaned - hashtags, mentions, hyperlinks and punctuation marks were removed (full stops, question marks and exclamation marks were left). Then all the texts were shuffled and sampled. From the dataset containing $\approx 5 * 10^5$ sentences keywords were extracted using three methods: KeyBERT [43], Multi Rake [44] and TextRank [45].

Pairs with a maximum difference between the number of keywords and the length of an original sentence equal to 5 were left. These operations resulted in a dataset of $\approx 1.7 * 10^5$ texts.

Since, in the dataset, the keywords always had the same form as the words in the output sentences, errors were added to the keywords. To obtain differences in keywords and outputs, random keywords were selected and reduced to the subform of the lemma (probability values are presented in Table III). The task was performed using the Spacy framework.

C. One-step method (Polish and English dataset, LLMs)

1) Models: Another approach to the problem was to finetune Large Language Model - Llama 3 Instruct [46] (the 8 billion parameter version) for creating questions based on given keywords. Again, the created dataset of pairs (keywords, output sentence) was used. Llama 3 has been finetuned with Lora and Adapter 2 [47] methods with LitGPT [48] framework.

The finetuning method that obtained the smallest loss function value on the Polish dataset was also used in finetuning the model on a dataset in English. Lora method was used in this task.

2) Datasets: The dataset from the previous section (III-B2) was used to finetune the models on Polish texts.

The dataset in English was prepared in a similar way. The sentences from five datasets (Aclmdb [49], Kaggle QA [50], Simple QA [51], Squad 2 [52], Tweet QA [53] and imperative sentences from Sentence Classification project [54]) were extracted and processed. The dataset was created analogously to III-B2. The final dataset consisted of $\approx 5 * 10^4$ announcing sentences, $\approx 6 * 10^4$ questions and $\approx 6 * 10^3$ imperative sentences.

For this method, when inferencing, the text data (in Polish) was translated into English and then the model response was translated back into Polish with Google Translator API (with the usage of Deep Translator package [55]).

During inference, a question pronoun was added to the form field description at its beginning and additional information was added at its end (if necessary) for one-step models - similar to the Rule-Based method, but here, these rules were much more truncated. The purpose of this was to force a specific form of the final sentence.

The aim of the one-step methods using smaller (Bart, PLT5) ($139 * 10^6$ and $275 * 10^6$ parameters) and Large (Llama) ($8 * 10^9$ parameters) Language Models was to produce a correct sentence in Polish (or English) and to improve its naturalness. The use of large models in sentences generation of correct questions and imperative sentences aimed to increase the naturalness of the utterance, which may have been lacking in a method based on complex rules and a grammar corrector model.

D. Polish Large Language Model prompting

In addition, the Polish LLM - Bielik 01 Instruct model [56] (with 7 billion parameters) was prompted to perform the given task. Two basic prompting methods were used: zero-shot and few-shot. The content of the used prompts is included in Appendix.

The flowchart of all the proposed methods is shown in Figure 1.

IV. Experiments

As part of the experiments, all proposed methods for generating correct sentences based on given keywords were evaluated on a real-world dataset provided by JT Weston. All experiments were performed on a computer running on Ubuntu 20.04 with a NVIDIA RTX 4090 GPU with 24 GB of memory.

A. Test Dataset

The test dataset comes from JT Weston's internal system applications. It contains 183 pairs from four different processes (keywords - field description with given question pronoun and additional info, output sentence - question, imperative or declarative sentence) in Polish language. The data annotation process was carried out by independent annotators in collaboration with JT Weston's business department.

B. Training configurations

1) Bart (grammar correction): Pretrained on the Polish language, the Bart Base model was finetuned. A grid search of hyperparameters for the learning rate and weight decay values has been performed with the learning rate $\in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. AdamW [57] optimizer was used with a weight decay equal to 10^{-7} and 10^{-6}. The best result on the validation set was obtained for the model trained for 9 epochs.

2) Bart and PLT5 (sentence generation based on keywords): For both models (Bart and PLT5 in base versions), a grid search of hyperparameters for the learning rate and weight decay values has been performed with learning rate $\in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. We used the AdamW optimizer with a weight decay equal to 10^{-7} and 10^{-6}. The best results were obtained for both models trained for 13 epochs.

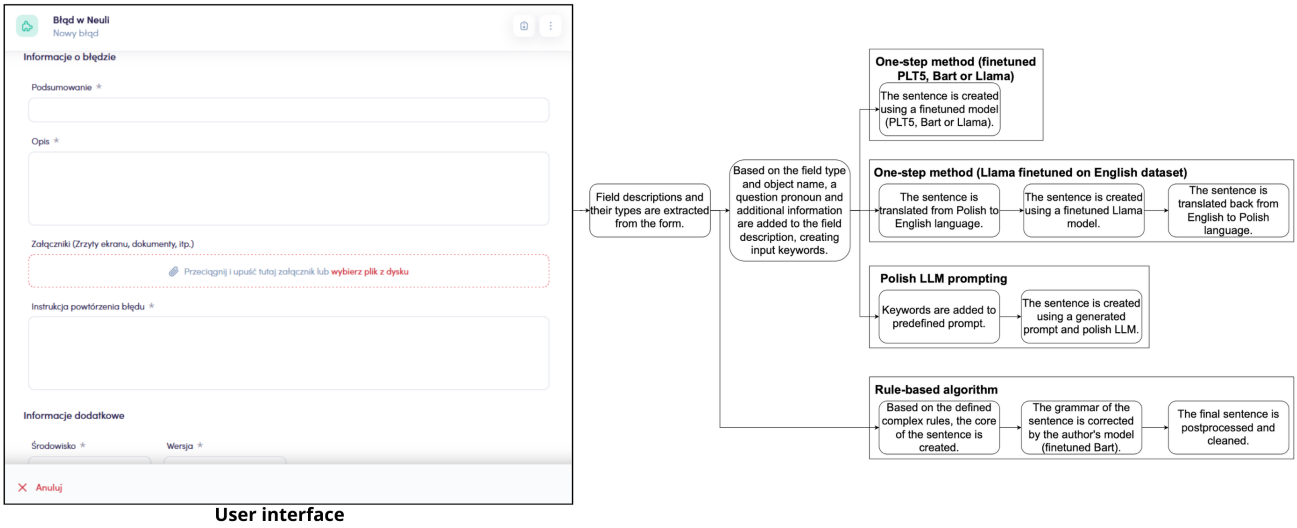

Figure 1. Diagram of the proposed methods for automatic creation of questions based on given keywords (during inference).

3) Llama (sentence generation based on keywords): For all finetuning methods, same hyperparameter set was used: AdamW optimizer with an initial learning rate equal to 10^{-3} and a weight decay equal to 10^{-2}. Moreover, we used linear warm-up followed by cosine annealing [58] as a learning rate scheduler. All the models were finetuned for 5 epochs.

C. Evaluation

Five metrics were used to evaluate the models in the text generation task: ROUGE [26], BLEU [25], BERTScore [27], BLEURT [28] and METEOR [24]. The hyperparameter values of the presented models' response decoding methods are presented in Table II in the Appendix. An evaluation of hallucination detection of the tested methods (Bart-greedy, Llama (Lora)-topk, Llama (Lora EN)-topk+topp, PL LLM prompting-few shot and Rule-Based-with GC) was also performed. The annotator received 35 random samples - model input texts and outputs generated by all five methods. The anotator's task was to determine whether the generated text contained hallucinations. The ratio of the number of texts with hallucinations to these 35 random texts is presented in Table V.

D. Results

The metrics values obtained for all the experiments performed are presented in Table I. The metric values are also visualized in Figure 2. The example outputs (in Polish) of the best-performing algorithms are presented in Appendix E (Tables VI and VII). During inference of evaluating models, we used various decoding strategies: *greedy, beam* search, *topk, topp* sampling, which are thoroughly described in section *IIIH* in [59]. Used hyperparameters are listed in Table II in the Appendix.

Rule-Based models outperform all other tested methods in all metrics. The method achieved ROUGE-1, ROUGE-2 and ROUGE-L scores of 0.990, 0.975, and 0.990, respectively, with grammar correction and 0.960, 0.902, and 0.961 without it. The best language model (Bart-greedy) achieved scores of 0.926, 0.810, and 0.924. A similar pattern is observed across the rest of the metrics. It is clear that grammar correction improves the model's performance for all investigated metrics. Additionally, in Table I some patterns for language models can be found. The best decoding strategies are: *greedy* (for Bart), *topp* (for PLT5), *topk* (for Llama) and *topk+topp* (for Llama Lora EN) (*topk* and *topp* decoding methods combined). However, Llama models have variability for different finetuning and decoding strategies. As expected, few-shot prompting has better results than zero-shot prompting. The best performing model is Bart with *greedy* decoding. Llama finetuned with the Lora method on the English dataset has the worst results, which are significantly lower than the other models. The exact reason for this behavior is described later in the article.

The results obtained for the detection of hallucinations (presented in Table V) are consistent with those obtained during the evalution of the model (Table I). Models pretrained on Polish language (Bielik - prompted using the few-shot method and Bart - fine-tuned on our dataset) generated the least texts containing hallucinations. Models pretrained on English language - Llama, generated the most texts containing hallucinations. The Rule-Based method, of course, did not generated hallucinations.

The Rule-Based method obtained the lowest average inference time values. This is, of course, due to the fact that no language models were used in this method itself (only language processing and analysis tools such as Spacy and Morfeusz 2 were used). The mean inference time of the Rule-Based algorithm together with the grammar correction model was less than 0.1 seconds per sentence, which allows the proposed method to be used in real time (which was one of the business requirements of the method). The inference time of the PLT5 model was about 0.1 seconds and that of the Bart model was about 0.05 seconds. This is of course related to the size of both models - the Bart model has fewer

Table I
METRIC VALUES OBTAINED IN ALL EXPERIMENTS. IN THE TABLE, THE HIGHEST METRIC VALUE FOR ALL THE METHODS IS MARKED IN BOLD. *MEAN INFERENCE TIME PER SENTENCE.

method		ROUGE-1	ROUGE-2	ROUGE-L	BLEU	BERTScore	BLEURT	METEOR	Mean time*
PLT5	greedy	0.910	0.800	0.911	0.745	0.979	0.799	0.904	0.1000 s
	beam	0.808	0.636	0.808	0.491	0.942	0.565	0.837	0.1413 s
	topk	0.910	0.796	0.910	0.741	0.979	0.800	0.903	0.0996 s
	topp	0.912	0.802	0.912	0.754	0.979	0.798	0.906	0.1008 s
Bart	greedy	0.926	0.810	0.924	0.780	0.981	0.807	0.924	0.0425 s
	beam	0.853	0.674	0.850	0.626	0.956	0.677	0.868	0.0538 s
	topk	0.926	0.817	0.923	0.777	0.979	0.786	0.919	0.0411 s
	topp	0.927	0.813	0.923	0.770	0.980	0.793	0.921	0.0416 s
Llama (Lora)	topk	0.914	0.790	0.913	0.713	0.976	0.762	0.921	0.8501 s
	topk+topp	0.914	0.788	0.912	0.706	0.975	0.759	0.921	0.8495 s
Llama (Adapter 2)	topk	0.908	0.820	0.906	0.740	0.973	0.737	0.914	1.3242 s
	topk+topp	0.892	0.804	0.889	0.721	0.968	0.713	0.900	1.3503 s
Llama (Lora, EN)	topk	0.567	0.285	0.560	0.257	0.869	0.281	0.531	1.7212 s
	topk+topp	0.569	0.287	0.561	0.256	0.868	0.279	0.535	1.7298 s
PL LLM prompting	zero shot	0.883	0.574	0.797	0.499	0.927	0.699	0.839	0.4602 s
	few shot	0.850	0.736	0.846	0.654	0.949	0.719	0.833	0.4600 s
Rule-Based	without GC	0.960	0.902	0.961	0.873	0.992	0.947	0.953	**0.0171 s**
	with GC	**0.990**	**0.975**	**0.990**	**0.964**	**0.998**	**0.981**	**0.986**	0.0717 s

parameters. The finetuned Llama models obtained the largest inference times. This is related to the number of parameters of these models ($8 * 10^9$). The Lora method obtained shorter inference times because, during the inference of a model finetuned by this method, the input data do not pass through the additional parameters (in contrast to the Adapter method). Long translation times in the method based on Llama finetuned on English texts exclude this method. The Polish LLM model (Bielik Instruct) is based on the Mistral model architecture [60], whose structure allows for much shorter inference times, despite the large number of parameters - $7 * 10^9$.

E. Justification of using specific LLMs

The Bart model is encoder-decoder model [36]. It has been chosen as an alternative to T5 model, which was first used for grammar correction in paper that served us as an inspiration [39]. The T5-like models are encoder-decoder model, so does Bart. However T5-like models were pretrained with masked language modeling as well as the SuperGLUE tasks whereas Bart was pretrained on a text denoising task which is closer to a grammar correction task and as suspected better results were achieved.

Concerning Llama choice from the bunch of possibilities of the usage of other models it is currently the best open source LLM on the market that is why we have used it. Another reason is that it is one of the few open source LLM, to our knowledge, that enables finetuning that we required for question generation.

V. CONCLUSIONS

In this paper, we present a novel method for the generation of questions. The method is based on rules extended by a grammar corrector. Our approach outperformed all widely used methods. Moreover, the most significant advantage of the Rule-Based method is its flexibility, speed and certainty of the output data. Rules can be easily added or modified depending on business needs. This solution also gives great control over the content of the output, its form, and tone.

The most popular methods for question generation based on finetuned language models (Bart, PLT5, Llama) showed worse results than our method. Moreover, those models are slow and require bigger computer resources (both for training and inference) and huge datasets for training.

The Rule-Based algorithm enhanced with grammar correction based on finetuned encoder-decoder models can generate questions with better metrics scores than LLMs-based methods. Our results showed that dedicated algorithms can outperform general methods such as LLMs. It should be emphasized that a grammar corrector improves scores even more. Results collected in a Table VI show that the Rule-Based algorithm with the grammar corrector can generate questions with the business-required quality.

The performance of all LLM-based models are similar. However, the Llama model finetuned on English texts with the Lora method has much lower scores than the other methods. This is probably related to errors in the translation of input data into English and back to Polish. This approach introduced additional bias to the method - related to adding synonyms and unnecessary words to the input data along with other translation artefacts. Examples of these errors are presented in the Table VIII.

Our novel Rule-Based method with a grammar corrector presented in this paper outperformed commonly used LLM-based methods, both in the quality of generated questions and in the speed of inference. In addition, it should be emphasized that our method is much faster than LLM-based methods. It is four times faster than BART and one hundred times faster than Llama (without a grammar corrector), while the model performance is on the same level.

ACKNOWLEDGMENT

This work was financed with European Union funds from the Smart Growth Operational Programme 2014-2020, Mea-

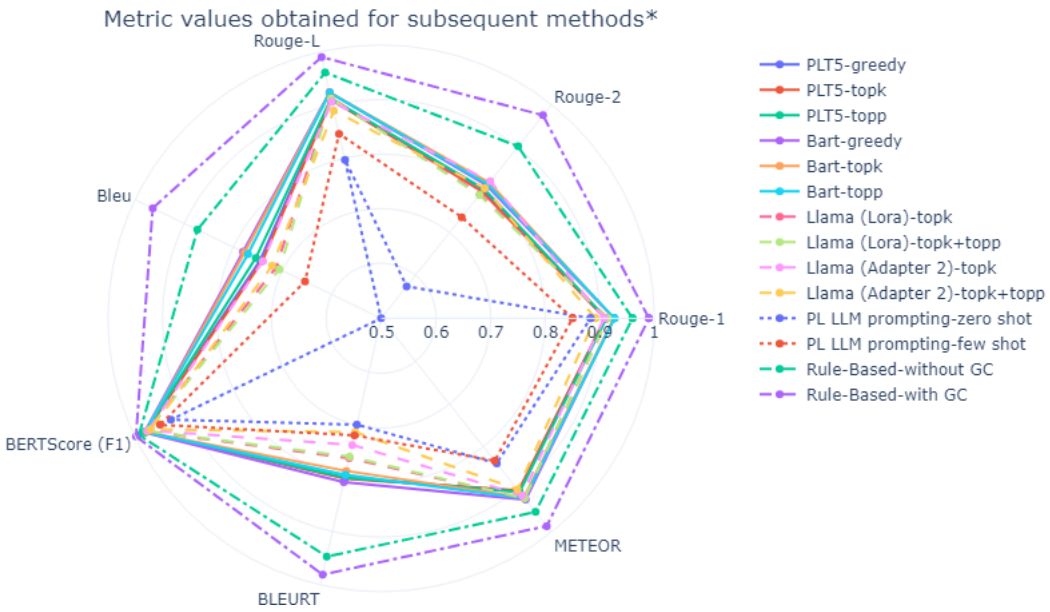

Figure 2. Metric values obtained for subsequent question generation methods. *To keep the chart clear, the worst-performed methods, PLT5-beam, Bart-beam and Llama finetuned on English data (Llama Lora, EN), have not been included in the chart.

sure 1.1.: R&D projects of enterprises. Submeasure 1.1.1.: Industrial research and development carried out by enterprises. The project was implemented during: 01.09.2023-01.07.2024.

REFERENCES

[1] Z. Wang, "Generating complex questions from knowledge graphs with query graphs," in *2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN)*, 2022, pp. 606–613.

[2] D. Rothman, *Transformers for Natural Language Processing: Build, train, and fine-tune deep neural network architectures for NLP with Python, Hugging Face, and OpenAI's GPT-3, ChatGPT, and GPT-4*. Packt Publishing Ltd, 2022.

[3] T. Gniazdowski, M. Bazan, and M. Marchwiany, "Test dataset to reproduce experiments," https://github.com/TomekGniazdowski/Automatic-questions-generation-based-on-keywords-using-language-models-paper-dataset, 2024.

[4] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015*, Jan. 2015.

[5] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 140–149. [Online]. Available: https://aclanthology.org/P16-1014

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[7] D. Lindberg, F. Popowich, J. Nesbit, and P. Winne, "Generating natural language questions to support learning on-line," in *Proceedings of the 14th European workshop on natural language generation*, 2013, pp. 105–114.

[8] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi, "Deep keyphrase generation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, R. Barzilay and M.-Y. Kan, Eds. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 582–592. [Online]. Available: https://aclanthology.org/P17-1054

[9] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: https://aclanthology.org/D14-1179

[10] S. R. Indurthi, D. Raghu, M. M. Khapra, and S. Joshi, "Generating natural language question-answer pairs from a knowledge graph using a rnn based question generation model," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 376–385.

[11] W. Hu, B. Liu, J. Ma, D. Zhao, and R. Yan, "Aspect-based question generation," *International Conference on Learning Representations, ICLR 2018*, 2018, workshop track.

[12] X. Shen, J. Chen, J. Chen, C. Zeng, and Y. Xiao, "Diversified query generation guided by knowledge graph," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 897–907.

[13] R. Doi, T. Charoenporn, and V. Sornlertlamvanich, "Automatic question generation for chatbot development," in *2022 7th International Conference on Business and Industrial Research (ICBIR)*. IEEE, 2022, pp. 301–305.

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[15] A. Kumar, S. Dandapat, and S. Chordia, "Translating web search queries into natural language questions," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, and T. Tokunaga, Eds. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1151

[16] R. Pandey, D. Chaudhari, S. Bhawani, O. Pawar, and S. Barve, "Interview bot with automatic question generation and answer evaluation," in *2023 9th international conference on advanced computing and communication systems (ICACCS)*, vol. 1. IEEE, 2023, pp. 1279–1286.

[17] G. Deena, K. Raja *et al.*, "Keyword extraction using latent semantic analysis for question generation," *Journal of Applied Science and Engineering*, vol. 26, no. 4, pp. 501–510, 2022.

[18] G. Deena and K. Raja, "Objective type question generation using natural language processing," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022.

[19] J. J. G. Torres, M. B. Bîndilă, S. Hofstee, D. Szondy, Q.-H. Nguyen, S. Wang, and G. Englebienne, "Automated question-answer generation for evaluating rag-based chatbots," in *Proceedings of the First Work-*

shop on Patient-Oriented Language Processing (CL4Health)@ LREC-COLING 2024, 2024, pp. 204–214.

[20] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2790–2799. [Online]. Available: https://proceedings.mlr.press/v97/houlsby19a.html

[21] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[22] T. e. a. Brown, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[23] M. Blšták and V. Rozinajová, "Automatic question generation based on sentence structure analysis using machine learning approach," *Natural Language Engineering*, vol. 28, no. 4, pp. 487–517, 2022.

[24] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[26] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[27] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=SkeHuCVFDr

[28] T. Sellam, D. Das, and A. Parikh, "BLEURT: Learning robust metrics for text generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 7881–7892. [Online]. Available: https://aclanthology.org/2020.acl-main.704

[29] Y. Huang, L. Sun, H. Wang, S. Wu, Q. Zhang, Y. Li, C. Gao, Y. Huang, W. Lyu, Y. Zhang *et al.*, "Trustllm: Trustworthiness in large language models," *arXiv preprint arXiv:2401.05561*, 2024.

[30] Z. Dong, Z. Zhou, C. Yang, J. Shao, and Y. Qiao, "Attacks, defenses and evaluations for llm conversation safety: A survey," *arXiv preprint arXiv:2402.09283*, 2024.

[31] C. Walker, C. Rothon, K. Aslansefat, Y. Papadopoulos, and N. Dethlefs, "Safellm: Domain-specific safety monitoring for large language models: A case study of offshore wind maintenance," *arXiv preprint arXiv:2410.10852*, 2024.

[32] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, https://github.com/explosion/spaCy.

[33] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1373–1378. [Online]. Available: https://aclanthology.org/D15-1162

[34] J. Nivre and J. Nilsson, "Pseudo-projective dependency parsing," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, K. Knight, H. T. Ng, and K. Oflazer, Eds. Ann Arbor, Michigan: Association for Computational Linguistics, Jun. 2005, pp. 99–106. [Online]. Available: https://aclanthology.org/P05-1013

[35] W. Kieraś and M. Woliński, "Morfeusz 2 – analizator i generator fleksyjny dla języka polskiego," *Język Polski*, vol. XCVII, no. 1, pp. 75–83, 2017.

[36] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[37] S. Dadas, "Polish bart base," https://huggingface.co/sdadas/polish-bart-base, 2022.

[38] A. Chrabrowa, Ł. Dragan, K. Grzegorczyk, D. Kajtoch, M. Koszowski, R. Mroczkowski, and P. Rybak, "Evaluation of transfer learning for Polish with a text-to-text model," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 4374–4394. [Online]. Available: https://aclanthology.org/2022.lrec-1.466

[39] S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, "A simple recipe for multilingual grammatical error correction," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 702–707. [Online]. Available: https://aclanthology.org/2021.acl-short.89

[40] P. Lison and J. Tiedemann, "OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, Eds. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 923–929. [Online]. Available: https://aclanthology.org/L16-1147

[41] M. Bañón, P. Chen, B. Haddow, K. Heafield, H. Hoang, M. Esplà-Gomis, M. L. Forcada, A. Kamran, F. Kirefu, P. Koehn, S. Ortiz Rojas, L. Pla Sempere, G. Ramírez-Sánchez, E. Sarrías, M. Strelec, B. Thompson, W. Waites, D. Wiggins, and J. Zaragoza, "ParaCrawl: Web-scale acquisition of parallel corpora," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 4555–4567. [Online]. Available: https://aclanthology.org/2020.acl-main.417

[42] C. Borowski, "Polish verb conjugator," 2023. [Online]. Available: https://github.com/chriseborowski/Polish-verb-conjugator

[43] M. Grootendorst, "Keybert: Minimal keyword extraction with bert." 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4461265

[44] S. Rose, D. Engel, N. Cramer, and W. Cowley, *Automatic Keyword Extraction from Individual Documents*. John Wiley & Sons, Ltd, 2010, ch. 1, pp. 1–20. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470689646.ch1

[45] F. Barrios, F. López, L. Argerich, and R. Wachenchauzer, "Variations of the similarity function of textrank for automated summarization," 2015, simposio Argentino de Inteligencia Artificial.

[46] AI@Meta, "Llama 3 model card," *https://huggingface.co/meta-llama/Meta-Llama-3-8B*, 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[47] R. Zhang, J. Han, C. Liu, A. Zhou, P. Lu, Y. Qiao, H. Li, and P. Gao, "LLaMA-adapter: Efficient fine-tuning of large language models with zero-initialized attention," 2024. [Online]. Available: https://openreview.net/forum?id=d4UiXAHN2W

[48] L. AI, "Litgpt," https://github.com/Lightning-AI/litgpt, 2023.

[49] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. [Online]. Available: http://www.aclweb.org/anthology/P11-1015

[50] R. Tatman, "Question-answer dataset." [Online]. Available: https://www.kaggle.com/datasets/rtatman/questionanswer-dataset

[51] A. Bordes, N. Usunier, S. Chopra, and J. Weston, "Large-scale simple question answering with memory networks," *CoRR*, vol. abs/1506.02075, 2015. [Online]. Available: http://arxiv.org/abs/1506.02075

[52] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, J. Su, K. Duh, and X. Carreras, Eds. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. [Online]. Available: https://aclanthology.org/D16-1264

[53] W. Xiong, J. Wu, H. Wang, V. Kulkarni, M. Yu, S. Chang, X. Guo, and W. Y. Wang, "TWEETQA: A social media focused question answering dataset," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, A. Korhonen, D. Traum, and L. Màrquez, Eds. Florence, Italy: Association

for Computational Linguistics, Jul. 2019, pp. 5020–5031. [Online]. Available: https://aclanthology.org/P19-1496
[54] D. C. Austin Walters, "Sentence classification." [Online]. Available: https://github.com/lettergram/sentence-classification
[55] N. B. et al., "Deep translator." [Online]. Available: https://github.com/nidhaloff/deep-translator
[56] K. Ociepa, L. Flis, K. Wróbel, S. Kondracki, SpeakLeash Team, and Cyfronet Team, "Introducing bielik-7b-instruct-v0.1: Instruct polish language model," 2024, accessed: 2024-19-07. [Online]. Available: https://huggingface.co/speakleash/Bielik-7B-Instruct-v0.1
[57] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7
[58] ——, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx
[59] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.06196
[60] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

APPENDIX

A. Decoding method hyperparameters

Decoding hyperparameters for used generative models for proposed methods are presented in Table II.

Table II
HYPERPARAMETER VALUES FOR USED DECODING METHODS.

Decoding method	Hyperparameter values
greedy	no repeat ngram size: 2
beam	num beams: 5
topk	topk: 50; temp: 0.3
topp	topp: 0.8; temp: 0.3
topk+topp	topk: 50; topp: 0.8; temp: 0.3

B. Error probabilities for datasets creation

For error probability values for datasets for the keywords to sentence models see Table III, for error probability values for dataset for grammar corrector see Table IV.

Table III
ERROR PROBABILITY VALUES FOR THE DATASET FOR THE KEYWORDS TO SENTENCE MODELS.

Error type	Probability
noun to lemma	0.3
verb to lemma	0.3
adjective to lemma	0.5
adverb to lemma	0.5

C. Evaluation of hallucinations

Evaluation of hallucination detection of the presented methods is presented in Table V.

D. Zero-shot and few-shot prompts for Polish LLM

An example of a zero-shot prompt for Polish LLM is presented in the Figure 3, an example of a few-shot prompt for Polish LLM is presented in the Figure 4.

Table IV
ERROR PROBABILITY VALUES FOR THE DATASET FOR THE GRAMMAR CORRECTOR MODEL.

Error type	Probability
noun to lemma	0.30
verb to lemma	0.30
adjective to lemma	0.85
adverb to lemma	0.85
lemma verb to another form	0.85
drop aux	0.75

Table V
AN EVALUATION OF HALLUCINATION DETECTION OF THE PRESENTED METHODS. THE ANNOTATOR RECEIVED 35 RANDOM SAMPLES - MODEL INPUT TEXTS AND OUTPUTS GENERATED BY ALL FIVE METHODS. ANNOTATORS DETERMINED WHETHER THE TEXT CONTAIN HALLUCINATIONS. PRESENTED RATIOS ARE THE NUMBER OF TEXTS WITH HALLUCINATIONS TO AMOUNT OF ALL THESE RANDOM TEXTS (35).

Evaluation	Hallucinations
Bart-greedy	5.71%
Llama (Lora)-topk	8.57%
Llama (Lora, EN)-topk+topp	22.86%
PL LLM prompting-few shot	2.86%
Rule-Based-with GC	**0.00%**

1) System prompt: Odpowiadaj krótko, precyzyjnie i wyłącznie w języku polskim.
2) User prompt: Jesteś asystentem AI, który bardzo dobrze tworzy pytania i zdania rozkazujące na podstawie słów kluczowych w języku polskim. Wykorzystując podane słowa kluczowe wygeneruj poprawne gramatycznie pytanie lub zdanie rozkazujące. Możesz zmienić formę przedstawionych słów lub dodać własne słowa by zdanie było logiczne i poprawne. Zwróć jedynie wygenerowane zdanie. Słowa kluczowe: *keywords*.

Figure 3. Zero-shot prompt for Polish LLM - Bielik.

1) System prompt: Odpowiadaj krótko, precyzyjnie i wyłącznie w języku polskim. Jesteś asystentem AI, który bardzo dobrze tworzy pytania i zdania rozkazujące na podstawie słów kluczowych w języku polskim. Zwróć jedynie wygenerowane zdanie. Przykłady zaprezentowano poniżej.
2) User prompt: Input: miejce usterki,
Output: Podaj proszę miejsce wystąpienia usterki.,
Input: wybierz nazwa klienta z listy,
Output: Wybierz proszę nazwę klienta z listy.,
Input: budżet projektu,
Output: Ile wynosi budżet opisywanego projektu?,
Input: zgłaszający problem,
Output: Kim jest osoba zgłaszająca problem?,
Input: *keywords*,
Output:

Figure 4. Few-shot prompt for Polish LLM - Bielik.

E. Outputs generated by the subsequent methods

In Tables VI and VII there are presented questions generated for subsequent, best performing methods. In table VIII there are presented errors in translations in the Llama finetuned with the Lora method on English dataset.

Table VI
QUESTIONS GENERATED BY THE SUBSEQUENT RULE-BASED METHODS. PLEASE NOTE THAT THE MODELS' INPUT INCLUDES ONLY KEYWORDS AND NO PUNCTUATION MARKS.

Input data	Rule-Based-without GC	Rule-Based-with GC
zaliczka zwrócona	Czy zaliczka zwrócona?	Czy zaliczka zostanie zwrócona?
potrzebna zaliczka	Czy potrzebna zaliczka?	Czy potrzebna jest zaliczka?
zaliczka do wypłaty	Czy zaliczka do wypłaty?	Czy zaliczka jest do wypłaty?
możliwość uzupełnienia wniosku	Czy możliwość uzupełnienia wniosku?	Czy jest możliwość uzupełnienia wniosku?
zgłoszenie delegacji z datą wsteczną	Czy zgłoszenie delegacji z datą wsteczną?	Czy zgłoszenie delegacji jest z datą wsteczną?
forma wypłaty wypłacanej zaliczki	Wybierz proszę forma wypłaty wypłacanej zaliczki z listy.	Wybierz proszę formę wypłaty wypłacanej zaliczki z listy.
waluta kosztów transportu	Wybierz proszę waluta kosztów transportu z listy.	Wybierz proszę walutę kosztów transportu z listy.

Table VII
QUESTIONS GENERATED BY THE SUBSEQUENT BEST PERFORMING ONE-STEP METHODS. THE INPUT DATA IS DATA FROM THE TABLE VI.

Bart-greedy	Llama (Lora)-topk	PL LLM prompting-few shot
Czy zaliczka może być zwrócona?	Czy zaliczka może być zwrócona?	Czy zaliczka zostanie zwrócona?
Czy potrzebna jest zaliczka?	Czy potrzebna jest zaliczka?	Czy potrzebna jest zaliczka?
Czy zaliczka do wypłaty?	Czy zaliczka jest do wypłaty?	Czy zaliczka do wypłaty jest przewidziana?
Czy jest możliwość uzupełnienia wniosku?	Czy istnieje możliwość uzupełnienia wniosku?	Czy możliwa jest opcja uzupełnienia wniosku?
Czy zgłoszenie delegacji jest datą wsteczną?	Czy zgłoszenie delegacji z datą wsteczną jest możliwe?	Czy delegacja powinna być zgłoszona z datą wsteczną?
Wybierz proszę formę wypłaty wypłacanej zaliczki z listy.	Wybierz proszę formę wypłaty wypłacanej z zaliczki z listy.	Wybierz proszę formę wypłaty zaliczki z listy.
Wybierz proszę walutę kosztów transportu z listy.	Wybierz proszę walutę kosztów transportu z listy poniżej.	Wybierz proszę walutę kosztów transportu z listy.

Table VIII
EXAMPLES OF INCORRECT TRANSLATIONS IN THE LLAMA FINETUNED WITH THE LORA METHOD ON ENGLISH DATASET. PLEASE NOTE THAT THE MODELS' INPUT INCLUDES ONLY KEYWORDS AND NO PUNCTUATION MARKS.

Polish prompt	Polish prompt translated to English	Llama Lora EN output	English output translated to Polish
czy zatwierdzasz rozliczenie	do you approve the settlement	Do you approve of the settlement?	Czy zgadzasz się z tą ugodą?
czy akceptujesz ten wniosek	do you accept this request	Do you accept this request?	Czy akceptujesz tę prośbę?
czy uregulowano płatność	whether the payment has been settled	When will it be known whether the payment has been settled?	Kiedy będzie wiadomo, czy płatność została rozliczona?
czy użyj zapisanego rachunku do zaliczki	or use saved account for down payment	Or use it to be saved up for a down payment.	Albo wykorzystaj je jako oszczędności na zaliczkę.
podaj proszę twój stopień znajomości mikroserwisów	please indicate your level of knowledge of microservices	Please indicate your level of knowledge of microservices.	Proszę podać poziom wiedzy na temat mikrousług.