Developing PUGG for Polish: A Modern Approach to KBQA, MRC, and IR Dataset Construction

Anonymous ACL submission

Abstract

Advancements in AI and natural language processing have revolutionized machine-human language interactions, with question answering (QA) systems playing a pivotal role. The knowledge base question answering (KBQA) task, utilizing structured knowledge graphs (KG), allows to handle extensive knowledge-intensive questions. However, a significant gap exists in KBQA datasets, especially for low-resource languages. Many existing construction pipelines for these datasets are outdated and inefficient in human labor, not utilizing modern assisting tools like Large Language Models (LLM) to reduce the workload. To address this, we have designed and implemented a modern, semi-automated approach for creating datasets, encompassing tasks such as KBQA, Machine Reading Comprehension (MRC), and Information Retrieval (IR), specifically tailored for low-resource environments. We executed this pipeline and introduced the PUGG dataset, the first Polish KBQA dataset, along with novel datasets for MRC and IR. Additionally, we provide a comprehensive implementation, insightful findings, detailed statistics and evaluation of baseline models.

1 Introduction

Question answering (QA) systems serve as a sophisticated interface between humans and computers. To further enhance their utility, we need QA systems that are capable of answering questions based on extensive knowledge (Petroni et al., 2021). The knowledge base question answering (KBQA) task addresses this need by using structured knowledge graphs (KG), to provide accurate and relevant answers (Lan et al., 2021). KBQA leverage these graphs, which are rich with interconnected entities and relationships, to decode complex queries and deliver precise answers. Importantly, systems that reason over KGs are more resistant to the phenomenon of hallucinations, common in large language models (LLM) (Baek et al., 2023). Additionally, the inherent flexibility of KGs facilitates easy modification and updating, ensuring the use of only the most current and accurate facts.

However, a significant gap exists in KBQA datasets. Many are schematic and not natural in their language, or they rely on discontinued knowledge graphs (Lan et al., 2021; Steinmetz and Sattler, 2021; Jiang and Usbeck, 2022). By *natural* we refer to naturally occurring questions (Kwiatkowski et al., 2019). While a wider range of KBQA datasets is available for English, most low-resource languages, including Polish, lack such resources (Korablinov and Braslavski, 2020). This scarcity is part of a broader issue prevalent in the field of NLP concerning low-resource languages (Augustyniak et al., 2022). Recognizing this gap, we set out to create a KBQA dataset for Polish. During extensive studies of existing works to find the most efficient methods for dataset creation, we faced several challenges. Many datasets were built on simpler predecessors (Korablinov and Braslavski, 2020; Kaffee et al., 2023), many construction pipelines were outdated and inefficient in terms of human labor, as they did not utilize modern tools that could reduce human work, such as assisting Large Language Models (LLM). LLMs have opened new opportunities for assisting human annotators, especially in low-resource languages where the range of pretrained models is limited (Gilardi et al., 2023; Kuzman et al., 2023).

Consequently, we decided to design, implement, and execute a modern approach to creating KBQA datasets, specifically tailored for the low-resource environment. We selected Wikidata as KG due to its extensive, multilingual coverage, its dynamic, open, and free nature (Vrandečić and Krötzsch, 2014). An advantageous byproduct of this pipeline was the concurrent development of machine reading comprehension (MRC) and information retrieval (IR) datasets, requiring no extra human an-

notation. MRC is essential for AI to understand and analyze texts like a human reader (Rajpurkar et al., 2016; Kwiatkowski et al., 2019), while IR is crucial for efficiently extracting relevant information from vast databases (Nguyen et al., 2017; Thakur et al., 2021).

We summarize our contributions as follows:

- We introduce the PUGG [1] dataset, which encompasses three tasks — KBQA, MRC, and IR[2]. This dataset features natural factoid questions in Polish and stands out as the first Polish KBQA resource [3]. To address a range of complexities, we have enriched the dataset by complementing natural questions with simpler, template-based questions.

- We propose a semi-automated dataset construction pipeline, specifically designed for low-resource environments. Accompanying this pipeline is a comprehensive implementation [4], along with insightful findings and detailed statistics. These provide valuable resources for future developers of datasets. Additionally, we developed and detailed custom methods, e.g. for entity linking, useful in diverse contexts.

- We provide an evaluation of baseline models, thereby establishing benchmarks for future research using the PUGG dataset.

2 Related Work

KBQA Existing KBQA datasets have been comprehensively studied and compared in existing works done by Korablinov and Braslavski (2020) and Jiang and Usbeck (2022). A significant finding is the lack of a Polish KBQA dataset. Most KBQA datasets are primarily in English, with exceptions like the Chinese NLPCC-KBQA (Duan and Tang, 2018), Russian RuBQ (Korablinov and Braslavski, 2020), the multilingual QALD (Perevalov et al., 2022) and MCWQ (Cui et al., 2022) (both not including Polish). The closest dataset resembling a KBQA task in Polish is the multilingual MKQA (Longpre et al., 2021), where approximately 42% of its 10,000 questions are answerable by Wikidata entities. However, MKQA cannot be classified as a true KBQA dataset due to the lack of annotated topic entities.

The study conducted by Korablinov and Braslavski (2020) outlines the various question generation techniques used in existing KBQA datasets. For generating natural questions in our study, we adopted a question generation technique based on query suggestion, originally introduced by Berant et al. (2013). This technique is effective for acquiring natural factoid questions likely to be posed to a QA system, similar to the approaches used in datasets like NQ (Kwiatkowski et al., 2019) and WikiQA (Yang et al., 2015), which were built from questions asked to search engines. For template-based questions, our approach involved creating questions from predefined reasoning templates, a common method in many KBQA datasets (Bordes et al., 2015; Su et al., 2016; Dubey et al., 2019). Several KBQA datasets used crowdsourced paraphrasing for question diversification (Talmor and Berant, 2018; Su et al., 2016; Dubey et al., 2019). In contrast, our approach automates this process, by incorporating humans only during final verification.

IR Recently, the BEIR-PL (Wojtasik et al., 2023) benchmark was created. It is an automatic machine translation of the BEIR (Thakur et al., 2021) benchmark, a popular zero-shot retrieval benchmark, which was originally only for the English language. The MQUPQA (Rybak, 2023) dataset is a composition of multiple already existing Polish and multilingual datasets, like CzyWiesz (Marcińczuk et al., 2013), MKQA (Longpre et al., 2021). Additionally, the MQUPQA dataset incorporates other automatic methods for question and answer generation, such as utilizing the generative capabilities of the GPT-3 model (Brown et al., 2020) or employing templates inspired by the structure of Wikipedia. A passage retrieval task was featured at PolEval (Łukasz Kobyliński et al., 2023) competition. It was composed of three datasets from various domains, Wikipedia based, e-commerce shop FAQ and legal questions. As of now, a Polish Information Retrieval Benchmark (PIRB) [5] provides a platform to evaluate prepared solutions across a variety of datasets. The models refered in this benchmark represent the current state-of-the-art in Polish IR.

[1] The name "PUGG" refers to "Pirate Pugg" a fictional character from "The Sixth Sally" of "The Cyberiad" by Stanisław Lem. Pirate Pugg is depicted as a being obsessed with gathering information.

[2] https://anonymous.4open.science/r/pugg-EC84

[3] The dataset license: CC BY-SA 4.0

[4] github.com/anonymized

[5] https://huggingface.co/spaces/sdadas/pirb

MRC QA datasets often have a close relationship with IR datasets. CzyWiesz dataset is a dataset based on the *Did you know?* section of Wikipedia, with provided answers and also relevant articles. Another example is the PolQA (Rybak et al., 2022) dataset, which is comprised of general questions from quiz shows, annotated with relevant passages from Wikipedia. The PoQuAD (Tuora et al., 2023) dataset is structured around questions that have been manually annotated to correspond with the best articles on Wikipedia, mirroring the methodology of the SQuAD (Rajpurkar et al., 2016) dataset. Contrastively, our dataset consists of naturally occurring questions, which are afterward annotated to relevant articles.

3 Definitions

In the tasks of KBQA, MRC, and IR, a common element is the textual question q. We denote set of questions as $\mathcal{Q}$. Despite *query* being common in the field of IR, we use *question* and *query* interchangeably, as our dataset's queries take the form of questions.

KBQA We denote KG as a multi-relational heterogeneous graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, composed of three elements: a set of entities $\mathcal{E}$, a set of relation predicates $\mathcal{R}$, and a set of triples (facts) $\mathcal{T}$. Each triplet $(h, r, t) \in \mathcal{T}$ indicates a relation predicate r between two entities, a head entity h and a tail entity t, where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ (Hamilton et al., 2017). In the KBQA task, a textual question q and associated topic entities $\mathcal{E}_{t,q} \subset \mathcal{E}$ are given. The objective is to retrieve answer entities $\mathcal{A}_q \subset \mathcal{E}$ that satisfy the question based on facts in the $\mathcal{G}$. Therefore, we denote KBQA dataset as $\mathcal{D}_{KBQA} = \{(q, \mathcal{E}_{t,q}, \mathcal{A}_q)\}$.

MRC In MRC, the aim is to answer a textual question q based on a given text passage p_q. We denote MRC dataset as $\mathcal{D}_{MRC} = \{(q, p_q, a_q)\}$, where a_q is the answer extracted from p_q.

IR The IR task focuses on finding a passage p from a large corpus that are relevant to a query q. The corpus $\mathcal{C}$ is defined as a set of passages, i.e., $\mathcal{C} = \{p_1, p_2, ..., p_n\}$. The IR dataset is denoted as $\mathcal{D}_{IR} = \{(q, p_q)\}$, where $p_q \in \mathcal{C}$ denotes a passage that is relevant to the query q.

4 Construction Pipeline

This section introduces the construction pipeline for the PUGG dataset, specifically designed for creating a dataset with natural and factoid questions in a semi-automated manner. This approach significantly reduces the workload of human annotators. We outline the pipeline's fundamental design, presented in Figure 1, emphasizing its adaptability to various environments. While this part focuses on the general framework, specific implementation details, such as the models and algorithms used, will be discussed in Section 5.

Question Formulation The initial step of our pipeline involves acquiring a variety of natural factoid questions. We initiated our process using existing datasets to minimize the need for manual annotation. From previously existing QA datasets, we extract question prefixes ranging from basic (*'who...'*, *'when...'*) to more specific (*'which Canadian athlete...'*, *'which theater co-created...'*). Then, the gathered prefixes are completed to formulate a set of questions. We can employ various methods, including rule-based approaches and language models (Das et al., 2021), and for natural questions, we can also integrate external services.

At this stage, we have a collection of question condidates q', as some of which may be incorrect. These inaccuracies are not a concern at this point, as they will be filtered out during the human verification process, detailed in Section 4.

Passage Construction The next phase involves text passages retrieval to answer the formulated questions. We use a data source with referenced graph entities, which in our case is Wikipedia. To find relevant articles for each question, various retrieval techniques can be employed, such as dense retrieval (Reimers and Gurevych, 2019) with additional reranking. Once relevant articles are identified, they are segmented into smaller passages and reranked to prioritize passages most likely to contain an answer.

All passages constructed in this phase are added to the passage corpus $\mathcal{C}$ needed for IR task.

Textual Answers, Answer Entities We select the most accurate passage as candidate passage p'_q and we apply a QA model, such as LLM or pretrained extractive model, to tag a span of passage denoting a candidate textual answer a'_q. Such textual answers contain hyperlinks to other articles, that are associated with specific Wikidata entities. We extract these entities and build set of candidate answer entities $\mathcal{A}'_q$.

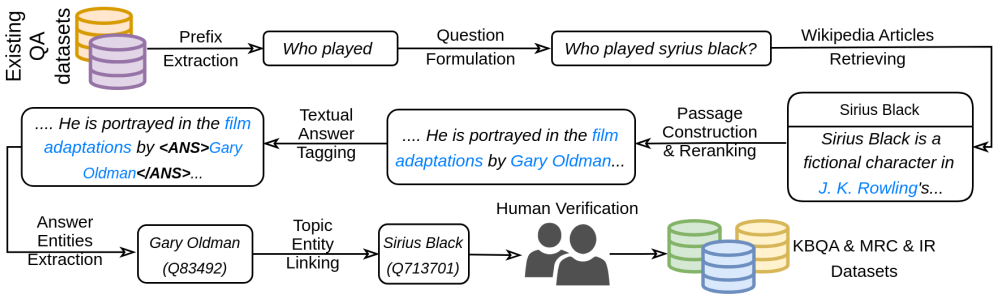

Figure 1: Overview of the proposed construction pipeline for natural questions. The figure shows the processing of a single example. Rounded rectangles represent acquired data, with blue text indicating a hyperlink to another Wikipedia article. Arrow descriptions indicate procedures. Symbol of people denotes step involving human verification depicted in Section 4: Human Verification and in Figure 4.

Topic Entities The subsequent step in our pipeline is performing entity linking process to identify and link the KG entities that are mentioned in the questions. We refer to them as candidate topic entities $\mathcal{E}'_{t,q}$.

Human Verification To this point, we have acquired all necessary data to construct the KBQA, MRC, and IR datasets: questions q' accompanied by a passage p'_q, textual answer a'_q, answer entities $\mathcal{A}'_q$, and topic entities $\mathcal{E}'_{t,q}$. All these elements are obtained through fully automated processes. While automation significantly reduces the need for human labor, it is not entirely error-proof. To assure the high quality of our dataset, we implement a human verification process. The detailed procedure of this human verification is depicted in Figure 2. During this process, candidate elements q', p'_q, a'_q, $\mathcal{A}'_q$, and $\mathcal{E}'_{t,q}$ undergo verification. This leads to the final elements q, p_q, a_q, $\mathcal{A}_q$, and $\mathcal{E}_{t,q}$, respectively. The final sets $\mathcal{A}q \subseteq \mathcal{A}'_q$ and $\mathcal{E}_{t,q} \subseteq \mathcal{E}'_{t,q}$ indicate that the validated entities are subsets of their initial candidate sets. Note that the verification procedure (Figure 2) consists of multiple conditions, which may result in the datasets varying in size. This is reflected in the relationship $|\mathcal{D}_{IR}| \geq |\mathcal{D}_{MRC}| \geq |\mathcal{D}_{KBQA}|$.

Template-based KBQA While the proposed pipeline generates natural questions, we also created template-based questions to enrich our dataset. We wanted to provide a broader training and evaluation platform, by offering a more schematic and straightforward set of questions, with ensured existence of a reasoning path between topic and answer entities. The template-based questions are also beneficial for semantic parsing-based KBQA methods (Lan et al., 2021).

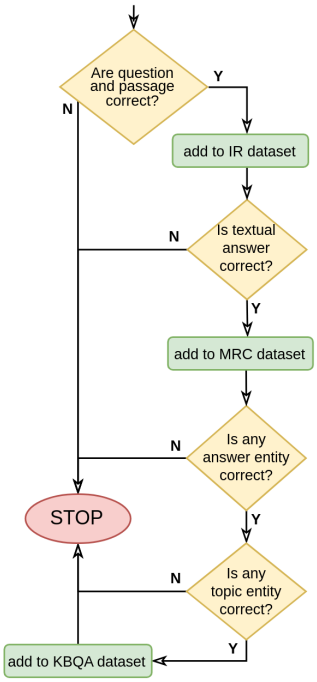

Figure 2: The human verification procedure for all acquired candidates.

The procedure of creating template-based questions is depicted in Figure 3. We begin by creating sets of templates and one in natural language, that represent specific reasoning paths in the KG. We specify potential entities and relations to be used within these templates. To construct questions, we insert these entities and relations into the natural language template. Then, we run the corresponding SPARQL queries to retrieve answer entities.

At this stage, the formulated questions might sound unnatural, especially in inflected languages like Polish. To address this, we use two strategies: word inflection and question paraphrasing. We automate the inflection process using NLP tools

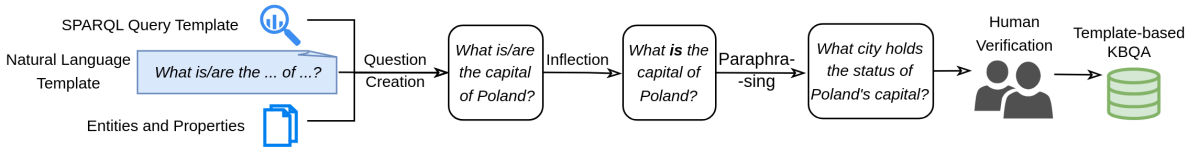

Figure 3: Overview of the proposed construction pipeline for template-based questions. The figure shows the processing of a single example. Symbol of people denotes step involving human verification to ensure all questions are meaningful.

like spaCy (Honnibal et al., 2020) or LLMs. For greater diversity and complexity, we also use LLMs to paraphrase the questions. Given the automation of these processes, we ensure the meaningfulness of all questions through human verification.

5 Pipeline Execution

This section delves into the specific implementation of the construction pipeline for the PUGG dataset, as previously outlined in a general framework in Section 4. Our implementation was adapted for Polish NLP resources, which face challenges like limited task-specific pretrained models and lower performance compared to English.

Question Formulation In implementing our question acquisition step, we utilized two Polish datasets, *CzyWiesz* (Marcińczuk et al., 2013) and *PoQuAD* (Tuora et al., 2023). Question prefixes were extracted either by taking the first $\{1, 2, 3\}$ tokens from each question or by extracting text up to the first occurrence of a named entity, using three NER models: $pl\_core\_news\_sm$, $pl\_core\_news\_lg$ from Spacy (Honnibal et al., 2020), and WikiNEuRal (Tedeschi et al., 2021). Each of these models provided a unique perspective in identifying named entities, thereby contributing to the variety of the prefixes. To formulate natural questions from these prefixes, we followed previous studies (Berant et al., 2013; Rybin et al., 2021) and used the Google Suggest API.

Passage Construction We followed established methodologies from prior research (Kwiatkowski et al., 2019) and employed the Google Search Engine [6] to retrieve Wikipedia articles relevant to each question. Using the API, we processed the top 10 search results, focusing on Wikipedia entries. Questions without a Wikipedia article in the top 10 results were discarded. The text and inter-article references of these Wikipedia articles were then obtained using the Wikipedia API[7]. The retrieved articles were segmented into shorter passages using a sliding window approach, with a window length of 120 tokens and a step size of 60 tokens. For each question, we reranked these passages employing the PyGaggle (Pradeep et al., 2023) library with the multilingual model *unicamp-dl/mt5-3B-mmarco-en-pt* (Bonifacio et al., 2021).

Textual Answers, Answer Entities For textual answer tagging, we employed *GPT-3.5-turbo* [8] (Brown et al., 2020) with an originally designed prompt, detailed in Appendix A. Due to model's generative nature and its tendency to alter or paraphrase original text, we developed a custom method to accurately extract tagged segments. This method is described in Appendix A. As previously described, candidate answer entities were directly referenced in the text, allowing for their straightforward extraction.

Topic Entities Implementing the entity linking step presented several challenges, primarily due to the lack of readily available tools or models for entity linking in the Polish language. Our testing of multilingual models like *mGENRE* (De Cao et al., 2022) and adapted for Polish *BLINK* (Wu et al., 2020), yielded unsatisfactory results, particularly for short contexts such as individual questions. Additionally, given the planned human verification stage, a method with high recall was desired. To address these challenges, we developed a heuristic method specifically tailored to our requirements and the available resources. The method is detailed in Appendix B.

Human Verification The general procedure for human verification is illustrated in Figure 2. We implemented this by dividing it into two distinct stages. The first stage focused on identifying two

[6] https://developers.google.com/custom-search/v1/overview

[7] https://pl.wikipedia.org/w/api.php

[8] https://platform.openai.com/docs/models/overview

aspects: questions with correctly assigned passages and questions where the textual answers within these passages were accurately tagged. The second stage of human verification had two parts: first, annotators marked the correct answer entities, and then they identified the correct topic entities. More details about annotation procedures and guidelines are presented in Appendix C.

Template-based KBQA The developed templates are detailed in Appendix E.1. It is important to note that while our template-based KBQA dataset contains fewer templates compared to other datasets, ours are more general. This is achieved by injecting not only entities but also relations into the templates, enhancing their diversity. We used entities from Wikipedia's Vital Articles Level 4 [9] and 173 manually selected relations. Any entities lacking a Polish label were excluded. Given the vast number of possible inputs (entities and relations) for the templates, and the fact that most will not yield answers, random input selection was not feasible. Therefore, we divided the process into two steps, each involving the execution of a SPARQL query. First, we gathered potential sets of inputs, and then, we selected some of these sets to retrieve answers. We also utilized the selected inputs to create questions using natural language templates.

Then, we conducted both inflection and paraphrasing of the constructed questions using the *GPT-3.5-turbo* model (Brown et al., 2020). Following this, we filtered out examples without high similarity to their original form, based on the longest common sequence analysis. The questions were verified by one annotator. The statistics of the verification can be found in Appendix E.1.

Outcome The execution of our pipeline resulted in the creation of the PUGG dataset, featuring three tasks: KBQA (natural and template-based), MRC, and IR. Statistics for each dataset are presented in Table 1, and detailed statistics of the pipeline steps are available in Appendix D. Due to the utilized sliding window, the passage corpus $\mathcal{C}$ was filtered to remove all passages overlapping with any p_c. As Wikidata is a vast KG and using it for research can be inconvenient, we provide sampled versions of the KG: Wikidata1H and Wikidata2H. These are subgraphs created by traversing 1 or 2 relations from each answer and topic entity, representing two different levels of data complexity.

[9] https://en.wikipedia.org/wiki/Wikipedia:Vital_articles

Dataset		Size
KBQA (natural)	*train*	2776
	test	695
total		3471
KBQA (template-based)	*train*	1697
	test	425
total		2122
KBQA (all)	*train*	4473
	test	1120
total		2122
MRC	*train*	6961
	test	1741
total		8702
IR	*corpus*	309621
	queries	10751

Table 1: Summary of dataset sizes.

6 Experimental Setup

In this section, we outline the evaluation methodology used to assess the performance of baseline models on the PUGG dataset.

KBQA For the KBQA baseline, we evaluated the performance of KAPING (Baek et al., 2023), a zero-shot framework that leverages a LLM for retrieving answer entities. We made a slight modification to the knowledge retriever module by incorporating a step that retrieves a subgraph of the KG by traversing n edges, regardless of their direction, from the topic entities. Following this, we follow the original procedure, which involves retrieving k triples based on their textual embeddings. For embedding purposes, we utilized the *MMLW-retrieval-roberta-large* retrieval model[10]. We employed *gpt-3-turbo* as the LLM, prompted with tailored queries as detailed in Appendix F. The hyperparameters were selected empirically, setting $k = 40$ and choosing n to be 3 for Wikidata1H and 2 for Wikidata2H. As a metric, we employed accuracy as the metric, which measures the proportion of answers included in the LLM's response for each question. It is calculated as follows:

$$\text{Accuracy} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{\text{num of incl. answers}_i}{|\mathcal{A}_i|}$$

While Baek et al. (2023) also used accuracy, we refined it by calculating the correct answer proportion per example and excluding entities' aliases, providing a more realistic measure of KBQA efficacy.

[10] https://huggingface.co/sdadas/mmlw-retrieval-roberta-large

MRC For MRC task, we selected models commonly used for the extractive question answering task. We trained and evaluated a *HerBert* (Mroczkowski et al., 2021) models in extractive fashion, alongside with a generative approach using the *plT5* (Chrabrowa et al., 2022) models. Models were trained for 10 epochs and evaluated with SQuAD metrics (Rajpurkar et al., 2016). *Exact match* measures the percentage of predictions that exactly match the gold answer. The *F1* metric measures the average token overlap between the prediction and ground truth answer, where both, prediction and answer, are treated as a bag of tokens.

IR Recently, IR has gained significant interest within the Polish research community and many models were developed and open for research community. These models have been already pretrained on large datasets, that is why we did not fine-tuned them to our dataset. The silver retriever (Rybak and Ogrodniczuk, 2023) model was trained on MAUPQA dataset. We also evaluated E5 (Wang et al., 2024) multilingual embedding models, which were trained on contrastive objective on large weakly-labeled text pairs and afterwards fine-tuned on existing datasets and are performing very well on Polish texts. MMLW retrieval models [11] were trained on parallel corpus with Polish-English text pairs with a *bge-large-en* (Xiao et al., 2023) teacher model and are currently on the top of PIRB leaderboard. We also provide results of well established BM25 (Robertson and Zaragoza, 2009) baseline with Morfologik [12] plugin in Elasticsearch.

Additionally, we conducted an evaluation of reranker models, focusing on those developed in the BEIR-PL benchmark, as well as, recent models that have appeared on PIRB leaderboard. Those models are based on *polish-roberta* [13] model with knowledge distillation from mT5-13B model introduced in mMARCO (Bonifacio et al., 2021) publication. For the purpose of reranking, we employed the BM25 retrieval algorithm to select the top 100 passages for subsequent analysis. Finally, we provide a score of the combination of the best retriever and reranker, namely *multilingual-e5-large retriever* and *polish-reranker-large-ranknet reranker*, to evaluate currently the best IR pipeline available. In order to compare the models, we calculated the well established metrics for IR task: MRR@k, NDCG@k, Recall@k(Thakur et al., 2021; Wojtasik et al., 2023).

[11] https://huggingface.co/sdadas/mmlw-retrieval-roberta-large

[12] https://github.com/allegro/elasticsearch-analysis-morfologik

[13] https://huggingface.co/sdadas/polish-roberta-large-v2

7 Results and Discussion

KBQA Summarized results are presented in Table 2. For both natural and template-based questions, the inclusion of KG significantly improves accuracy. The overall accuracy is not high, underscoring the challenging nature of the newly introduced PUGG dataset. This complexity highlights its potential as a valuable resource for advancing research and development in the field of KBQA. As expected, reasoning over 1-hop (1H) KG was easier than over 2-hop (2H) KG, reflecting the increased complexity of KG. There is a clear gap in efficacy between natural and template-based questions. That was expected, as template-based questions were designed to be easier. Interestingly, they benefit more from the use of KG. We think that can be caused by their inherent complexity and variability. Moreover, our pipeline for natural questions do not ensure existence of appropriate reasoning paths.

Mode	KG	Retriever	Accuracy
KBQA (natural)			
w/o KG	-	-	0.275
w/ KG	1H	3-hop	0.342
w/ KG	2H	2-hop	0.334
KBQA (template-based)			
w/o KG	-	-	0.210
w/ KG	1H	3-hop	0.674
w/ KG	2H	2-hop	0.669
KBQA (all)			
w/o KG	-	-	0.250
w/ KG	1H	3-hop	0.468
w/ KG	2H	2-hop	0.461

Table 2: Results of the KBQA baselines.

MRC The results achieved by the MRC baselines, as presented in Table 3, suggest that extractive models excel in identifying exact matches within the text. On the other hand, large generative models have demonstrated a capacity to achieve a high degree of general answer overlap, as reflected by their F1 scores. In comparison to the baseline results disclosed in the PoQuAD publication, which reported exact match and F1 scores of 66.22 and 81.39 respectively, the current results suggest that the dataset is a grater challenge for the models.

Model name	Exact Match	F1
herbert-base-cased	42.91	66.41
herbert-large-cased	**46.81**	70.42
plt5-base	22.86	57.63
plt5-large	38.88	**71.52**

Table 3: Results of the MRC baselines.

Model name	NDCG@10	MRR@10	Recall@10	Recall@100
Retriever baselines				
BM25	0.371	0.318	0.549	0.809
silver-retriever-base-v1.1	0.523	0.457	0.733	0.923
mmlw-retrieval-roberta-base	0.645	0.601	0.805	0.925
mmlw-retrieval-roberta-large	0.700	0.653	0.849	0.946
multilingual-e5-base	0.667	0.616	0.828	0.943
multilingual-e5-large	**0.741**	**0.694**	**0.888**	**0.972**
Retriever+Reranker baselines				
BM25+herbert-large-msmarco	0.707	0.677	0.797	0.809
BM25+polish-reranker-base-ranknet	0.701	0.671	0.792	0.809
BM25+polish-reranker-large-ranknet	0.723	0.697	0.802	0.809
multilingual-e5-large+polish-reranker-large-ranknet	**0.813**	**0.770**	**0.942**	**0.972**

Table 4: Results of the IR baselines. The baselines are categorized into two groups: retriever baselines and retrievers with reranking baselines. For the reranking baselines, the top 100 retriever results undergo reranking.

IR The scores presented in Table 4 reveal that the dataset poses a significant challenge for the lexical BM25 approach. The questions have limited lexical overlap, therefore this method is not effective. Nonetheless, current dense retrieval models are exhibiting high performance. Surprisingly, the *mmlw-retrieval-roberta-large* model, despite being currently ranked at the top of PIRB benchmark, still falls behind the *multilingual-e5-large* model. This suggests that the dataset is a valuable resource for assessment and should be included in the PIRB benchmark in the future. The reranker models improved the BM25 rankings significantly, and the combination of dense retriever with a reranker has achieved remarkably high scores across all metrics.

8 Limitations and Future Work

This section outlines the limitations of our study and potential directions for future work. (1) The natural questions are open domain, focused on location and time and are created and answered from the Polish cultural, political, and historical perspective. (2) The pipeline for natural questions may sometimes miss certain answer entities. This is due to the fact that not all answers are always explicitly referenced in the textual answer. (3) Some of the KBQA natural questions might not have corresponding facts in the KG, as our pipeline does not guarantee the existence of an appropriate reasoning path between topic and answer entities. However, as Wikidata is continuously updated and expanded, this limitation may diminish in the future. (4) The questions might contain grammatical imperfections or mental shortcuts, yet remain understandable. (5) Automated annotation with LLM led to variability in the precision of tagged answers in MRC task, due to the absence of specific tagging guidelines.

9 Conclusion

To address the significant gap in resources for low-resource languages, our work introduces the PUGG dataset, the first Polish KBQA dataset, which also encompasses MRC and IR tasks. It consists of natural and template-based factoid questions. The dataset is the outcome of our proposed semi-automated construction pipeline, specifically designed for low-resource environments. Leveraging modern tools as annotation assistants has allowed us to significantly reduce the need for human labor. Additionally, we developed and detailed custom methods, such as for entity linking, which are useful in various contexts. The PUGG dataset, along with our pipeline's comprehensive implementation, insightful findings and detailed statistics provides valuable insights for future research. Furthermore, the evaluation of baseline models on this dataset reveals its challenging nature, underscoring its potential to advance the field and contribute to the development of more robust QA systems.

References

Lukasz Augustyniak, Kamil Tagowski, Albert Sawczyn, Denis Janiak, Roman Bartusiak, Adrian Szymczak, Arkadiusz Janz, Piotr Szymański, Marcin Wątroba, Mikoł aj Morzy, Tomasz Kajdanowicz, and Maciej Piasecki. 2022. This is the way: designing and compiling lepiszcze, a comprehensive nlp benchmark for polish. In *Advances in Neural Information Processing Systems*, volume 35, pages 21805–21818. Curran Associates, Inc.

Jinheon Baek, Alham Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, pages 70–98, Toronto, ON, Canada. Association for Computational Linguistics.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a".

Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, , Roberto Lotufo, and Rodrigo Nogueira. 2021. mmarco: A multilingual version of ms marco passage ranking dataset.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Aleksandra Chrabrowa, Łukasz Dragan, Karol Grzegorczyk, Dariusz Kajtoch, Mikołaj Koszowski, Robert Mroczkowski, and Piotr Rybak. 2022. Evaluation of transfer learning for Polish with a text-to-text model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4374–4394, Marseille, France. European Language Resources Association.

Ruixiang Cui, Rahul Aralikatte, Heather Lent, and Daniel Hershcovich. 2022. Compositional Generalization in Multilingual Semantic Parsing over Wikidata. *Transactions of the Association for Computational Linguistics*, 10:937–955.

Bidyut Das, Mukta Majumder, Santanu Phadikar, and Arif Ahmed Sekh. 2021. Automatic question generation and answer assessment: a survey. *Research and Practice in Technology Enhanced Learning*, 16(1):5.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual Autoregressive Entity Linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

Nan Duan and Duyu Tang. 2018. Overview of the nlpcc 2017 shared task: Open domain chinese question answering. In *Natural Language Processing and Chinese Computing*, pages 954–961. Springer International Publishing.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *The Semantic Web – ISWC 2019*, pages 69–78, Cham. Springer International Publishing.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. In *IEEE Data Eng. Bull.*, volume 40, pages 52–74.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Longquan Jiang and Ricardo Usbeck. 2022. Knowledge graph question answering datasets and their generalizability: Are they enough for future research? In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3209–3218, New York, NY, USA. Association for Computing Machinery.

Lucie-Aimée Kaffee, Russa Biswas, C. Maria Keet, Edlira Kalemi Vakaj, and Gerard de Melo. 2023. Multilingual Knowledge Graphs and Low-Resource Languages: A Review. *Transactions on Graph Data and Knowledge*, 1(1):10:1–10:19.

Vladislav Korablinov and Pavel Braslavski. 2020. Rubq: A russian dataset for question answering over wikidata. In *The Semantic Web – ISWC 2020*, pages 97–110, Cham. Springer International Publishing.

Taja Kuzman, Igor Mozetič, and Nikola Ljubešić. 2023. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. A survey on complex knowledge base question answering: Methods, challenges and solutions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4483–4491. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Michał Marcińczuk, Marcin Oleksy, and Jan Kocoń. 2017. Inforex — a collaborative system for text corpora annotation and analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 473–482, Varna, Bulgaria. INCOMA Ltd.

Michał Marcińczuk, Adam Radziszewski, Maciej Piasecki, Dominik Piasecki, and Marcin Ptak. 2013. Evaluation of baseline information retrieval for Polish open-domain question answering system. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 428–435, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Robert Mroczkowski, Piotr Rybak, Alina Wr ’oblewska, and Ireneusz Gawlik. 2021. HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2017. MS MARCO: A human-generated MAchine reading COmprehension dataset.

Aleksandr Perevalov, Dennis Diefenbach, Ricardo Usbeck, and Andreas Both. 2022. Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers. In *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pages 229–234.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a Benchmark for Knowledge Intensive Language Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Ronak Pradeep, Haonan Chen, Lingwei Gu, Manveer Singh Tamber, and Jimmy Lin. 2023. Pygaggle: A gaggle of resources for open-domain question answering. In *Advances in Information Retrieval*, pages 148–162, Cham. Springer Nature Switzerland.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Henry Rosales-Méndez, Barbara Poblete, and Aidan Hogan. 2018. What should entity linking link? In *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management, Cali, Colombia, May 21-25, 2018*, volume 2100 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Piotr Rybak. 2023. MAUPQA: Massive automatically-created Polish question answering dataset. In *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*, pages 11–16, Dubrovnik, Croatia. Association for Computational Linguistics.

Piotr Rybak and Maciej Ogrodniczuk. 2023. Silverretriever: Advancing neural passage retrieval for polish question answering.

Piotr Rybak, Piotr Przybyła, and Maciej Ogrodniczuk. 2022. Improving question answering performance through manual annotation: Costs, benefits and strategies.

Ivan Rybin, Vladislav Korablinov, Pavel Efimov, and Pavel Braslavski. 2021. Rubq 2.0: An innovated russian question answering dataset. In *The Semantic Web*, pages 532–547, Cham. Springer International Publishing.

Nadine Steinmetz and Kai-Uwe Sattler. 2021. What is in the kgqa benchmark datasets? survey on challenges in datasets for question answering on knowledge graphs. *Journal on Data Semantics*, 10(3):241–265.

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for QA evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 562–572, Austin, Texas. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Ryszard Tuora, Aleksandra Zwierzchowska, Natalia Zawadzka-Paluektau, Cezary Klamra, and Łukasz Kobyliński. 2023. Poquad - the polish question answering dataset - description and analysis. In *Proceedings of the 12th Knowledge Capture Conference 2023*, K-CAP '23, page 105–113, New York, NY, USA. Association for Computing Machinery.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Konrad Wojtasik, Vadim Shishkin, Kacper Wołowiec, Arkadiusz Janz, and Maciej Piasecki. 2023. Beir-pl: Zero shot information retrieval benchmark for the polish language.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. Zero-shot entity linking with dense entity retrieval. In *EMNLP*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Łukasz Kobyliński, Maciej Ogrodniczuk, Piotr Rybak, Piotr Przybyła, Piotr Pęzik, Agnieszka Mikołajczyk, Wojciech Janowski, Michał Marcińczuk, and Aleksander Smywiński-Pohl. 2023. Poleval 2022/23 challenge tasks and results. In *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*, volume 35 of *Annals of Computer Science and Information Systems*, page 1243–1250. IEEE.

A Textual Answers Tagging

The designed prompt is presented in Table 5. The annotated spans were extracted from the LLM's responses using lemmatization and longest common sequence analysis.

B Topic Entity Linking

The designed entity linking method primarily relies on Wikipedia search engine, title similarity, and information about the *neighborhood of the question*.

The **Wikipedia search engine** is accessed via the MediaWiki API [14]. This search system identifies page titles or content that match a given textual query. **Title similarity** is measured by assessing the similarity of provided texts, utilizing both the longest common sequence and the longest common prefix approaches. To construct the **neighborhood of the question**, we retrieved Wikipedia pages from the top 10 Google search results, and then extracted the first five links from each of these articles. These results are then used to determine whether the entity found by the algorithm belongs to such a neighborhood. It is important to note that, in this context, *the neighborhood of the question* is not associated with the KG.

As the output, we expect four types of entities: exact entities, neighborhood entities, named entities, and combined entities. Detailed information on this process can be found in the pseudocode provided in Algorithm 1.

Algorithm 1 Entity Linking Method

Input:
Q - input question.
Constants:
$L \leftarrow$ [noun, adjective, proper noun, unknown]
$T \leftarrow$ tokenize(Q)
$N \leftarrow$ named_entities(Q)
Output:
E_{exact} - set of entities closely matching the title of Wikipedia pages
E_{nbhd} - set of entities not precisely matching Wikipedia titles but belonging to the question neighborhood
E_{named} - set of named entities belonging to the question neighborhood
E_{comb} - set of entities formed by combining two or more words

Algorithm:

```
for each t ∈ T do
    if pos(t) ∈ L then
        res ← search_wikipedia(t)
        l ← lemma(t)
        E_exact ← high_similarity(res, l)
for each n ∈ N do
    res ← search_wikipedia(n)
    E_named ← in_neighborhood(res)
for each t ∈ T do
    if pos(t) in L then
        res ← search_wikipedia(n)
        E_nbhd ← in_neighborhood(res)
for each t ∈ T do
    if pos(t) == 'noun' then
        R ← get_nouns(children(t))
        A ← get_adjectives(children(t))
        R_q ← R × [t]
        A_q ← A × [t]
        for each q ∈ R_q ∪ A_q do
            res ← search_wikipedia(q)
            E_comb ← in_neighborhood(res)
```

C Human Verification

All annotators were employed in Poland and fluent in Polish. They are familiar with the Polish culture and social context. During verification we ensured that the data does not contain any private data or offensive content.

C.1 First Stage

To ensure high-quality data, the annotation team included both annotators and a super-annotator. he process involved: (1) initial guideline preparation (2) full review of annotator decisions reviewed by a super-annotator and (3) targeted review of problematic examples by the super-annotator. This process refined the guidelines and focused on resolving ambiguities in annotations. Examples with improperly formulated questions or lacking information for accurate answers were rejected, especially those with

[14] https://www.mediawiki.org/wiki/API:Search

Textual Answer Tagging Prompt

pl:

```
User:
    Cytat to dokładna kopia tekstu słowo w słowo. Podam tobie tekst i pytanie.
        ↪ Twoim zadaniem będzie znalezienie w tekście DOKŁADNEGO cytatu. Cytat
        ↪ musi być najbliższy odpowiedzi lub taki, który może być potencjalną
        ↪ odpowiedzią. Musi to być najkrótszy możliwy cytat w tekście. Nie należy
        ↪  zmieniać żadnych słów. Nie odmieniaj słów. Nie dodawaj żadnych
        ↪ dodatkowych słów, abym mógł go skopiować. Więc proszę nie zmieniać
        ↪ nawet kapitalizacji.
Assistant:
    Jasne, przytoczę tylko dokładny cytat. Nie będę dodawał żadnych słów. Nie będę
        ↪  zmieniał słów. Nie będę zmieniał przypadków słów. Nie zmienię wielkoś
        ↪ ci liter.
User:
    Context: "[START]Elżbieta II (; ur. 21 kwietnia 1926 w Londynie, zm. 8 wrześ
        ↪ nia 2022 w Balmoral)  królowa Zjednoczonego Królestwa Wielkiej Brytanii
        ↪  i Irlandii Północnej z dynastii Windsorów od 6 lutego 1952 (koronowana
        ↪  2 czerwca 1953) do 8 września 2022.[END]"
    Question: w którym roku urodziła się królowa elżbieta ii?
    A: "
Assistant:
    21 kwietnia 1926"
User:
    Context: "[START]{context}[END]"
    Question: {question}
    A: "
```

en:

```
User:
    A quote is an exact copy of the text word for word. I will give you the text
        ↪ and the question. Your task will be to find the EXACT quote in the text
        ↪ . The quote must be the closest to the answer or one that could be a
        ↪ potential answer. It must be the shortest possible quote in the text.
        ↪ Do not change any words. Do not inflect words. Do not add any
        ↪ additional words so that I can copy it. So please don't even change the
        ↪  capitalization.
Assistant:
    Sure, I will just quote the exact quote. I will not add any words. I will not
        ↪ change the words. I will not change the word cases. I will not change
        ↪ the case of the letters.
User:
    Context: "[START]Elizabeth II (; born April 21, 1926 in London, died September
        ↪  8, 2022 in Balmoral) - Queen of the United Kingdom of Great Britain
        ↪ and Northern Ireland of the Windsor dynasty from February 6, 1952 (
        ↪ crowned June 2, 1953) to September 8, 2022.[END]"
    Question: in what year was Queen Elizabeth ii born?
    A: "
Assistant:
    April 21, 1926"
User:
    Context: "[START]{context}[END]"
    Question: {question}
    A: "
```

Table 5: Textual answer tagging prompt.

significant grammatical or lexical errors that made them incomprehensible. Technically, this step involved flagging documents in the Inforex system (Marcińczuk et al., 2017), with the following set of flags: (1) *correct*: indicates both the question and answer are correct in the passage. (2) *incorrect question*: indicates the question is formulated incorrectly. (3) *incorrect passage*: indicates the passage does not answer the question. (4) *incorrect fragment*: indicates the answer is located elsewhere in the passage.

C.2 Second Stage

This stage was carried out by two annotators. To facilitate a consistent and measurable approach, we separated out 10% of the examples as common for both annotators, while the rest were individually assigned. These shared examples served as a basis

for calculating annotation metrics and ensuring reliability and consistency in the annotation process. Annotating the correct answers was a straightforward task. However, the annotation of topic entities presented more complexity. As Rosales-Méndez et al. (2018) have pointed out, there is no consensus on the concept of an entity and what entity linking should link to, as it varies greatly depending on the application. Due to the absence of universally acknowledged guidelines, we defined a topic entity as a source entity from which the reasoning method can begin its process. In cases where annotators were uncertain about either answer or topic entities, the problematic examples were rejected to maintain the quality of the dataset. The entire second stage of annotation process was carried out using a spreadsheet application. During the annotation of answer entities and topic entities, we achieved Cohen's kappa scores of 0.785 and 0.675, respectively, indicating high inter-annotator agreement.

D Detailed statistics

The detailed statistics of the pipeline steps are presented in Table 6.

Data	#
Natural	
Questions from existing QA datasets	17019
Prefixes	33467
Formulated questions	90666
Wikipedia articles	18055
Textual answer tagging input	31780
Successfully parsed tags	19296
Correct passages	10751
Correct textual answers	8772
Questions after answer entity ver.	3832
Questions after topic entity ver.	3509
KBQA examples	3471
MRC examples	8702
IR examples	10751
Template-based	
Executed templates	14400
After filtering	4231
After verification	2122

Table 6: Detailed statistics of the executed pipelines: natural and template-based. Note that the final dataset examples differ from those in the corresponding previous steps due to several manual interventions. These include deduplication, where we identified and removed duplicates, and manual entity linking.

E Template-based KBQA

E.1 Templates

We have developed 8 templates for schematic question creation, detailed in Table 7. We distinguish the following three general techniques.

N-hop templates are utilized to retrieve information by traversing N relations from the given entity.

Reverse N-hop templates function similarly, but involve traversing in the reverse direction.

The **Entity Mask** technique enriches questions by referring to the answer without direct mention. For example, instead of naming "Ludwig van Beethoven", we might use "composer".

E.2 Paraphrasing and Inflection Prompts

Table 8 presents the prompts utilized for inflecting and paraphrasing questions constructed using the natural language templates.

E.3 Human Verification

Inflected and paraphrased questions were verified using with the following set of annotation flags: *correct*, *incorrect*, and *resembling*.

Correct implies the semantic meaning of the processed question remains unchanged compared to the original. **Incorrect** flags a change in semantic meaning. For instance, the original question *'Who is the creator of the web browser?'* paraphrased as *'What material is the web browser created of?'* illustrates this change. It's also worth mentioning that incorrect questions often involve the reversal of properties: *Whose doctoral supervisor is Max Perutz?* was paraphrased as *Who is Max Perutz's doctoral supervisor?*. The fact that LLMs may struggle to understand reverse connections, was also highlighted in a paper by Berglund et al. (2023).

During annotation, we noticed some question patterns frequently repeated in specific templates like one-hop templates. We labeled these as **resembling** and excluded them from the final dataset. For example, 'Where was X born?', was common due to the 'place of birth' being a prevalent relation for people on Wikidata.

The statistics of verification are presented in Table 9.

F KBQA Baseline Prompts

We adapted the LLM prompt from KAPING (Baek et al., 2023) by translating and slightly modify-

Template name		Natural Language Template	Examples	SPARQL Template
One-hop	pl	Jakie ... ma ...?	Q: Jakie {imię} ma {Ludwig van Beethoven}? A: {Ludwig}.	SELECT ?answerEntity WHERE {{ wd:Q255 wdt:P735 ?answerEntity. }}
	en	What is the ... of ...?	Q: What is the {given name} of {Ludwig van Beethoven}? A: {Ludwig}.	
One-hop with entity mask	pl	Jak nazywał się ..., które jest ... ...?	Q: Jak nazywał się {metropolia}, które jest {miejsce śmierci} {Ludwig van Beethoven}? A: {Wiedeń}.	SELECT ?answerEntity WHERE {{ wd:Q255 wdt:P20 ?answerEntity. ?answerEntity wdt:P31 wd:Q200250. }}
	en	What was the name of the ..., which is the ... of ...?	Q: What was the name of the {metropolis}, which is the {place of death} of {Ludwig van Beethoven}? A: {Vienna}.	
Two-hop	pl	Jakie ... ma ... ...?	Q: Jakie {obywatelstwo} ma {matka} {Ludwig van Beethoven}? A: {Niemcy}.	SELECT ?answerEntity WHERE {{ wd:Q255 wdt:P25 ?relatedEntity. ?relatedEntity wdt:P27 ?answerEntity. }}
	en	What is the ... of ...'s ...?	Q: What is the {country of citizenship} of {Ludwig van Beethoven}'s {mother}? A: {Germany}.	
Reverse one-hop	pl	Czyim ... jest ...?	Q: Czyim {student} jest {Carl Czerny}? A: {Ludwig van Beethoven, Antonio Salieri}.	SELECT ?answerEntity WHERE {{ ?answerEntity wdt:P802 wd:Q215333. }}
	en	Whose ... is ...?	Q: Whose {student} is {Carl Czerny}? A: {Ludwig van Beethoven, Antonio Salieri}.	
Reverse one-hop with mask entity	pl	Jak nazywał się ..., którego ... jest ...?	Q: Jak nazywał się {kompozytor}, którego {rodzeństwo} jest {Kaspar Anton Karl van Beethoven}? A: {Ludwig van Beethoven}.	SELECT ?answerEntity WHERE {{ ?answerEntity wdt:P3373 wd:Q6374627. ?answerEntity wdt:P106 wd:Q36834. }}
	en	What was the name of the ... whose ... is ...?	Q: What was the name of the {composer} whose {sibling} is {Kaspar Anton Karl van Beethoven}? A: {Ludwig van Beethoven}.	
Reverse two-hop	pl	Czyim ... jest ..., a ... jest ...?	Q: Czyim {student} jest {Ferdinand Ries}, a {nauczyciel} jest {Joseph Haydn}? A: {Ludwig van Beethoven}.	SELECT ?answerEntity WHERE {{ ?answerEntity wdt:P802 wd:Q213558. ?answerEntity wdt:P1066 wd:Q7349. }}
	en	Whose ... is ..., and ... is ...?	Q: Whose {student} is {Ferdinand Ries}, and {teacher} is {Joseph Haydn}? A: {Ludwig van Beethoven}.	
Reverse two-hop with mask entity	pl	Jak nazywał się ..., którego ... jest ..., a którego ... jest ...?	Q: Jak nazywał się {kompozytor}, którego {przyczyna śmierci} jest {marskość wątroby}, a którego {miejsce śmierci} jest {Wiedeń}? A: {Ludwig van Beethoven}.	SELECT ?answerEntity WHERE {{ ?answerEntity wdt:P509 wd:Q147778. ?answerEntity wdt:P20 wd:Q1741. ?answerEntity wdt:P106 wd:Q36834. }}
	en	What was the name of the ... whose ... is ... and whose ... is ...?	Q: What was the name of the {composer} whose {cause of death} is {cirrhosis of the liver}, and whose {place of death} is {Vienna}? A: {Ludwig van Beethoven}.	
Mixed	pl	Jakie ... ma ..., którego ... jest ...?	Q: Jakie {miejsce urodzenia} ma {kompozytor}, którego {ojcem} jest {Johann van Beethoven}? A: {Bonn}.	SELECT ?answerEntity WHERE {{ ?relatedEntity wdt:P106 wd:Q36834. ?relatedEntity wdt:P22 wd:Q2153541. ?relatedEntity wdt:P19 ?answerEntity. }}
	en	What is the ... of the ... whose ... is ...?	Q: What is the {place of birth} of the {composer} whose {father} is {Johann van Beethoven}? A: {Bonn}.	

Table 7: The question templates used for template-based questions.

ing it to emphasize the need for listing entities in their non-inflected form. The adapted prompt is presented in Table 10.

Inflection Prompt

pl:

```
User:
      Zmień błędne końcówki wyrazów w pytaniu. Pamiętaj, że nie wolno zmieniać
          ↪ podstaw słów, zastępować ich synonimami ani dodawać nowych. Nie można
          ↪ zmieniać kolejności słów.
Assistant:
      Jasne, poprawię błędne końcówki wyrazów w pytaniu. Nie będę zmieniał kolejnoś
          ↪ ci słów. Nie będę dodawał nowych słów. Nie będę zastępował synonimami.
User:
      "Czyim dzieci jest Maria Gorecka?"
Assistant:
      "Czyim dzieckiem jest Maria Gorecka?"
User:
      "Jak nazywał się gmina miejska w Niemczech, który jest miejsce pobytu Adam
          ↪ Mickiewicz?"
Assistant:
      "Jak nazywała się gmina miejska w Niemczech, która była miejscem pobytu Adama
          ↪ Mickiewicza?"
User:
      "{question}"
```

en:

```
User:
      Change the incorrect word endings in the question. Remember not to change the
          ↪ base words, replace them with synonyms, or add new ones. You cannot
          ↪ change the word order.
Assistant:
      Sure, I will correct the incorrect word endings in the question. I will not
          ↪ change the word order. I will not add new words. I will not replace
          ↪ them with synonyms.
User:
      "Whose children is Maria Gorecka?"
Assistant:
      "Whose child is Maria Gorecka?"
User:
      "What was the name of the urban municipality in Germany, which is the
          ↪ residence of Adam Mickiewicz?"
Assistant:
      "What was the name of the urban municipality in Germany, which was the
          ↪ residence of Adam Mickiewicz?"
User:
      "{question}"
```

Paraphrasing Prompt

pl:

```
User:
      Proszę, przeformułuj następujące pytanie, zachowując jego sens.
Assistant:
      Jasne, zrobię to, nie zmieniając sensu pytania.
User:
      "Czyim dzieckiem jest Maria Gorecka?"
Assistant:
      "Kim są rodzice Marii Goreckiej?"
User:
      "{question}"
```

en:

```
User:
      Please, paraphrase the following question while maintaining its meaning.
Assistant:
      Sure, I'll do that without changing the question's meaning.
User:
      "Whose child is Maria Gorecka?"
Assistant:
      "Who are the parents of Maria Gorecka?"
User:
      "{question}"
```

Table 8: Inflection and paraphrasing prompts used for template-based KBQA.

Template name	Correct	Incorrect	Resembling
Reverse One Hop	307	176	0
Reverse One Hop With Mask	220	312	0
Mixed	231	224	0
Reverse Two Hop With Mask	167	275	34
Reverse Two Hop	398	88	0
One Hop	137	393	89
Two Hop	301	290	0
One Hop With Mask	185	335	69

Table 9: The number of correct, incorrect, and resembling questions according to the manual verification for template-based questions.

KBQA Baseline Prompt (w/o KG)	
pl:	`Pytanie: {question}` `Encje które są odpowiedzią:`
en:	`Question: {question}` `Entities which are the answer:`
KBQA Baseline Prompt (w/ KG)	

pl:	`Poniżej znajdują się fakty w postaci trójek grafu wiedzy w formacie (encja, relacja, encja), mające znaczenie do udzielenia odpowiedzi na pytanie.`
	`{triples}`
	`Pytanie: {question}`
	`Encje które są odpowiedzią:`

en:	`Below are facts in the form of knowledge graph triples in the format (entity, relation, entity), relevant to answering the question.`
	`{triples}`
	`Question: {question}`
	`Entities which are the answer:`

Table 10: KBQA baseline Prompts.