# Answering Polish Trivia Questions with the Help of Dense Passage Retriever

**Aleksander Smywiński-Pohl, Dmytro Zhylko**
(AGH University of Science and Technology / Enelpol)

**Krzysztof Wróbel** (Enelpol / Jagiellonian University / AGH University of Science and Technology)

**Magdalena Król** (AGH University of Science and Technology)

**Abstract**

This paper discusses the problem of Question answering using Dense Passage Retriever in Task 4 during 2021 edition of PolEval. Our goal was to show the process of automatic answering trivia questions using language models and Wikipedia database. The best solution created by the authors utilized Dense Passage Retrieval approach for extractive question answering combined with Natural Language Inference for boolean questions. The training data for document retrieval and extractive question answering were obtained by employing distant supervision. The obtained solution reached 50.96% accurracy giving second place in the competition.

**Keywords**

question answering, dense passage retrieval, Polish language model, trivia questions, Wikipedia

## 1. Introduction

Question answering (QA) is one of the most interesting tasks within Natural Language Processing (NLP). Since Alan Turing has defined his famous test for intelligence (Turing 2009), question answering remains one of the most challenging issues related to human and artificial intelligence. However, a system able to answer any question a human can answer easily remains an unfinished goal. To achieve this goal a growing number of QA datasets is made available: SQuAD (Rajpurkar et al. 2016), WikiQA (Yang et al. 2015), CNN/DailyMail (Hermann et al. 2015), MS MARCO (Bajaj et al. 2016), TriviaQA (Joshi et al. 2017), Natural

questions (Kwiatkowski et al. 2019), to name just the most popular of them. Yet, as with many NLP resources, these datasets are available only for English. In Polish, there are only a few datasets related to QA: "Czy wiesz"[1], Polish Legal Question Answering Dataset (LQuAD-PL, to appear) and the dataset which was made available during the PolEval 2021 competition. The last dataset is an interesting one since it includes only the questions and the answers as the training data. Most of the remaining datasets contain excerpts relevant for answering the question.

There are several of competing approaches related to question answering. They might be roughly categorized into: extractive QA, abstractive QA and closed-book QA. In an end-to-end system the first two approaches are usually supplemented by a selective QA module, frequently called a retriever. A typical complete QA pipeline for the first two approaches is as follows: from a large body of documents (e.g. passages from Wikipedia) the retriever selects a much smaller subset. In the next step the reader inspects them and either selects a span of consecutive tokens (for extractive QA) or forms an answer, using the selected passages as input in a sequence-to-sequence setting (for abstractive QA). Closed-book QA is the most recent trend, which is made possible thanks to the availability of extremely large language models, such as GPT-3 (Brown et al. 2020), T5 (Raffel et al. 2020) and mT5 (Xue et al. 2021). These generative solutions are able to answer questions using only the world knowledge preserved in the model itself. Although they do not achieve scores as high as the remaining approaches, they reflect the human ability of answering a question without consulting any external knowledge sources.

The experiments conducted by the authors during the PolEval 2021 competition were designed to explore the first and the last approach. The majority of them employed the extractive QA paradigm, which is implemented in DPR (Dense Passage Retriever; Karpukhin et al. 2020). Yet, for yes/no questions this approach seems to be invalid. For this reason, we employed an approach based on Natural Language Inference. We also conducted some experiments following the closed book approach, which utilizes a sequence-to-sequence architecture.

## 2. Data

The data provided by the organizers included questions and answers from the Polish TV competition called "Jeden z dziesięciu" (translated literally as "One out of ten", a Polish version of the British TV show "Fifteen to one"). It included three subsets: dev-0 (1000 question and answer pairs), test-A (2500 question and answer pairs; the answers were provided during the last stage of the competition) and test-B (2500 question and answer pairs; the answers were provided after the competition finished).

According to the rules of the competition, both dev-0 and test-A could be used to train the QA model. We have decided to use test-A as a base for the training set and dev-0 as a base for the validation set.

[1] `http://nlp.pwr.wroc.pl/en/tools-and-resources/resources/czy-wiesz-question-answering-dataset`

A sample QA pair from the dataset is given below:

| | | |
|---|---|---|
| (PL) | Urodę której części twarzy podkreśla mascara? | rzęs |
| (EN) | Which part of the face is highlighted by mascara? | eyelashes |

Polish is an inflected language. The organizers decided to use that feature and required that the answer is correctly inflected for number, case and possibly gender. This is not fully visible in the translation, since English does not inflect for case, but the answer "rzęs" is a plural form in genitive, while a typical entry in a dictionary would be "rzęsa" which is a singular form in the nominative case. Matching the answer exactly would be impossible for the extractive approach for most of the examples, making that approach impractical. But the authors of the competition decided to use a different scoring function. They compute a Levenshtein distance between the provided answer and the reference answer. If the distance divided by the length of the reference answer was smaller than 0.5, the answer was treated as correct. Since the inflected part of the word is usually shorter than half of the word, for most of the answers a nominal case or some other case appearing directly in the text was enough to treat the answer as correct, assuming the proper word or words were selected.

Although this approach was feasible for scoring the final results, it was not helpful for training the QA models (both the retriever and the reader). For the retriever, it was necessary to decide whether a given text snippet includes the answer. For the reader, it was necessary to precisely indicate the consecutive sequence of tokens, which are the answer. To resolve these issues, a lemmatizer from the KRNNT library was employed (Wróbel 2017) – when matching the snippet for the answer the literal and the base forms of the tokens in the answer were matched against the literal and the base forms of the tokens in the snippet. The answers were also pre-processed in order to remove the leading preposition since some of them included such a term (e.g. "we Włoszech" – "in Italy").

# 3. Method

## 3.1. Distant supervision

As explained in Section 2, the dataset didn't include text snippets that could be used to train the retriever and the reader models directly. A distant supervision approach was employed to resolve this issue. The authors decided to use only Polish Wikipedia as the reference corpus with the snippets. The Wikipedia dump was processed using WikiExtract library[2]. The authors used a dump of Wikipedia from 2019, which is available on the PolEval website[3] (Smywiński-Pohl 2019). The WikiExtract library removes elements such as navigation structure and so-called infoboxes, leaving a clean Wikipedia text, preserving the Wikipedia links in a separate file[4]. The only additional step that was necessary was splitting the text into snippets. Since the text was already sentence-split, it was possible to create snippets containing whole

[2] `https://github.com/cycloped-io/wikiextract`

[3] `http://2019.poleval.pl/index.php/tasks/task3`

[4] The links were not utilized in the experiment.

sentences. The authors decided to use snippets containing up to 90 words, allowing for longer sequences to preserve full sentences.[5]

To select a set of candidate snippets for a given question, we have used a dense passage representation provided by DPR. The model training followed the same procedure as described in Section 3.2 (HerBERT-base was used as the base model), but the training data included only "Czy wiesz" and LQuAD-PL datasets. The model was used to select 100 answer candidates for each question. The candidates were lemmatized and matched against the answers. The longest common subsequence algorithm was used to match the parts of the answer. The longest subsequence of consecutive tokens was returned as the result, allowing for gaps up to 2 tokens. It was required that at least 80% of the tokens from the answer have to be present in the candidate snippet, to treat it as a positive example. Otherwise, the snippet was treated as a negative example. The positive candidate examples were inspected in the order provided by the DPR retriever, and the snippet with the largest number of matched tokens was selected as the reference, positive example. In the case of a tie, a snippet higher on the DPR rank list was selected. The snippets prepared in that way served as training examples for both the retriever and the reader. The negative examples were treated as the hard negative examples in the DPR parlance, since they were similar in content to the positive example. In the case of the reader, the matched sequence served as the reference answer.

The retriever dataset needs one additional remark: the training set included a single positive example selected according to the above rules and a number of hard negative examples. Yet, this scheme may not be applied to the validation dataset! This might be obvious, since the validation dataset has to reflect the test dataset, however, the usual practice allows to select the validation dataset as the subset of the given training set. This mistake was in fact made by the authors, leading to detrimental results in a number of experiments.

## 3.2. Retriever

We have used Dense Passage Retriever (DPR; Karpukhin et al. 2020) as the retriever module. We have employed HerBERT-base[6] and HerBERT-large[7] as the starting language models (Mroczkowski et al. 2021). They were trained on "Czy wiesz", LQuAD-PL and the training subset of the dataset described in the Section 3.1. The top-1 accuracy of the models was 63.6% and 68.0% respectively. The best models were selected according to the accuracy on the validation set. All models were trained for 20 epochs with the best checkpoints on 15-th and 16-th epoch for smaller and larger model accordingly. FAISS (Johnson et al. 2021) was used to store the dense representation of the snippets. Since we didn't have to optimize for retrieval speed, we have used the flat access mode. The server required approx. 40GB of RAM to store all Wikipedia passages (approx. 3.750 million passages).

[5] This value was somehow arbitrary, but corresponds roughly to 100 tokens-length reported in the DPR paper (Karpukhin et al. 2020).

[6] `https://huggingface.co/allegro/herbert-base-cased`

[7] `https://huggingface.co/allegro/herbert-large-cased`

## 3.3. Reader

The reader model we used comes also from DPR, which combines a re-ranker with a model able to select the span of tokens containing the answer for the question. DPR uses a simple approach regarding the selection of the best answer: at first, it sorts the passages provided by the retriever and selects the first passage; then, it selects the answer tokens in that passage according to the reader model. It should be noted that the model used to re-rank the passages has access to full attention between the question and the passage, while the retriever optimizes the dot-product between the dense representations of the question and the answer. Yet, both models are trained using a similar approach – the negative examples are the passages provided explicitly in the training set and in-batch negatives, i.e. the other passages that are present in the batch. This allows for more efficient training and provides one important hyperparameter, i.e. the number of hard negatives used for training. This value combined with the batch size determines the total number of negatives for a given positive example and directly impacts the memory requirements of this method.

## 3.4. Boolean questions

Boolean questions, such as "Czy w państwach starożytnych powoływani byli posłowie i poselstwa?" ("Were envoys and legations called in the ancient countries?") cannot be answered using the extractive approach. Such questions were identified by a regular expression, matching "Czy" ("whether") at the beginning of the questions and excluding this word in its remaining part. The exclusion was necessary, since questions requiring selection of one of the provided options, such as "Czy sombrero to kapelusz, danie czy taniec?" ("Is sombrero a hat, a dish or a dance?") also have "Czy" at the beginning.

The approach applied to the boolean questions was the following: all passages returned by DPR were selected as the first sentence for Natural Language Inference task, while the question that yielded these passages was treated as the second sentence. If the answer to the question was "yes", the pairs of sentences were marked as entailment – if it was "no", they were marked as a contradiction. A separate set of questions with random passages were marked as neutral for the given question. HerBERT-large model was fine-tuned on the CDSCorpus (Krasnowska-Kieraś and Wróblewska 2019, Wróblewska and Krasnowska-Kieraś 2017) and then further tuned on the above dataset. During inference for the question and the retrieved passages the model was used to determine the relation between them. Pairs with neutral relations were discarded. The answer was selected as a majority vote between entailment and contradiction, with entailment ("yes") winning in the case of a tie.

## 3.5. Closed-book QA

A closed-book method was also tested on the provided dataset. This approach does not require a set of passages to read the answer, since it employs a sequence-to-sequence paradigm, where the question is the input sequence, while the answer is the output sequence. The only knowledge available is the model itself. This method may yield any reasonable results

only if the pre-trained model is very large and it predicts the next token (i.e. it is a *classical* language model). The number of such models supporting Polish is very small: PapuGaPT-2 (Wojczulis and Kłeczek 2021), plT5-large[8] and mT5 models family (Xue et al. 2021). Since the training of such models is time-consuming, we have decided to test only the last family, aiming at using the largest models: mT5-large, mT5-xl and mT5-xxl. The last model is especially problematic, since its size is larger than the VRAM capacity of the most popular V100 GPUs. Transformers library provides experimental support for running that model on multiple cards (via parallel model feature[9]) and even on one card (e.g. Deepspeed (Rasley et al. 2020) provides a CPU-offload feature which is integrated into Transformers[10]).

Yet, these features are experimental and setting up a working pipeline is currently a cumbersome experience. A (costly) alternative is usage of Google TPU-pods which provide direct and well-tested support for very large models (mT5 was trained on Google TPU platform). For that reason we have decided to test the closed-book approach using the Google infrastructure.

## 4. Experiments

Approximately 30 experiments were conducted in order to estimate the impact of the different features and parameters on the final results:

— using different strategies regarding the title of the article,

— using different snippet searching strategies (sparse vs. dense passage retrieval),

— using a different number of negative contexts,

— using different sizes of the model,

— using different training data,

— using different strategies for the validation dataset.

Since the research was driven by the competition, none of the above features and strategies were tested systematically. Still, some observations may be drawn based on the partial results.

The most important observation was a correct preparation of the validation dataset – in many experiments this dataset was taken directly from the training dataset, while that dataset included only one snippet with the correct answer. The experiments conducted in that setting showed that with the growing number of candidate snippets, the result deteriorated. Changing the validation dataset to include all snippets returned by the retriever showed that it is necessary to include 80 snippets to obtain the best result. Fixing this error resulted in test accuracy jumping from 24.84% to 46.84% (22 pp.).

The second important observation regarded the size of the model used to train the reader. By substituting HerBERT-base with HerBERT-large, we have obtained almost 11 pp. improvement on the validation dataset – the best result for the *base* model was 49.62%, while the best

[8] `https://huggingface.co/allegro/plt5-large`

[9] `https://github.com/huggingface/transformers/pull/7772`

[10] `https://huggingface.co/transformers/main_classes/deepspeed.html`

result for the *large* model was 60.61%. This result is consistent with the observation that the winning submission used mT5-xxl model, which is the largest pre-trained model including support for Polish.

The third important observation was assessing the impact of the number of negative examples on the results on the validation dataset. Changing the default value of 3 negative examples to 10 gave more than 11 pp. improvement (21.22% to 32.88%), while 15 negative examples gave further 17 pp. improvement (32.88% to 49.62%). So the correct number of negative examples has a great impact on the final result.

## 5. Results

The best result obtained by the team was 50.96 giving the second place on the leaderboard (equally with Piotr Rybak's submission) – cf. Table 1.

Table 1: The final results of the competition

| **Contestant name** | **Result** |
|---|---|
| Mateusz Piotrowski | 71.68 |
| Aleksander Smywiński-Pohl | **50.96** |
| Piotr Rybak | 50.96 |
| Darek Kłeczek | 46.44 |
| Karol Gawron | 36.12 |
| BI Insight | 0.96 |

That result was obtained combining the following elements:

1. The DPR retriever used to generate the passages was trained on "Czy wiesz" dataset, LQuAD-PL and the distantly-supervised training part of the PolEval Task-4 (i.e. the data described in Section 3.1).
2. The DPR retriever model was based on HerBERT-base.
3. Top-80 passages were used during inference.
4. The DPR reader model was based on HerBERT-large and it was trained only on the training part of the PolEval Task-4.
5. There were 12 negative contexts used during training.
6. The batch size was limited to 1.
7. The number of epochs/steps was 1.92.
8. Boolean questions were answered according the the NLI model, trained on CDSCorpus and the data from Section 3.1 and utilized HerBERT-large.

Regarding the closed-book approach, the best result on the validation dataset was 19.60%. It was obtained using mT5-xl model trained for 1000 steps.[11] Due to high cost of the TPUs we have not run the inference on the test dataset.

## 6. Acknowledgments

This work was supported by the Polish National Centre for Research and Development – LIDER Program under Grant LIDER/27/0164/L-8/16/NCBR/2017 titled "Lemkin – intelligent legal information system" and in part by the PLGrid Infrastructure.

## References

Bajaj P., Campos D., Craswell N., Deng L., Gao J., Liu X., Majumder R., McNamara A., Mitra B., Nguyen T., Rosenberg M., Song X., Stoica A., Tiwary S. and Wang T. (2016). *MS MARCO: A Human Generated MAchine Reading COmprehension Dataset*. arXiv:1611.09268.

Brown T. B., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D. M., Wu J., Winter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I. and Amodei D. (2020). *Language Models are Few-Shot Learners*. In Larochelle H., Ranzato M., Hadsell R., Balcan M. and Lin H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, vol. 33, pp. 1877–1901. Curran Associates, Inc.

Hermann K. M., Kočiský T., Grefenstette E., Espeholt L., Kay W., Suleyman M. and Blunsom P. (2015). *Teaching machines to read and comprehend*. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS'15). Volume 1*, pp. 1693–1701, Cambridge, MA, USA. MIT Press.

Johnson J., Douze M. and Jégou H. (2021). *Billion-Scale Similarity Search with GPUs*. „IEEE Transactions on Big Data", 7, p. 535–547.

Joshi M., Choi E., Weld D. and Zettlemoyer L. (2017). *TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension*. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Karpukhin V., Oguz B., Min S., Lewis P., Wu L., Edunov S., Chen D. and Yih W.-t. (2020). *Dense Passage Retrieval for Open-Domain Question Answering*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*, pp. 6769–6781, Online. Association for Computational Linguistics.

[11] Finally due to financial and time constraints we haven't decided to use the mT5-xxl model.

Krasnowska-Kieraś K. and Wróblewska A. (2019). *Empirical Linguistic Study of Sentence Embeddings*. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5729–5739, Florence, Italy. Association for Computational Linguistics.

Kwiatkowski T., Palomaki J., Redfield O., Collins M., Parikh A., Alberti C., Epstein D., Polosukhin I., Devlin J., Lee K., Toutanova K., Jones L., Kelcey M., Chang M.-W., Dai A. M., Uszkoreit J., Le Q. and Petrov S. (2019). *Natural Questions: A Benchmark for Question Answering Research*. „Transactions of the Association for Computational Linguistics", 7, p. 452–466.

Mroczkowski R., Rybak P., Wróblewska A. and Gawlik I. (2021). *HerBERT: Efficiently pretrained transformer-based language model for Polish*. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pp. 1–10, Kiyv, Ukraine. Association for Computational Linguistics.

Raffel C., Shazeer N., Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W. and Liu P. J. (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. „Journal of Machine Learning Research", 21(140), p. 1–67.

Rajpurkar P., Zhang J., Lopyrev K. and Liang P. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rasley J., Rajbhandari S., Ruwase O. and He Y. (2020). *Deepspeed: System Optimizations Enable Training Deep Learning Models with over 100 Billion Parameters*. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3505–3506.

Smywiński-Pohl A. (2019). *Results of the PolEval 2019 Shared Task 3: Entity Linking*. In Ogrodniczuk M. and Łukasz Kobyliński (eds.), *Proceedings of the PolEval 2019 Workshop*. Institute of Computer Science, Polish Academy of Sciences.

Turing A. M. (2009). *Computing Machinery and Intelligence*. In Epstein R., Roberts G. and Beber G. (eds.), *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, pp. 23–65. Springer Netherlands, Dordrecht.

Wojczulis M. and Kłeczek D. (2021). *papuGaPT2 — Polish GPT2 Language Model*. `https://huggingface.co/flax-community/papuGaPT2`.

Wróbel K. (2017). *KRNNT: Polish Recurrent Neural Network Tagger*. In *Proceedings of the 8th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pp. 386–391. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu.

Wróblewska A. and Krasnowska-Kieraś K. (2017). *Polish Evaluation Dataset for Compositional Distributional Semantics Models*. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 784–792, Vancouver, Canada. Association for Computational Linguistics.

Xue L., Constant N., Roberts A., Kale M., Al-Rfou R., Siddhant A., Barua A. and Raffel C. (2021). *mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer*. In *Proceedings*

*of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 483–498, Online. Association for Computational Linguistics.

Yang Y., Yih W.-t. and Meek C. (2015). *WikiQA: A challenge dataset for open-domain question answering*. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.