Open-domain QA Polish model

Maciej Domaradzki, Jakub Krajewski and **Krystian Król**
University of Warsaw
m.domaradzki@student.uw.edu.pl
j.krajewski@students.mimuw.edu.pl
k.krol@students.mimuw.edu.pl

Abstract

We propose an approach to create model, which answers questions from the Polish TV quiz 'Jeden z Dziesięciu'. We describe our model, which operates directly on polish sentences and present the results. We also discuss another approach, which uses model created for English language and translates questions and answers.

1 Introduction

This project is intended to give answers to general knowledge questions asked in Polish. The main task it addresses is question answering, typical for popular TV quiz shows, such as Fifteen to One or more specifically Jeden z Dziesięciu, as the questions are asked in Polish. Another problem we tackle with our project is natural language understanding as we need to learn from data, which in our case is just the Polish articles from Wikipedia. We decided to rely solely on Wikipedia, as it is probably the largest encyclopedia available in Polish language and contains more than 1.5 million articles (3). Moreover, being open-soured and relying on contributors, Wikipedia pushes the democratization of knowledge and ideas. Nevertheless, the crowd-source nature of the data make the source human-oriented, which in turn makes the natural language understanding part particularly important, since in human language some information gets carried in implicit form. We evaluate our model on the dataset called "test-B" provided as a part of task 4 (2), namely "Question answering challenge", in the PolEval 2021 (1) competition organized by the Institute of Computer Science in the Polish Academy of Sciences. We compare our results to the benchmarks and results provided by the contest's organizer.

To achieve our goal, we propose a model composed of two modules. The first module is responsible for retrieving most relevant articles from the cached database of Wikipedia articles. Following (Semnani and Pandey, 2020), we use articles from the cached database in the second module of our model to provide context. Based on the information from within the articles and the question, the final module is responsible for providing answers.

The performance of model is measured by accuracy, which in our case means providing exactly the same number retrieved by regular expression in case of numerical question or comparison of Levenshtein distance between expected answer and given answer.

2 Related work

Answering open-domain questions is an active research topic. Typically, such tasks are divided into two parts. Firsty, a subset of relevant documents is found. In the second stage the aim of the model is to find the exact answer in the text. In the article (Chen et al., 2017) authors use a search component based on bigram hashing and TF-IDF matching. The final output is then generated by search in Wikipedia paragraphs performed by a multilayer recurrent neural network. Large improvement in solving the task was achieved by (Yang et al., 2019a) by using a BERT-based reader and Anserini Information Retrieval toolkit. This approach was further developed in (Yang et al., 2019b) by adding data augmentation. Promising results have been achieved in (Semnani and Pandey, 2020) by using a slightly different approach. The proposed pipeline consists of a traditional BM25-based information retriever, RM3-based neural relevance feedback, neural ranker, and a machine reading comprehension stage. Authors of paper (Lewis et al., 2020) show that it can be beneficial to first find k most relevant articles in the database and then use it as a context for the model. Most of the available research papers focus on the English language. For Polish, a system combining question answering and entity recognition was presented in (Przybyła, 2016).

3 Model

The model comprises of two parts - context retrieving and answering question based on context. The first part also calculates the score - how much information does the article seem to carry. This value is used to decide the best answer out of all answers for each possible context article.

The hyper-parameters describing relation between question-answering output and context score were fine-tuned according to "test A" question base provided during the competition.

3.1 Context retriever

When retrieving context, we treat questions as set of words. Not all words carry the same quality of information. For each question we are interested only in those words, which are essential for the question, hence we do keyword extraction according to the RAKE algorithm proposed in (Rose et al., 2010). The only difference is that, instead of phrases and their scores, we extract just the list of keywords.

The problem with Polish language is that it has many ways to conjugate the word. We compared two approaches: StempelStemmer (5) and removal of last character. The latter solution yield better results, as stemmer often has problem with proper nouns. Including more red-hearings is also a small price to pay for including all relevant articles, which could have been omitted due to the conjugation.

Then for each article from database we measure it's relevance to question by comparing it with the list of key words making up the question. More specifically we use pattern matching of each key word in each article and calculate the metric upon match. The metric puts higher focus on articles containing multiple key words and length of those key words, but penalizes broadly used key words - namely the more common the key word, the smaller the metric is. Top-k articles are then retrieved to serve as a context for question-answering part. The score can be described as

$$s_W(c) = \sum_{w \in W} \mathbb{1}_{w \in c} \frac{|w|}{|w \in C|},$$

where:
s_W - score function for a given question
W - set of key words
c - single article
C - set of all articles ($|w \in C|$ means number of times w appears in all articles).

3.2 Question answering

We decided to used pretrained model bert-base-multilingual-cased-finetuned-polish-squad2 (4). This is a multilingual model created by Google and finetuned on polish Q&A task. To work it needs question and context from which it generates the answer. We also decided to used tokenizer delivered with this model. We choose the best answer as below:

$$argmax_a \sum_{c \in K} s_W(c) \cdot B_{score}(a) \cdot \mathbb{1}_{a=B(c,q)}$$

where:
s_W - score function for a given question
W - set of key words
a - potential answer
K - set of top-k articles from context retriever
c - single article
q - question
B - question-answering model
$B_{score}(a)$ - evaluation of the answer by the model

4 Experimental Set-Up

As a database for our model we decided to use Wikipedia, more specifically the 2022-06-01 dump of Polish Wikipedia. We then applied (6) to retrieve plain text from xml format. Finally, from each article we extracted just the abstract, as experimentally we established that in most cases abstract carry more qualitative information, which yield better results provided as an abstract in comparison with providing model with whole article. Another advantage of using just the abstract is that we have small database, hence easier to lookup, which enabled us to decrease single question answering time from 4 minutes to 40 seconds.

5 Results and Discussion

The performance of our model in solving open-domain question answering task has been measured on a test set by accuracy. In case of numerical answers, providing exactly the same number retrieved by regular expression has been measured. For textual answers, Levenshtein distance has been used. We expect the distance to be of no more than ½ of the length of the gold standard answer. We compare our results

to the WIKI_SEARCH baseline as described in (2).

Accuracy Accuracy measured on the test set equals 34%. This result outperformes baseline (13%). However, we have not been able to reach the best results provided in (1). It is also important to note that the Top-5 score (measured by checking if any of the 5 answers ranked highest by the algorithm is correct) is significantly better, reaching 54%. We have observed several problems during evaluation of the model related to the construction of the model and testing procedure. Some of them are highlighted below.

Language structure Polish language is very syntactically rich. Each word usually occurs in multiple forms, depending on the surrounding context (e.g. declension of nouns/conjugation of verbs). This poses a challenge to any Natural Language Processing algorithm. In many cases, our model returns the proper word, but not in the correct grammatical form. In one of the questions, model returned "morskiego" while the expected answer was word "morska". The model also sometimes had problems in choosing the right part of speech, like returning "włoskiego" instead of "z Włoch".

Types of questions In some cases the expected form of the answer can is indicated by the way question is asked. In addition to open form questions, the dataset contains numerical or multiple choice (yes/no) problems. As our model was not directly trained to discriminate between types of questions, it has sometimes struggled to find the correct result. This is especially visible in yes/no questions. For example in the question "Czy Paryż leży we Francji?" we get an answer "Francja średniowieczna". There are several examples similar to that one.

Evaluation procedure In several questions, the model was able to return a reasonable answer, which could not pass because of the used metric. For example, the model returned short form "EWWiS" instead of "Europejska wspólnota węgla i stali". Also, some questions were ambiguous. The expected answer for a question regarding cabaret of artists Zenon Laskowik and Bohdan Smoleń was "Tey". Our model returned "Teyatr", which is a name of another group formed by the two.

Follow-Up Work The problems we noticed during the evaluation can provide a basis for further development of the algorithm. One of the possible directions would be to use the fact in many cases, the correct answer can be found in the Top-k predictions. Therefore a multi-stage procedure could be used. After determining k proposals, another component of the model would be responsible for choosing the most probable final answer.

Another idea would be to address the problem of different types of questions. For example, we could first classify the problems and then use different, specialized models for each category. This approach is promising, as some of the questions in the dataset are in fact multiple-choice (answer is to be chosen out of several proposed options). This type of problems is generally easier to solve and models tend to achieve better results in such setting than in general question answering (Vo et al., 2022).

To address the syntactic richness of Polish language, we could extend our model to recognize subwords and include part-of-speech tagging for the purpose of ensuring final results to be in the correct form.

Translation approach While we were working on the project, we also considered another approach, which was to translate the questions and use the full pipeline proposed in (Chen et al., 2017). This idea is potentially promising, thanks to the development of modern translators and high performance of the original DrQA model. On the other hand, this solution might pose problems, because some of the questions specifically address Polish context (e.g. cultural events, Polish sayings). Another potential problem, on the side of translating answers, may be Proper names. Some of them are different between languages (for example "Naples" is "Neapol" in Polish), which may be hard to catch for the machine translator.

Realisation of this idea is possible by using publicly available translators (7) or (8) and full implementation of the DrQA paper available at (9). Finally, we were not able to conduct experiments in that setting, due to technical problems and high memory and computation requirements (probably one of bottlenecks is the document retrieval part of the mentioned DrQA implementation).

6 Conclusions

We have created the model that outperforms the baseline and described another, translation-based approach. We also indicated the problems of our model. We have shown that one of the difficulties in generating good results comes from Polish language being syntactically rich. We also indicated problems with evaluation procedure.

References

1. Poleval 2021.

2. Poleval 2021 :: Task 4.

3. Wikipedia polskojęzyczna.

4. bert-base-multilingual-cased-finetuned-polish-squad2.

5. Stempel stemmer. https://github.com/dzieciou/pystempel.

6. Wikiextractor. https://github.com/attardi/wikiextractor.

7. Argos translate. https://github.com/argosopentech/argos-translate.

8. Marian translate. https://github.com/marian-nmt/marian.

9. Drqa implementation. https://github.com/facebookresearch/DrQA.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Piotr Przybyła. 2016. Boosting question answering by deep entity recognition.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. *Automatic Keyword Extraction from Individual Documents*, chapter 1. John Wiley & Sons, Ltd.

Sina J. Semnani and Manish Pandey. 2020. Revisiting the open-domain question answering pipeline.

Dinh-Huy Vo, Anh-Khoa Do-Vo, Tram-Anh Nguyen-Thi, and Huu-Thanh Duong. 2022. An approach for multiple choice question answering system.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-end open-domain question answering with. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019b. Data augmentation for bert fine-tuning in open-domain question answering.