A Question Answering Benchmark Database for Hungarian

Attila Novák and **Borbála Novák**
Pázmány Péter Catholic University, Faculty of Information Technology and Bionics
Práter u. 50/a, 1083 Budapest, Hungary
{surname.firstname}@itk.ppke.hu

Tamás Zombori and **Gergő Szabó** and **Zsolt Szántó** and **Richárd Farkas**
University of Szeged, Institute of Informatics
Árpád tér 2, 6720 Szeged, Hungary
ztamas2000@gmail.com {gszabo,szantozs,rfarkas}@inf.u-szeged.hu

Abstract

Within the research presented in this article, we created a new question answering benchmark database for Hungarian called MILQA. When creating the dataset, we basically followed the principles of the English SQuAD 2.0, however, like in some more recent English question answering datasets, we introduced a number of innovations beyond SQuAD: e.g., yes/no-questions, list-like answers consisting of several text spans, long answers, questions requiring calculation and other question types where you cannot simply copy the answer from the text. For all these non-extractive question types, the pragmatically adequate form of the answer was also added to make the training of generative models possible.

We implemented and evaluated a set of baseline retrieval and answer span extraction models on the dataset. BM25 performed better than any vector-based solution for retrieval. Cross-lingual transfer from English significantly improved span extraction models. [1]

1 Introduction

In this research, our goal was to create a Hungarian question answering dataset that enables the training of Hungarian question answering systems and the automatic evaluation of their performance. In the paper we first review existing systems and resources, then describe the annotation procedure we followed and features of the dataset, closed by the presentation and evaluation of baseline retrieval and extractive answer span extraction models trained and tested on the dataset.

[1]The dataset and trained models can be found on GitHub and the Hugging Face Model Hub searching for the term MILQA.

2 Background

Early question answering databases were either very small in size or did not contain questions in the form of grammatical interrogative sentences, but they consisted of so-called cloze-type "questions": these are declarative sentences, a part of which is masked and this part must be filled in based on the text. The latter resources were machine-generated, so they were easy to create, but the sentences containing the masked part do not resemble real questions at all.

One of the most important milestones in the series of databases used for training question answering systems was the English SQuAD database (Stanford Question Answering Dataset) (Rajpurkar et al., 2016) created at Stanford University. This is a much larger database than the previous ones, containing more than 108,000 question-answer pairs in its first version, which was later further supplemented with questions that could not be answered based on the given text passage (151,000 questions, (Rajpurkar et al., 2018)) in the second version. The publicly available training and tuning set contains 143,000 (93,000 answerable and 50,000 unanswerable) questions. In addition to its size, this resource can be considered a breakthrough because, on the one hand, unlike previous resources containing cloze-type questions (e.g. CNN/Daily Mail (Hermann et al., 2015)), it actually contained well-formed questions and on the other hand, it was not built of multiple-choice questions (e.g. MCTest (Richardson et al., 2013) or WikiQA (Yang et al., 2015)). Furthermore, it gave a huge boost to the development of question answering systems.

Among question answering datasets and systems, we can distinguish extractive and generative approaches. In the case of the former, the answer is simply a highlighted part of the text (as if we

Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII), pages 188–198

were working with a text highlighter, this is what SQuAD is like), and in the case of the latter, the answer is actually formulated in well-formed human language (e.g. MS MARCO (Nguyen et al., 2016), NarrativeQA (Kočiský et al., 2018)). In addition, some of the QA databases contain questions that require the execution of multi-step inference chains to arrive at an answer (multi-hop/multi-step QA tasks). This not only means a greater complexity of the underlying logical derivation, but this type of task can also go beyond the level of individual documents or text fragments, if the given question can only be answered by combining the information contained in several different documents or text fragments (e.g. HotpotQA (Yang et al., 2018), NarrativeQA (Kočiský et al., 2018)).

In the case of the multi-step question answering tasks and SQuAD, it was the task of the annotators to formulate questions based on given texts. Companies operating large search engines, however, created resources in which relevant documents were collected based on frequent questions entered into the search engine, and the annotators selected or formulated the answers using these results. Natural Questions (NQ, Kwiatkowski et al. (2019)) based on questions entered into the Google search engine belongs to the former extractive type. In NQ, the documents used as context were Wikipedia articles, similar to SQuAD. The MS MARCO QnA dataset based on Microsoft Bing queries belongs to the latter abstractive/generative type (Nguyen et al., 2016). Resources based on existing quiz and literacy question sets were also created using similar web query techniques (e.g. TriviaQA (Joshi et al., 2017)).

Perhaps one of the sources of SQuAD's popularity was that it assumes a relatively simplistic model, according to which a single coherent span of text can be selected as an answer for each answerable question, which greatly simplifies the implementation of SQuAD-based systems. This restriction can be implemented well if the annotators are instructed to ask only questions that can be answered in this manner. However, in the case of a non-negligible part of real-life questions, the answer is some kind of list, the elements of which do not necessarily occupy a single contiguous span of the text. In such cases, a single span including all relevant answers may contain a significant amount of text that is irrelevant to the answer. For example, in the Natural Questions dataset based on real questions, the answer is not a single span for 6.9% of the questions. In the case of SQuAD, the context of the questions (the part of the text in which the answer to the question must be found) has a relatively limited length: between 150 and 4000 characters, with an average of 740 characters, which also limits the complexity of the task.

Yes-no questions naturally occur in datasets similar to NQ (Natural Questions: 2.5%) that originate from actually asked questions. Typically, the answer to these questions is not a selected part of the text, but a (usually probable, not clear) yes/no answer follows from a relevant part of the text. There are also datasets specifically containing only yes-no questions (e.g. BoolQ (Clark et al., 2019), also based on Wikipedia, AmazonYesNo (Dzendzik et al., 2019), based on texts related to Amazon product reviews, or the biomedical PubMedQA based on article abstracts (Jin et al., 2019)). At the same time, BoolQ and AmazonYesNo show significant overlap with the yes-no questions in NaturalQuestions and AmazonQA (Gupta et al., 2019) databases (in the case of Amazon resources, there is essentially a subset relationship).

In biomedical question sets of "natural origin", similarly to NQ, the proportion of "non-SQuAD-compatible" questions is often much higher than previously mentioned in relation to the NQ database. For example, in the case of the Clinical Questions Collection (CQC) data set (D'Alessandro et al., 2004; Ely et al., 1997, 1999) containing questions formulated by actual practicing doctors during their daily professional activities and the PubMed Query Log Dataset (Herskovic et al., 2007) composed of questions formulated by PubMed users in a single day, the proportion of yes-no questions is 28.1%, and that of list-type answers is 21.9% (Yoon et al., 2022).

In addition to the lack of list-type answers and the scarcity of yes-no questions, another problem with extractive datasets arises from the fact that questions about a given text often do not use the same words that appeared in the original context. During the compilation of SQuAD, annotators were encouraged to paraphrase the part of the question anchored to the context when formulating the questions, and not simply copy it. This in itself is not necessarily a serious problem for neural models based on current pre-trained language models, since these usually have sufficiently abstract internal semantic representations to often avoid that

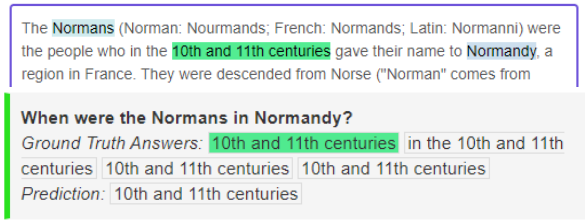

Figure 1: In SQuAD, relevant prepositions are usually not included in the answer

paraphrases confuse them. However, if we use a verb in the question different from the one in the original context, then this often involves a different argument frame, which means that the given expression should often appear in the answer in a form different from that in the original text. In the case of SQuAD, the solution to this problem was that prepositions were not made part of the answer, but only the minimal lexical content (annotators were instructed to do so, see Fig. 1.).

In the case of English, an essentially isolating language, this solves the above problem in most cases, but at the price that the answer of the system is often not formulated in a pragmatically appropriate form (the latter would include the preposition). In the case of languages, where case is marked morphologically, this solution obviously does not work. In such cases an extractive QA system will definitely give an inadequate answer, because it returns the answer with the original case appearing in the text. At the same time, this does not represent a real problem if the answer is presented as highlighted text in context, since in this case the user does not feel that the machine "answered in a strange manner", but rather that it "highlighted the answer correctly in the text". If, however, the answer is presented as an answer, then it is definitely necessary to move on and use a generative model.

We illustrate the problem with an example in Hungarian. In the context of *Péternek az idegeire ment a zaj.* 'The noise got on Peter's nerves.' (here *Péter* 'Peter' is in the dative case), the adequate short answer to the question *Kit idegesített a zaj?* 'Who was annoyed by the noise?' would be *Pétert* (in accusative), but this cannot be extracted in this form from the original context. Here, the complete sentence would be an adequate (but not minimal) answer to the question. However, this is often not the case, especially when the original context contains the answer in a derived form. In the context *A Duna Európa második leghosszabb folyama az oroszországi Volga után.* 'The Danube is the second-longest river in Europe, after the Volga in Russia.', the adequate answer to the question *Melyik országban található Európa leghosszabb folyama?* 'In which country is the longest river in Europe?' would be *Oroszországban* 'In Russia' (inessive of *Oroszország*). The word form *oroszországi* in the original context is an adjective derived from the name of the country (and as such, it is decapitalized). Here, the original sentence would not be an adequate answer, either.

There are some additional question types: question-answer pairs that require counting, the execution of some arithmetic operations, or comparison *(how many, how much, which is the most... etc.),* which are not a problem even for people with minimal education, but the models must be specially prepared to perform such tasks in order to prevent the machine from failing miserably. The DROP (Dua et al., 2019) question-answer database primarily focuses on such questions.

Some resources approach the problem of answering questions in the context of a dialogue. The questions are often ambiguous or incomplete, and additional information is needed to answer them. Data sets such as ShARC (Saeidi et al., 2018) aim at modeling such situations. Training the groundbreaking ChatGPT model of OpenAI required extensive dialog modeling resources as well as further human-in-the-loop annotation for reinforcement learning.

2.1 Non English resources

All the previously mentioned question answering databases (and countless others) are in English. At the same time, the presented methods have been adapted to many other languages, and multilingual question answering datasets have also been created.

Relatively many and large datasets in Chinese have been created. The best known is DuReader (He et al., 2018) based on Baidu searches and Baidu Zhidao, a Chinese question-and-answer platform.

Based on the SQuAD approach, French (FQuAD 2.0, Heinrich et al. 2022, almost 80000 questions), Korean (KorQuAD 2.0, Youngmin Kim 2020, 100000 questions), Russian (SberQuAD, Efimov et al. 2020) and German (GermanQuAD, Möller et al. 2021, approx. 14000 questions) resources have also been created. XQuAD (Artetxe et al., 2019) contains translations of 1190 question-answer pairs related to 240 paragraphs from the

SQuAD 1.1 tuning set (dev. set) by professional translators in 10 languages.

The MLQA benchmark database covering six other languages in addition to English (Lewis et al. (2020); about 12,000 question-answer pairs for English and 5-6 thousand question-answer pairs for the other languages), is built around quasi-equivalent Wikipedia sentences to which the questions were translated from English by translators. SQuAD has been machine-translated into several languages (e.g., Korean, Hindi, Japanese, Spanish, Czech, French, and the languages included in the MLQA dataset).

11 typologically diverse languages are covered by the TyDi QA dataset (Clark et al. (2020); a total of 200,000 question-answer pairs), which is also based on Wikipedia. The questions were formulated based on the introductory section of the articles only, but you could ask anything related to the topic. Thus, most of the questions formulated in TyDi QA do not have an answer, but where there is, the method guarantees that the question is formulated differently than the answer.

3 A new Hungarian question answering benchmark dataset

Within the research presented in this paper, we created the first publicly available extractive question answering benchmark dataset in Hungarian. When creating the database, we largely followed the principles of SQuAD 2.0, however, similar to some of the more recent English Q&A databases (Natural Questions, MS MARCO, DROP) mentioned in section 2, we introduced a number of new question-answer types, which contain more difficult but more realistic tasks.

Similarly to SQuAD 2.0, the corpus is characterized by the following: **a)** high-quality Wikipedia articles serve as context for the questions, **b)** factual (not opinion-type) questions are included, **c)** also contains questions that are not answered in the given text, **d)** in the original text, we marked the shortest possible answer to the given question (if any), **e)** when formulating the questions, we paraphrased the original text, so in most cases the answer cannot be found using a lexical search, **f)** the questions can be interpreted not only in the context of the given text, but also as independent questions (e.g. they do not contain unanchored pronouns).

Compared to SQuAD, we introduced the following innovations (special question types are explicitly marked in the database): **a)** There may be more than one short answer to the given question in the given text (list type answer, approx. 8.5% of the answered questions). **b)** In addition to the short answer, we also gave a long answer, which includes all the relevant information necessary to answer the question (min. 1 clause, often several sentences). **c)** It contains yes-no questions (about 9%). Here, in addition to the long answer containing the essential circumstances, an explicit yes/no answer is also specified (or the lack of a clear binary answer is indicated). **d)** The unanswerable questions (about 28.3% of the questions) are relevant questions related to the given topic, not questions generated by substitution from questions having an answer. **e)** There are also questions that can only be answered after performing counting or arithmetic operations (similarly to the DROP database). Calculations involve counting of listed elements, calculation of dates, durations and other quantities with simple arithmetic operations. **f)** Some of the unanswerable questions are tricky questions, where people would easily infer an answer from the text based on wrong default assumptions. These cases were marked separately, and the assumed answer was also indicated. **g)** If the expression in the text does not correspond to the form in which the given question should be answered (e.g. the original case ending is not appropriate), the annotators have provided the form of the answer appropriate in the context of the question.

3.1 Creation of the corpus

In order to create the data set that forms the basis of the database, we selected articles from the Hungarian Wikipedia marked as featured or high quality articles, and sorted them based on their page visit counts between the beginning of 2016 and the end of 2021. From this list, the annotators selected the articles to be annotated based on their personal interests in order to avoid that the annotation task become unpleasant or boring to them. They were also encouraged to abandon and report articles they found low quality or uninteresting and to move on to a new task. We used the first section of each article and, in addition, a maximum of 10 randomly selected sections of at least 500 characters. Similarly to SQuAD, the units were paragraphs, but paragraphs shorter than 500 characters were combined, and we omitted those longer than approximately

1200 characters (text sections of this size could be clearly displayed on the annotation interface).

The annotation interface was created by customizing version 1.4 of the Label Studio open source web annotation platform. It was a relatively complex task to make the interface suitable for asking questions, marking the corresponding answers, and marking special question and answer types in a intuitive manner, but we managed to create a relatively easy-to-use user interface and workflow for the annotators. (Figure 2.).

Questions were added as text markup by the annotators. Answerable questions were numbered. We used the span annotation feature of Label Studio, usually used to do named entity annotation, to mark the long/short answers. Questions and answers were matched on the basis of the question number. List answers were marked as a set of spans referring to the same question number. As overlapping spans marking answers to different questions could easily clutter the annotation interface, shortcuts could be used to make answer spans belonging to other questions invisible. The answers could be marked as yes/no/arithmetic/non-extractive/wrong (for tricky unanswerable questions), and an explicit non-extractive answer was entered for arithmetic and non-extractive questions.

The annotation system provided the annotators with continuous statistics on the progress, and they could also invoke the display of all questions and extracted short and long answers belonging to the given context to check that the answers were marked as intended. The annotation was made by five annotators. Apart from the more problematic cases that were later re-edited, the time required for the work can be estimated well based on the editing time stored by Label Studio: it took roughly 85 seconds per question to formulate the questions and mark the long and short answer spans and the eventual reformulation of the answer if necessary.

A part of the corpus containing 2391 questions (including 1751 answerable questions) consisting of 36 articles (roughly 10% of the corpus) was separated for a test/tuning set, and two independent annotations were made for this part. The annotation work, which did not require writing questions, progressed faster: it took an average of 46 seconds to mark the long and short answers.

Type	number	ratio
There is an answer	**16992**	71,67%
. Yes-no	1621	9,20%
. . Yes	859	52,99%
. . No	638	39,36%
. . Uncertain	124	7,65%
. Not an extractive answer	4452	26,20%
. Arithmetics	427	2,51%
. List	1455	8,56%
. Not SQuAD-compatible	5203	30,62%
No answer	**6716**	28,33%
. Tricky no answer	629	9,37%
Sum	**23708**	100

Table 1: Distribution of question and answer types in the dataset. For subtypes, the ratio column indicates the ratio within the given main type.

3.2 Features of the corpus

The database contains a total of 23,700 (17,000 answerable and 6,700 unanswerable) questions. Questions were created for 142 Wikipedia articles. In Table 1, we have summarized the occurrence of special question and answer types in the corpus.

9.20% of the questions are yes-no questions. In the case of 7.65% of these, there is no clear yes/no answer, but the text reflects that the opinions on the given question are diverse, the results are mixed, or there is uncertainty. At the same time, this is not the same as the case of unanswerable questions, where the text does not answer the question at all: here the text explicitly reveals that the world is not black and white from the point of view of the given question. In the case of yes-no questions, the span annotation is relevant in the sense that the answer follows from the marked spans. The yes-no question type is not SQuAD-incompatible in itself: the original SQuAD also contains yes-no questions, which were all formulated in a way that a nice extractive answer could be given to them. What is new here is that the annotation includes an explicit marking for this type of questions and whether the answer is yes or no. About 9% of unanswerable questions are yes-no questions.

The annotation environment and specification allowed annotators to work free from the usual restrictions in SQuAD (i.e. that the answer should be exactly a single span in the text). This resulted in more than 30% of the questions that have an answer in the text being not SQuAD compatible. 26.2% of the (answered) questions are not extrac-

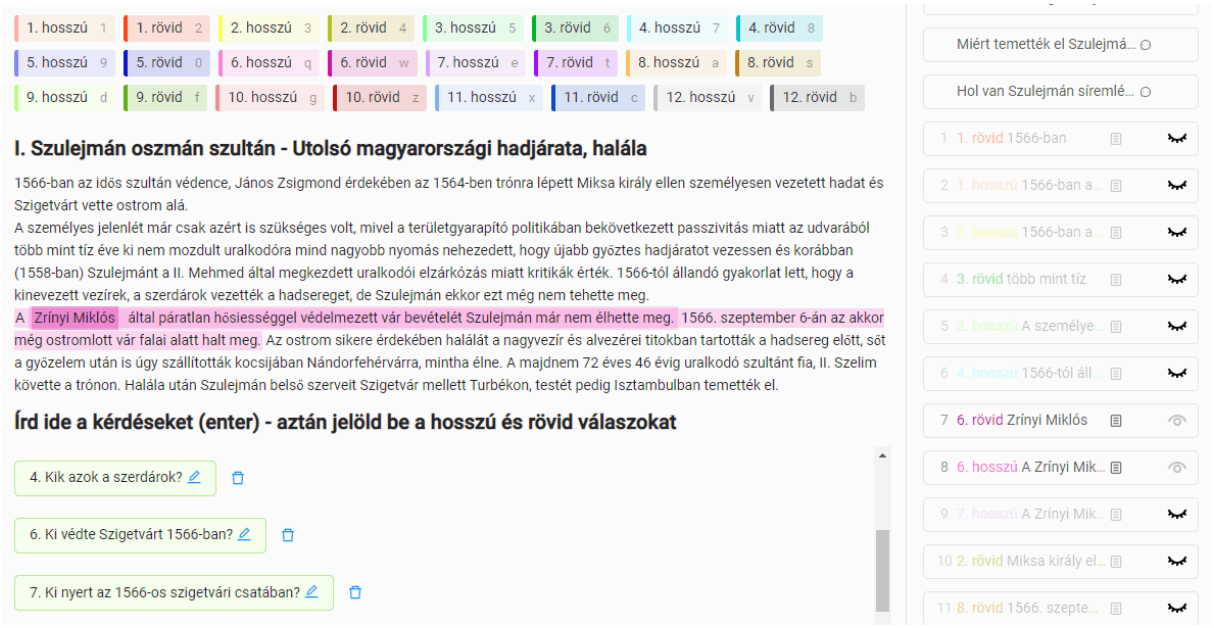

Figure 2: The annotation interface for the corpus is based on Label Studio

tive: the natural form of the answer to the given question would be different from what is in the text (e.g. the given expression would need to have a different case ending to be an adequate answer to the question). To answer 2.51% of the questions, some calculations need to be performed (similarly to those in the DROP database; the answer cannot be copied from the text for these either, so they are included in the former 26.2%). And for 8.56% of the questions, SQuAD's "single contiguous answer span" assumption is not fulfilled (this set also partially overlaps with cases where the form needs to be modified to be adequate).

9.37% of unanswerable questions are tricky. For these, one tends to derive an answer based on some rule-of-thumb assumptions (even by doing calculations), the result of which could easily prove to be wrong. For example, in a particular paragraph of the Normandy landings article, from the fact that the fleet units participating in the landings had three commanders, one might infer that there were three fleet units; in fact, there were only two, and there was a commander-in-chief.

As for question words, the most common questions ask about the subject (>17%), dates/times (>10%), reasons (>8%), quantities (>7%) and places ($\sim$ 7%).

4 Models and performance

We created and evaluated a number of document retrieval and reader (answer span extraction) models using the dataset. For document retrieval, we evaluated both traditional lexical and various vector-based retrieval models. For span extraction, we finetuned both a monolingual Hungarian model and multilingual models. We also tested to what extent cross-lingual transfer from English can be applied to this specific task.

4.1 Document retrieval models

The first model we applied for document retrieval was a BM25-based solution (Robertson and Zaragoza, 2009) using Elasticsearch. BM25 (Best Matching 25) is a simple and effective ranking function widely used in information retrieval systems. It takes into account term frequency, document length and inverse document frequency to calculate the score representing the relevance of a document to a query. Our first experiment concerned the question to what extent traditional preprocessing steps like lemmatization or part-of-speech-based term filtering can improve retrieval performance. We expected some improvement, because Hungarian is a morphologically rich language. We performed preprocessing using components of the HuSpaCy library (Orosz et al., 2022; Szabó et al., 2023). In this experiment, we tested the accuracy of selecting the exact paragraph corresponding to answerable questions from all paragraphs in the dataset. The results are shown in Table 2. We have found that applying lemmatization and a simple PoS-based filter to eliminate wh-words improves retrieval per-

Preprocessing	R@1	R@3	R@4	R@5	R@10	R@300	MRR@300	@300-w-time
Base	0.438	0.595	0.627	0.655	0.729	0.896	0.538	466.17 s
PoS	0.448	0.603	0.636	0.665	0.741	0.878	0.547	262.25 s
Lemma	0.647	0.807	0.835	0.858	0.908	0.984	0.740	505.53 s
PoSLemma	0.656	0.814	0.844	0.866	0.916	0.984	0.748	385.31 s

Table 2: Evaluation of the effect of preprocessing on BM25 retrieval performance. Evaluated on all answerable questions and the corresponding paragraph from a pool of all paragraphs. R@1..300: Recall/match with a cutoff at position 1 ... 300. MRR@300: Mean Reciprocal Rank (with retrieval cutoff at 300 documents). Lemma: applying lemmatization. PoS: applying a simple PoS-based filter to eliminate wh-words from the query.

formance significantly.

In the follow-up retrieval experiments, all query results in which the gold answer was present *exactly* in the form given in the dataset, was accepted as a valid hit. First, we tested how performance (recall/MRR) of the retrieval model depends on the document entity type stored in the database. The results are shown in Table 3. Results in the upper half of the table are for configurations where only articles covered in the dataset were added to the document pool. In the configurations shown in bottom half of the table, we increased the size of the document pool 30 fold by adding further 4927 randomly selected Wikipedia articles.

We also evaluated sentence-transformer-based embedding and dense passage retrieval (DPR) models for context retrieval (on the base in-dataset-passages-only pool). There is no such model specifically trained for Hungarian, so we tested an English model trained specifically on QA datasets (multi-qa-mpnet-base-dot-v1) and multilingual models (which were trained on semantic similarity/paraphrase rather than QA tasks). We also tested a multilingual DPR model (it is a pair of encoders; one for the question and another for the context: dpr-(question/ctx)_encoder-bert-base-multilingual). We used the retrieval engines implemented in Haystack (Deepset GmbH, 2022). We compared the results with Haystack's BM25 implementation, which differs from our own in that it does not involve lemmatization. The results are shown in Table 4.

All embedding-based models performed significantly worse than the simple and fast BM25 model. Of the vector-based models, multilingual models covering Hungarian finetuned on paraphrase databases performed best. The DPR model had the weakest performance in spite of being both multilingual and specifically trained for QA passage retrieval. The English-only QA-trained *mpnet* model performed significantly better than the multilingual paraphrase-based *distiluse-bmc-v1* model (USE: Universal Sentence Encoder), which does not cover Hungarian, either.

4.2 Reader models

In our experiments concerning reader models, we finetuned baseline answer span extraction models. Here we used only the unproblematic SQuAD-compatible questions in the dataset (i.e. where the extracted answers need not be reformulated to be adequate and arithmetic reasoning is not needed.) There was one exception to this: we created two versions of each model variant that differed in how multispan answers were handled. In one version, individual spans were handled in the training and test set as if they were independent question answer pairs. In another version, questions with multispan answers were omitted from both the training and the test set. The *with multispan* and *no multispan* columns of Table. 5 on model evaluation correspond to these model versions. The models do not currently properly handle multispan answers, because they consider the most likely span only. As an orthogonal dimension, we created and evaluated models on short and long answers. The long answers task is easier: only the clauses relevant to the question need to be identified rather without focusing on the actual answer.

We finetuned models from scratch from the Hungarian BERT base model huBERT (Nemeskey, 2021) on the short and long answers in the dataset (hubert-base-T in Table. 5). The model turned out to be undertrained for the short answer task. So we experimented with knowledge transfer from SQuAD 2.0. We tested one model finetuned from huBERT on a machine translated version of SQuAD 2.0 (huBert-squadv2[2]), and two XLM-RoBERTa-based models finetuned by Deepset di-

[2] https://huggingface.co/mcsabai/huBert-fine-tuned-hungarian-squadv2

	R@1	R@3	R@4	R@5	R@10	R@300	MRR@300	@300-w-time
In-dataset articles only								
Base	0.662	0.816	0.846	0.868	0.919	0.984	0.753	453.48 s
Paragraphs	0.475	0.621	0.651	0.675	0.736	0.872	0.567	502.12 s
Sections	0.577	0.741	0.772	0.791	0.837	0.896	0.671	839.22 s
Articles	0.824	0.879	0.885	0.888	0.896	-	-	-
In-dataset + 4927 random articles								
Paragraphs	0.412	0.562	0.593	0.618	0.682	0.860	0.506	486.17 s
Sections	0.485	0.664	0.704	0.729	0.792	0.891	0.593	708.12 s
Articles	0.617	0.733	0.754	0.768	0.804	0.904	0.686	20188.95 s

Table 3: Retrieval performance wrt. document entity types in the document pool. Evaluated on all answerable questions. The rows represent the configuration of document entities in the database. Base: In-dataset paragraphs only. Paragraphs: all paragraphs of all Wiki articles in the pool. Sections: all sections of articles. Articles: all full articles. R@1..300: Recall/match with a cutoff at position 1 ... 300. MRR@300: Mean Reciprocal Rank (with retrieval cutoff at 300 documents).

Model	Lang/training	R@10	MRR@10
haystack BM25		**0.817**	**0.626**
multi-qa-mpnet-base-dot-v1	English only QA	0.483	0.285
paraphrase-multilingual-MiniLM-L12-v2	multiling. paraphrase	0.566	0.315
distiluse-base-multilingual-cased-v1	15 lang USE	0.299	0.150
distiluse-base-multilingual-cased-v2	50+ lang USE	**0.589**	**0.326**
dpr-encoder-bert-base-multilingual	m-BERT-based DPR	0.281	0.123

Table 4: Evaluation of vector-based retrieval models on the base in-dataset-passages-only pool. BM25 far outperformed all of them. The best model performance is in bold.

rectly on SQuAD 2.0 (xlmr-(base/large)-squad2[3]). Zero-shot performance of these models is shown in the zero-shot section of Table 5. As these models were not trained to identify long answers, they unsurprisingly perform poorly on that task (with the exception of question types where short answers tend to be full clauses, like *why* questions). Also xlmR-base-squad2 performed worse than huBert-squadv2 across the board in spite of the fact that xlmR-base is more resource-hungry (in part due to its extensive multilingual token dictionary and the corresponding embeddings), so we did not include xlmR-base-squad2 in the further finetuning experiments. On the other hand, all these models performed better on the short answer task than hubert-base-T finetuned from scratch.

In the next round, we finetuned huBert-squadv2 and xlmR-large-squad2 on our train data . The models perform much better than huBert-base-T. One surprising result, however, is that while F_1 scores consistently improved, exact match scores worsened compared to the short answer span zero-shot models. We need to investigate why this happened. xlmR-large-squad2-T performs best in this group. On the other hand, this model is much more resource hungry than the monolingual BERT-base-sized models.

Finally, we turned to the Retro-Reader model type, which involves a cascade of sketchy and intensive reader models (Zhang et al., 2021). The training and evaluation of these models is in progress, but preliminary results presented in Table 5 show that they outperform all other models on the short answer task. On the other hand, training these models requires about twice as much computation as the vanilla single transformer models as they are combination of two models. Inference also requires twice as much computation and memory.

5 Conclusions

We presented a new QA benchmark database in Hungarian, that in several aspects, goes beyond SQuAD-type datasets: it is not limited to single contiguous short extractive answer spans, contains yes/no questions, non-contiguous multispan short answers, long answers, questions requiring arith-

[3] https://huggingface.co/deepset/xlm-roberta-large-squad2

model	short answers				long answers			
	with multispan		no multispan		with multispan		no multispan	
	F_1	EM	F_1	EM	F_1	EM	F_1	EM
Zero-shot models								
huBert-squadv2	0.595	0.473	0.653	0.538	*0.331*	*0.170*	*0.332*	*0.171*
xlmR-base-squad2	0.553	0.442	0.612	0.507	0.323	0.182	0.325	0.183
xlmR-large-squad2	0.646	0.516	0.712	0.591	0.372	0.204	0.373	0.205
Transformers QA models finetuned on the train set								
huBert-base-T	0.439	0.258	0.486	0.304	0.701	0.383	0.706	0.388
huBert-squadv2-T	0.659	0.404	0.737	0.469	0.742	0.423	0.747	0.429
xlmR-large-squad2-T	0.686	0.439	0.768	0.512	0.766	0.436	0.772	0.441
Retro-Reader QA models finetuned on the train set								
hubert-base-RR	0.675	0.555						
huBert-squadv2-RR	0.702	0.572						
xlmR-large-squad2-RR	**0.724**	**0.623**						

Table 5: Performance of extractive reader models on short and long answer spans with and without multispan answers.

metic reasoning, and other questions where the answer cannot be simply copied from the text. The annotation was created using a customized Label-Studio-based annotation platform. The annotators were encouraged to get actively involved in selecting the texts to be annotated and to abandon annotation of uninteresting or low quality texts in order to make the annotation task less boring and demotivating. We also trained and evaluated baseline models for document retrieval and reader models for answer span extraction. Cross-lingual knowledge transfer naturally facilitated by multilingual transformer models was found to be beneficial for the quality of the trained models.

Limitations

In light of the near human-like lingustic performance of the groundbreaking ChatGPT model that has attracted unprecedented public attention, one can't help feeling extremely humble about the importance of the work presented in this paper on a basically extractive QA dataset in a niche agglutinating language (even if it contains annotation that can be used for training generative models capable of handling questions that cannot be answered adequately in an extractive manner). On the other hand, while we obviously do not have the resources needed to train, finetune or even run the sort of large language models that have the chance of replicating ChatGPT's behavior, models that can more-or-less decently handle the much less resource-intensive task of extracting and displaying relevant answers from stored documents in a language not too much interesting for big tech companies can be trained and run even on hardware available in our modestly equipped academic environment. Not to mention that this approach also inherently avoids the most imminent and difficult-to-handle problem of large generative models that they tend to hallucinate seemingly very convincing non-facts and to generate toxic content.

The resource is also very limited in extent compared to similar English resources both concerning size and the number of parallel annotations. In our baseline model training experiments, we have not tackled the problem of multispan answers, questions requiring counting or arithmetic reasoning, and we have not trained generative models to handle questions that cannot be answered adequately in an extractive manner.

Acknowledgements

This research was implemented with support provided by grant FK 125217 of the National Research, Development and Innovation Office of Hungary financed under the FK 17 funding scheme, and by the Ministry of Innovation and Technology NRDI Office and subsequently the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory Program.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Deepset GmbH. 2022. Haystack. Computer software.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2019. Is it dish washer safe? Automatically answering "yes/no" questions using customer reviews. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 1–6, Minneapolis, Minnesota. Association for Computational Linguistics.

Donna M D'Alessandro, Clarence D Kreiter, and Michael W Peterson. 2004. An evaluation of information-seeking behaviors of general pediatricians. *Pediatrics*, 113(1):64–69.

Pavel Efimov, Andrey Chertok, Leonid Boytsov, and Pavel Braslavski. 2020. SberQuAD – Russian reading comprehension dataset: Description and analysis. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 3–15, Cham. Springer International Publishing.

John W Ely, Jerome A Osheroff, Mark H Ebell, George R Bergus, Barcey T Levy, M Lee Chambliss, and Eric R Evans. 1999. Analysis of questions asked by family doctors regarding patient care. *Bmj*, 319(7206):358–361.

John W Ely, Jerome A Osheroff, Kristi J Ferguson, M Lee Chambliss, Daniel C Vinson, and Joyce L Moore. 1997. Lifelong self-directed learning using a computer database of clinical questions. *Journal of family practice*, 45(5):382–390.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C. Lipton. 2019. AmazonQA: A review-based question answering task. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4996–5002. International Joint Conferences on Artificial Intelligence Organization.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

Quentin Heinrich, Gautier Viaud, and Wacim Belblidia. 2022. FQuAD2.0: French question answering and learning when you don't know. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2205–2214, Marseille, France. European Language Resources Association.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Jorge R Herskovic, Len Y Tanaka, William Hersh, and Elmer V Bernstam. 2007. A day in the life of PubMed: analysis of a typical day's query log. *Journal of the American Medical Informatics Association*, 14(2):212–220.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. 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 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 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)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. GermanQuAD and GermanDPR: Improving non-English question answering and passage retrieval. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 42–50, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dávid Márk Nemeskey. 2021. Introducing huBERT. In *XVII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 3–14, Szeged. Szegedi Tudományegyetem, Informatikai Intézet.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

György Orosz, Zsolt Szántó, Péter Berkecz, Gergő Szabó, and Richárd Farkas. 2022. HuSpaCy: An Industrial-strength Hungarian Natural Language Processing Toolkit. In *XVIII. Magyar Számítógépes Nyelvészeti Konferencia*, pages 59–73, Szeged. Szegedi Tudományegyetem, Informatikai Intézet.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. 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.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.

Gergő Szabó, György Orosz, Zsolt Szántó, Péter Berkecz, and Richárd Farkas. 2023. Transformer-alapú HuSpaCy előelemző láncok [Transformer-based HuSpaCy pipelines]. In *XIX. Magyar Számítógépes Nyelvészeti Konferencia*, pages 305–317, Szeged. Szegedi Tudományegyetem, Informatikai Intézet.

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.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Wonjin Yoon, Richard Jackson, Aron Lagerberg, and Jaewoo Kang. 2022. Sequence tagging for biomedical extractive question answering. *Bioinformatics*, 38(15):3794–3801.

Seungyoung Lim;Hyunjeong Lee;Soyoon Park;Myungji Kim Youngmin Kim. 2020. KorQuAD 2.0: Korean QA Dataset for Web Document Machine Comprehension. *Journal of KIISE*, 47:577–586.

Zhuosheng Zhang, Junjie Yang, and Hai Zhao. 2021. Retrospective reader for machine reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14506–14514.