KolZchutQA - closed book Hebrew question answering dataset for the Israeli rights domain

T.Ladijinsky *

June 2023

Abstract

Current closed-book question answering NLP models in the hebrew language have made significant strides in recent years, but their ethical implications and limitations in specific domains remain understudied. To address this gap, I introduce kolZchutQA - the Israeli Rights Domain Closed-Book Hebrew Question-Answering Dataset. The dataset comprises a diverse range of questions related to various legal rights and scenarios, sourced from the Israeli website כל-זכות. I collected paragraphs from the website pages and generated 450 questions along with their corresponding answers from the contextual information. I initially evaluated the results using the finetuned HeBERT model, which was previously trained on an automatic translation of the English SQuAD dataset.
The implementation of the project can be found here: KolZchutQA GitHub repository.

1 Introduction

Closed Book Question Answering is an rapidly developing area of research in Natural Language Processing that aims to develop algorithms and models that can answer questions without any external knowledge. The goal of Closed Book Q&A is to simulate human reasoning by relying only on previously known data. The application potential of this area of research is vast, with the possibility of automating the response to frequently asked questions through the use of a knowledge base as context. This approach enables the extraction of relevant answers to customer inquiries from a set of documents, thereby streamlining customer service processes and improving overall efficiency as described in Fig.1 . One of the primary challenges in Closed Book Q&A is the sub-problem of extracting an answer from a context text. This task involves identifying the optimal answer from a given context that best matches the posed question. The complexity of natural language and the potential variability of extracted answers make this sub-problem particularly challenging. Addressing these difficulties requires effectively coping with language ambiguity, consolidating information from multiple sources and leverage content during the training process [1]. To address these challenges, researchers have explored various approaches to Closed Book Q&A. One promising approach involves leveraging pre-trained language models, such as BERT and GPT [2]. Evaluation metrics for Closed Book Q&A systems include metrics such as EM and F1-score.

2 Related Work

Creating datasets in this domain may be challenging due to several reasons. Firstly, the answer space for questions is often limited, leading to a lack of variability in the generated data. Additionally, the generation of answers often requires domain-specific knowledge, making it difficult to obtain annotations from non-experts. Secondly, it is difficult to evaluate closed book Q&A systems due to the lack of ground truth answers. In many cases, there is no single correct answer to these types of questions, leading to discrepancies in the generated data. Furthermore, the evaluation of such systems often requires human judgment, making it time-consuming and resource-intensive.

*Dept. of Computer Science, Holon Institute Of Technology, Holon, Israel

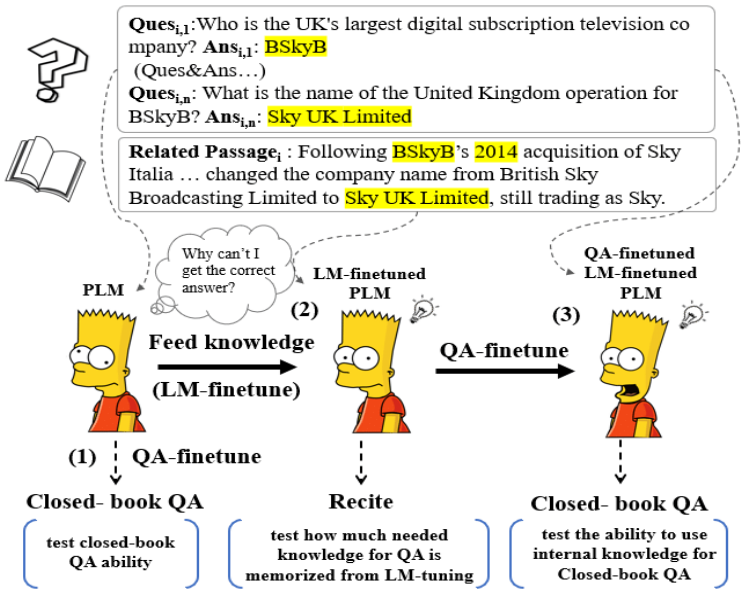

Figure 1: Process of LLMs for closed-book QA [2]

2.1 Datasets in the English Language

WikiQA [3] The WikiQA corpus is a publicly available set of question and sentence pairs, collected and annotated for research on open-domain question answering. In order to reflect the true information need of general users, Bing query logs were used as the question source. Each question is linked to a Wikipedia page that potentially has the answer. Because the summary section of a Wikipedia page provides the basic and usually most important information about the topic, sentences in this section were used as the candidate answers. The corpus includes 3,047 questions and 29,258 sentences, where 1,473 sentences were labeled as answer sentences to their corresponding questions.

SQuAD [4] Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. SQuAD 1.1, the previous version of the SQuAD dataset, contains 100,000+ question-answer pairs on 500+ articles. SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering. Best results currently on the dataset is IE-Net (ensemble model) with EM score of 90.939% and an F1 score of 89.452

2.2 Datasets in the Hebrew Language

PARASHOOT [5] The first question answering dataset in modern Hebrew, which follows the format and crowd sourcing methodology of SQuAD, comprises approximately 3000 annotated examples, aligning with other question answering datasets in low-resource languages. The mBert model currently achieves the best results on this dataset, with an EM score of 32.0% and an F1 score of 56.1%.

hebwiki-qa [6] The Hebrew dataset used in this study is an automatic translation of the English SQuAD dataset, developed by Technion Data and Knowledge Lab. The best performing model on this dataset is the heBert model, achieving an EM score of 42.6% and an F1 score of 55.9%.

Figure 2: Corpus sample

3 Dataset

I present KolZchutQA, a comprehensive question answering dataset in Hebrew, designed in a format that follows the widely used SQuAD format. This dataset is carefully curated, consisting of a diverse range of examples that encompass various topics and fields in the rights domain. Each example in the dataset is structured as a triplet, comprising a paragraph that provides context, a thought-provoking question, and a specific span of text from the paragraph that serves as the accurate answer to the question.

3.1 Corpus

I collected random pages from כל-זכות website covering a range of topics such as worker rights and pension rights. I have created a Python script that scrapes text data from multiple web pages and store the scarped data in a pandas DataFrame. From each page the script parses the HTML content of the page using the BeautifulSoup library. After that the script searches for all the lists <ul> in the HTML content and if the length of the content is greater than 500 characters than the text of the unordered list is added to the our corpus as described in Fig.2 .

3.2 Annotation

I used the cdQA-annotator open source tool to manually annotate the questions for the dataset, following the process outlined in Fig.3. In total, I created 450 questions for the dataset. For each paragraph in the dataset, I generated up to 10 questions that are explicitly answered by the text. The answer spans were kept minimal and selected to contain the relevant information addressing the question.

Dataset	Paragraphs	Questions
Train	112	400
Validation	13	25
Test	14	25

4 Results

I will evaluate the effectiveness of the hebert-finetuned-hebrew-squad (hewiki-qa[6] best model), which is a pre-trained BERT model fine-tuned on the Hebrew- translated SQuAD dataset, by testing its EM and F1 scores. Then, I will fine-tune the model again using our KolZchutQA dataset and predict the test results to determine whether our post-finetuned model performs better than the pre-finetuned model in terms of EM and F1 scores.

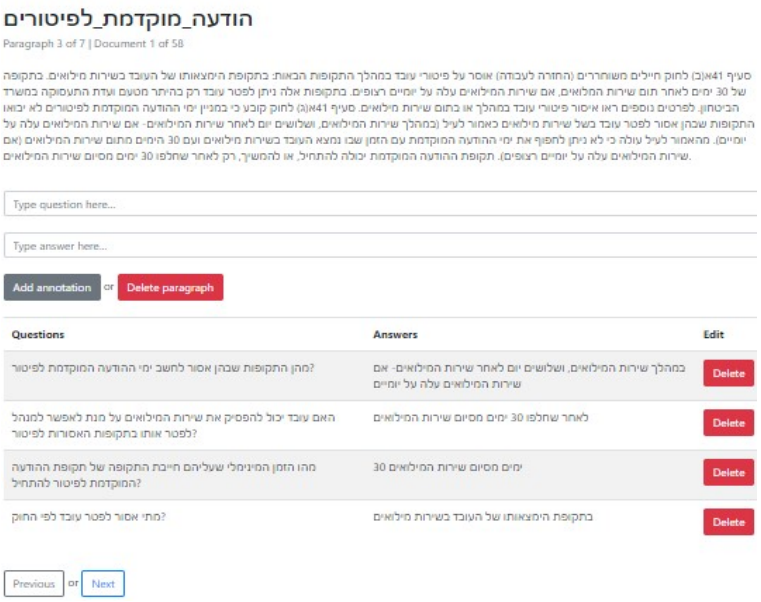

Figure 3: The Annotation process

4.1 Evaluation Methods

The advantage of using both F1 score and EM is that they provide complementary information about the model's performance. F1 score measures the trade-off between precision and recall, and thus captures the model's ability to make correct predictions while minimizing false positives and false negatives. EM, on the other hand, measures the exact match percentage, which indicates the model's ability to provide a precise and accurate response. While existing studies on Q&A models have general applicability whether in English or in Hebrew, I innovate by providing a dataset specialized in a specific domain. models that are specialized to a particular domain can be trained on more focused and curated data. This can result in higher quality training data and ultimately improve the performance of the Q&A model. F1 score is a single-line metric that combines Precision and Recall. Precision is the ratio of true positives to the sum of true positives and false positives, representing the accuracy of positive predictions. Recall is the ratio of true positives to the sum of true positives and false negatives, representing the ability to capture all actual positive instances.

$$F1score = 2 * \frac{precision * recall}{precision + recall}$$

$$ExactMatch = \frac{number - of - exact - matches}{total - number - of - predictions}$$

4.2 Baseline Model

The baselines results on the test set are EM score of 48% and an F1 score of 65.971%.
The results indicate that the task of closed-book Q&A in Hebrew NLP is challenging, with significant potential for further advancement.

4.3 Experiment Results

I fine-tuned hebert-finetuned-hebrew-squad , HeBert model trained on the hebwiki-qa corpus.
I selected the best model by validation set performance over the following hyperparameter grid:

- num_train_epochs = 4
- per_device_train_batch_size $\in \{8, 12, 16\}$.

- learning_rate $\in \{4e-6, 6e-6, 8e-6, 1e-5\}$.

The best hyper-parameters found are :
num_train_epochs = 4, per_device_train_batch_size = 8, learning_rate = 1e-5.
The post-finetuned results on the test set are EM score of 52% and an F1 score of 75.105%.

5 Discussion

Our specific domain based question answering model reached a better result compared to general based model. One key advantage of domain-specific closed book Q&A models is their ability to leverage domain-specific knowledge and context, resulting in more accurate and relevant answers. In a specific domain, the language used, terminologies, and nuances are unique, and a domain-specific model can capture and utilize this information effectively. Additionally, domain-specific models can be fine-tuned on a narrower dataset, which allows for better optimization and adaptation to the specific domain's characteristics.
Further research may include:

- Expanding the question dataset.
- Changing scraping method to generate better written paragraphs.
- Further training of the heBert fine-tuned model.
- Generating questions using a Q&A generation algorithms from corpus data .

This dataset can be a valuable tool for legal research, legal education and public legal awareness. The models trained on KolZchutQA can help people quickly find answers to legal questions and understand their legal rights and responsibilities. The dataset can also be used to create interactive learning materials and make legal information easier to access.

References

[1] Reinald Kim Amplayo et al. *Query Refinement Prompts for Closed-Book Long-Form Question Answering*. 2022. arXiv: 2210.17525 [cs.CL].

[2] Cunxiang Wang, Pai Liu, and Yue Zhang. "Can Generative Pre-trained Language Models Serve as Knowledge Bases for Closed-book QA?" In: *CoRR* abs/2106.01561 (2021). arXiv: 2106.01561. URL: https://arxiv.org/abs/2106.01561.

[3] Yi Yang, Wen-tau Yih, and Christopher Meek. "WikiQA: A Challenge Dataset for Open-Domain Question Answering". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 2013–2018. DOI: 10.18653/v1/D15-1237. URL: https://aclanthology.org/D15-1237.

[4] Pranav Rajpurkar et al. "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". In: *CoRR* abs/1606.05250 (2016). arXiv: 1606.05250. URL: http://arxiv.org/abs/1606.05250.

[5] Omri Keren and Omer Levy. "ParaShoot: A Hebrew Question Answering Dataset". In: *CoRR* abs/2109.11314 (2021). arXiv: 2109.11314. URL: https://arxiv.org/abs/2109.11314.

[6] *TechnionTDK/hebwiki-qa*. https://github.com/TechnionTDK/hebwiki-qa?fbclid=IwAR0Xbq-s1xu2gH8BS35zgFgNCeHIJ6wVZws4gqHCZ_VucbgiIngpHNTWApU. Accessed: 2023-04-30.