Domain Adaptation in Transformer models: Question Answering of Dutch Government Policies

Berry Blom[0009−0001−5001−7658] and João L. M. Pereira[0000−0002−3247−5524]

University of Amsterdam
berry96@live.nl, j.p.pereira@uva.nl

Abstract. Automatic answering questions helps users in finding information efficiently, in contrast with web search engines that require keywords to be provided and large texts to be processed. The first Dutch Question Answering (QA) system uses basic natural language processing techniques based on text similarity between the question and the answer. After the introduction of pre-trained transformer-based models like BERT, higher scores were achieved with over 7.7% improvement for the General Language Understanding Evaluation (GLUE) score.
Pre-trained transformer-based models tend to over-generalize when applied to a specific domain, leading to less precise context-specific outputs. There is a marked research gap in experiment strategies to adapt these models effectively for domain-specific applications. Additionally, there is a lack of Dutch resources for automatic question answering, as the only existing dataset, Dutch SQuAD, is a translation of the SQuAD dataset in English.
We propose a new dataset, PolicyQA, containing questions and answers about Dutch government policies and use domain adaptation techniques to address the generalizability problem of transformer-based models.
The experimental setup includes the Long Short-Term memory (LSTM), a baseline neural network, and three BERT-based models, mBert, RobBERT, and BERTje, with domain adaptation. The datasets used for testing are the proposed PolicyQA dataset and the existing Dutch SQuAD. From the results, we found that the multilanguage BERT-model, mBert, outperforms the Dutch BERT-based models (RobBERT and BERTje) on the both datasets. By introducing fine-tuning, a domain adaptation technique, the mBert model improved to 94.10% of F1-score, a gain of 226% compared to its performance without fine-tuning.

Keywords: Natural Language Processing · Question answering · Transformers · Domain adaptation · Dutch.

1 Introduction

Question answering is concerned with automatically answering questions posed by humans in natural language. We can distinguish question answering into open-domain question answering, questions about any topic; and closed-domain

question answering (questions under a specific domain). An example of an open-domain question would be "What did Albert Einstein win the Nobel Prize for?". This question is based on broad unrestricted knowledge and general ontologies. An example of a closed-domain question would be "What are my rights and obligations with a purchase agreement?", this question is asked by a citizen to the Dutch government about its policy. The responses to closed-domain questions are limited in terms of text availability, and are from a particular narrow domain. Additionally, there are two approaches to create an answer: the Generative Question Answering (GQA) generates text based on the context and the Extractive Question Answering (EQA) extracts the correct answer (a passage) from the context.

This study concentrates on closed-domain EQA concerning Dutch policy data. The importance of EQA lies in its ability to efficiently navigate large data volumes, pulling verifiable context-specific information directly from the source text. This capability is particularly vital in policy analysis, where precision and transparency are paramount. Recently, transformer based models which take raw text without almost no pre-processing and uses an attention mechanism for context, has led to advances in natural language processing tasks such as EQA [10]. Despite their advantages, pre-trained transformers models are prone to overfitting when applied to specific domains due to a large number of parameters [21]. Also, pre-training biases can result in erroneous model decisions [9].

This research aims to adapt three BERT-based models (BERTje [18], RobBERT [3], mBert [13]) by exploring domain adaptation techniques (e.g., fine-tuning) to Dutch policy data for EQA on a Dutch government policy dataset, an unexplored domain. The code and data used is made publicly available[1].

Our central Research Question (RQ) and Sub-Research Questions (SRQ) are:
RQ1 How do three BERT-based models (BERTje, RobBERT, mBert), a mix of multilingual and Dutch transformer models, and domain adaptation techniques perform in answering questions of Dutch government policies and questions translated from SQuAD dataset when compared to the Long Short-Term Memory (LSTM), a baseline model?
SRQ1.1 How does the baseline model (LSTM) effectively perform in answering Dutch questions?
SRQ1.2 How do the three BERT-based models effectively perform in answering Dutch questions?
SRQ1.3 How do the three BERT-based models with fine-tuning, a domain adaptation technique, effectively perform in answering Dutch questions?
SRQ1.4 What effect do fine-tuning the three BERT-based models using different learning rates per model layer have on the performance of answering Dutch questions?

The paper structure is as follows: Section 2 presents the related work in question answering; Section 3 details the new PolicyQA dataset; Section 4 outlines the experimental setup; Section 5 presents the results; Section 6 presents the

[1] `https://github.com/berryxmas/domain-adaptation-transformers-forQA`

discussions of the results and new findings; and finally, Section 7 summarizes the main conclusions of this work and avenues of future improvements.

2 Related Work

This section explores the related work in Question Answering. Pre-trained language models have proven to be successful at the task of Extractive Question Answering (EQA), however, generalizability remains a challenge for most of the models. Pearce et al. [11] show that the BERT model [17] performs best on the English SQuAD 2.0 dataset [14] since the context, questions, and answers are all straightforward and the answers are purely extractive.

Before transformers, the approach used for Question Answering was Long Short-Term Memory (LSTM) networks [19], which are a type of recurrent neural network that are able to learn order dependence in sequence prediction problems and is the precursor of the Transformer model. Unlike normal feedforward neural networks, LSTM has feedback connections. This way, the network can process entire sequences of data.

Wang and Jiang [20] introduced an end-to-end neural architecture for answering questions of the English SQuAD dataset. The architecture is based on a match-LSTM model. This model goes through the tokens sequentially. At each position, a weighted vector representation is obtained. The weighted vector is then combined with the current token and fed to an LSTM.

One of the first QA systems for the Dutch language was SimpleQA [5] which was capable of answering Dutch questions where the answer was a location or a person. This QA system consisted of six steps to answer Dutch questions on which the answer type is a person or a location. The question was analyzed, rewritten, retrieved with the Google API, and the best-ranked answer was picked.

Araci [1] performed research in sentiment analysis, in the financial domain. A challenging task is to perform financial sentiment analysis due to the specialized language. Araci [1] hypothesized that pre-trained language models could be used for this problem because they require fewer labels and can be further specialized towards a domain using a domain-specific corpus. He introduced FinBERT to tackle NLP tasks in the financial domain [1].

Hazen et al. [4] compared a BERT-QA model on two QA datasets: the English SQuAD dataset 2.0 [14] and the BMW automobile manual training. An interesting observation was that the model trained on the English SQuAD dataset and tested on the Auto dataset gave a lower score than when trained only on the Auto dataset.

Isotalo [6] constructed a Dutch Question Answering dataset from reading comprehension exams for Dutch secondary school students. mT5, a large pre-trained text generation model was used and that resulted in low scores even when trained on the same dataset.

Rouws et al. [16] created a new dataset, Dutch SQuAD, which is a machine-translated version of the original SQuAD v2.0 English dataset [16]. The research

Related Work	Year	Language	Domain	Task
Araci et al. [1]	2019	English	Financial context	Sentiment Analysis
Hazen et al. [4]	2019	English	BMW Automobile manual	Question Answering
Isotalo et al. [6]	2021	Dutch	Reading comprehension	Question Answering
Rouws et al. [16]	2022	Dutch	Labour Agreements (CAO)	Question Answering

Table 1. Summary of the related work in domain adaptation with domain-specific datasets.

demonstrates how to improve QA models with domain adaptation, by comparing pre-trained Dutch models, such as BERTje [18] and RobBERT [3], versus multilingual models like mBert [13].

Table 1 summarizes the existing related work in extractive QA with domain-specific datasets.

3 PolicyQA: A Dutch Government Policies Question and Answers Dataset

There is a lack of Dutch NLP resources, especially for EQA. For this reason, a new Dutch dataset is specifically designed for EQA. Government policies are long documents that are hard to read for users. PolicyQA is a challenging dataset with actual utility for the real world.

Government policies. Dutch citizens can ask questions to the Dutch government about government policies. This dataset contains the most commonly asked questions by citizens. It appeared in 2016 and the amount of questions is subject to change. The dataset is maintained by the Ministry of General Affairs and updated weekly by domain experts from the Dutch government and is publically available through an API[2].

We use the API of the Dutch government to get the fields introduction and content. The introduction is the official answer, which is constructed by domain experts from the Rijksoverheid[3], which is the Dutch government. The content is related to the introduction and gives complementary information. The introduction is expected to always be present in the content text. However, by looking at Figure 1, we can see this is not the case. The answer is not part of the content and in this case is even incorrect. To solve this problem, we merged both fields into one called context because the introduction field is not always correct and provides a longer answer. The collected text is pre-processed as all characters are set to lower-case and the HTML tags (including non-alphanumeric characters) are removed.

Additionally, to support extractive question answering, we added an extra field with the short answer to the question, which is a substring of the context field text. To fill the answer field, we manually labeled the Policy QA Dataset for the first 500 questions and answers. We specified a short answer, two to ten words from the context field to be able to perform extractive question answering.

[2] `https://www.rijksoverheid.nl/opendata/vac-s`
[3] `https://www.government.nl/#governmentnl`

Question: Does a bank have to make and keep a copy of my passport?
Annotated Answer: not obliged to make a copy or a scan of your ID.
Introduction: Banks and financial service providers are not obliged to make a copy or a scan of your ID. They are, however, obliged to check and record the details of your proof of identity. This is stated in the Money Laundering and Terrorist Financing Prevention Act (WWFT).
Content: Financial institution conducts mandatory customer due diligence. Financial institutions are required to carry out customer due diligence in certain cases. For example, if you become a customer of a bank or an insurer. Therefore, the institution checks and records your identity. The bank or insurer often makes a copy of your passport or your European identity card. (...)

Fig. 1. This question (translated to English) is about whether a bank has to make a copy of your ID or not. The answer is only in the Annotated Answer, which says "not obliged to make a copy of your ID". The Content contains a contradiction and says "The bank or insurer often makes a copy of your passport or your European identity card.".

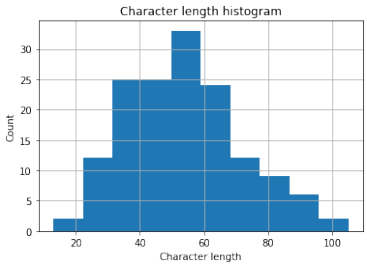

Fig. 2. Character Length for question in PolicyQA dataset

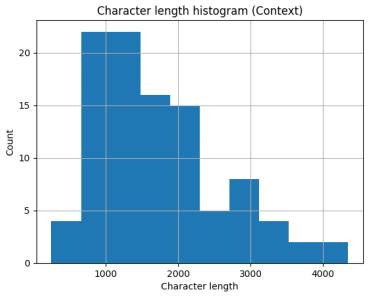

Fig. 3. Character Length for context in PolicyQA dataset

Dataset statistics. PolicyQA Dataset contains 1980 questions and contexts. The questions are relatively short, the amount of characters ranges from 10 to 110 characters. The Context is longer, the amount of characters ranges between 300 and 4400 characters and generally, it is between 800 and 1500 characters. The context contains one outlier with over 9000 characters.

Two charts with the length of characters are shown in Figure 2 for questions and Figure 3 for context.

4 Experimental setup

The PolicyQA dataset, which was introduced in Section 3, is used to test the models. As well as the Dutch SQuAD Dataset, that was obtained by machine translating the original SQuAD v2.0 dataset from English to Dutch. We divide the experiments in two parts, extractive question answering and domain adaptation.

The experiments ran inside Google Colab, which has a Tesla T4 Graphics Processing Unit (GPU).

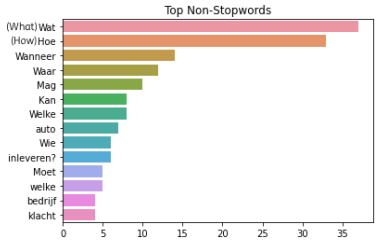

Fig. 4. Top non-stopwords for Policy QA questions

4.1 Datasets

This research uses two separate datasets: (i) the PolicyQA dataset containing government policy data and (ii) the Dutch SQuAD2. The data is preprocessed and fed to four QA models: an LSTM model, Dutch RobBERT model, Dutch BERTje model and mBert.

PolicyQA: The data is from the Dutch government, as described in Section 3, supplementary labels were annotated manually leading to 500 answers available to train and test extractive question answering. One example of the final and cleaned dataset is shown in Figure 1. This question is about whether a bank is obliged to make and keep a copy of your passport. The short answer is derived from the context and says that a bank is not obliged to make and keep a copy of your passport. Also, the answer start character is set to 42 because the short answer begins from the 42nd character.
To get insights of the types of questions in the training set, we count the questions that begin with a few common start words. As we can see in Figure 4 the most common non-stopword is What ("Wat") and a close second non-stopword is How ("Hoe").

Dutch SQuAD: The Dutch SQuAD is a translation of the original SQuAD 2.0 and contains 104.348 answers. The original SQuAD 2.0 [15] is a reading comprehension dataset which consists of questions posed by crowdworkers on Wikipedia articles where the answer is a segment of the text. The dataset also contains unanswerable questions, this is used to test if the system is capable of determining when no answer could be given. Figure 6 shows the character length of the context in the Dutch SQuAD dataset. The histogram shows that context range from 150 characters to 2300 characters and generally, it is between 600 and 1100 characters. We can observe, in Figure 5, that most common questions begin with "How", "What", and "Is".

The Dutch SQuAD dataset is split into a training set and a test set, this is a 90/10 split. This split was chosen because it was the original split from the source data. The PolicyQA dataset is also split into a training set and a test set. The training set contains 80% of the samples and the test set contains 20% of the samples. This split was chosen arbitrarily.

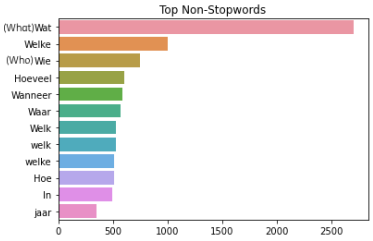

Fig. 5. Top non-stopwords for Dutch Squad v2.0 questions

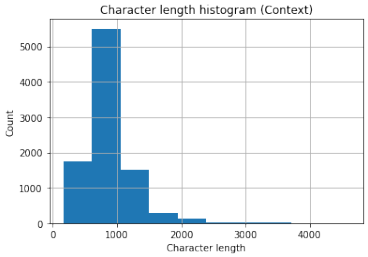

Fig. 6. Frequency of question types

4.2 Extractive Question Answering Techniques

Baseline In these experiments, we use match-LSTM, which is an adjusted version of the LSTM model. The difference between match-LSTM and LSTM is that match-LSTM contains an extra layer, which is called an answer pointer. This layer selects a set of tokens as the answer, this way used for extractive question answering. The match-LSTM sequentially aggregates the matching of each token to the weighted premise and uses the aggregated matching result to make a final prediction [20]. We also use word embeddings from GloVe to initialize the model. GloVe [12] functions as global vectors for word representation.

For the implementation of the LSTM model, we use a 3-layer bidirectional LSTM with h = 128 hidden units for both context and question encoding. The data is tokenized and lemmatized, also the training examples are sorted by length of the context and divided into batches of 32. All the hidden units of LSTM and word embedding have a dropout of p = 0.3 as in [2].

BERT- models We use the available BERT-models that support Dutch: mBert [13] is a multilanguage model trained on a diverse set of languages including Dutch; BERTje [18] and RobBERT [3] are both trained only on Dutch texts. The main difference is that BERTje uses BERT[17] base model and RobBERT uses Roberta [8], which comparatively with BERT is a larger neural network (i.e., contains more parameters) and is not trained using the next sentence prediction task. In addition to these models, we also utilized a pre-trained mBert model that was specifically trained on the Dutch SQuAD dataset. This choice was motivated by the expectation that a model pre-trained on a similar task in the same language (Dutch) would have an enhanced understanding of the language's nuances, thus potentially improving performance in our specific task. We use the Huggings Faces' PyTorch implementations[4] of the three models and the mBert trained on Dutch SQuAD.

[4] https://huggingface.co/docs/transformers/v4.20.1

4.3 Domain Adaptation

Transformer models can suffer from performance and instability, this is often the case with large models and small datasets. Therefore, in our experiments, we further train the pre-trained BERT-based models in domain data, a process called fine-tuning. To help this process, we apply hyperparameter search on the weight decay and learning rate parameters using Adam optimizer to find the set of hyperparameters that resulted in the best model performance. When we refer to fine-tuned in our experiments, it refers to fine-tuning in domain data with hyperparameter search.

Moreover, since domain adaptation tasks have benefitted from setting higher learning rates in the top layers, we add the Layer-wise Learning Rate Decay (LLRD) [22] technique as an additional step in our fine-tuning process. LLDR sets a different learning rates for each layer in the model, by decreasing its values from top to bottom layers.

The default learning rate used for all BERT-based models was 5.5 and the calculated learning rate is 3.6e-06. The default weight decay was set to 0.0 and the calculated weight decay was 0.01. However, applying these parameters in practice presented challenges. For instance, we found that the lower learning rate significantly increased the time taken for our models to converge, requiring more computational resources than we initially planned.

4.4 Evaluation metrics

To evaluate the quality of the extracted answers, we apply the following commonly used evaluation metrics in EQA:

F1-score: is a commonly used metric that by a harmonic mean combines precision and recall into a single metric. For EQA it compares the tokens between the true and the extracted answer. Precision is the number of correct tokens in the extracted answer (i.e., that appear in the true answer) divided by the total number of tokens in the extracted answer. Recall is the number of correct tokens in the extracted answer divided by the total number of tokens in the true answer.

Exact Match (EM): is calculated for a model by averaging over the individual answers. The score is either 1 or 0 per answer. If all characters of the extracted answer exactly match all characters of the true answer, then EM is 1, otherwise is 0.

5 Results

In Table 2, we report the F1-score and the EM score calculated per dataset and per model. The baseline model, LSTM, is used on the Dutch SQuAD dataset and PolicyQA. Also, the three BERT-based models and mBERT trained on Dutch

		PolicyQA		**Dutch SQuAD**	
		F1	EM	F1	EM
LSTM		22.77	1.24	30.25	17.80
BERTje	pre-trained	15.44	0.00	59.80	54.30
	fine-tuned	57.25	26.00	61.23	55.43
	fine-tuned with LLRD	27.00	6.89	60.54	55.00
RobBERT	pre-trained	14.94	0.00	47.90	39.57
	fine-tuned	56.71	20.00	52.40	43.60
	fine-tuned with LLRD	30.45	5.38	50.30	40.92
mBert	pre-trained	16.07	0.00	64.67	61.26
	fine-tuned	61.20	29.00	77.29	69.20
	fine-tuned with LLRD	36.90	8.43	68.60	64.66
mBert	pre-trained	28.88	11.00	77.29	69.20
Dutch	fine-tuned	**94.10**	**83.50**	**79.28**	**72.38**
SQuAD	fine-tuned with LLRD	85.70	78.93	78.37	71.55

Table 2. Results for EQA on Dutch SQuAD and PolicyQA dataset using LSTM and three BERT-based models (mBert, BERTje, RobBERT).

SQuAD are described pre-trained without any additional domain training, fine-tuned in same domain data, and fine-tuned using the Layer-wise Learning Rate Decay (LLRD) technique described in Section 4.3.

Testing the baseline model, LSTM, on the PolicyQA dataset resulted in an F1 score of 22.77% and an EM of 1.24. The multilingual BERT (mBert) pre-trained on the Dutch SQuAD dataset achieved the highest score with an F1 of 94.10 and an EM of 83.50. In this case, mBert was pre-trained on the Dutch SQuAD dataset and trained on the training data of the PolicyQA dataset.

Another observation we made was the increase in F1 score and EM score when the number of annotated samples increased. This was tested on for the highest performing model identified in Table 2, mBert trained on Dutch SQuAD. For every 100 samples, an 80/20 split was chosen arbitrarily for the train set and the test set. For example, for 100 samples 80 samples were used for training and 20 samples were used for testing. The samples for testing came from the last 100 samples which were never used for training the model. Interestingly, our experiments indicated that the F1 score increased linearly by approximately 3% for each additional set of 100 training samples. It is important to note that the testing samples were consistently drawn from the last 100 samples, ensuring they were never used in model training.

By fine-tuning the BERT-based models (mBert, BERTje, RobBERT) using the government policies, the F1 score improved. Also, the Layer-wise Learning Rate Decay (LLRD) technique was used during training. This technique did not result in an improvement of the F1 score.

The Dutch SQuAD was used as a second dataset. For this dataset, the LSTM model achieved an F1 of 30.25% and an EM of 17.80. The highest F1 and EM scores were obtained with mBert (F1 of 69.67% and EM of 66.26). In this approach, the model was trained on the Dutch SQuAD dataset without LLRD.

6 Discussion

Baseline - LSTM. By comparing the baseline approach with our results, we can see there is a big difference in F1 score as well as EM. In the research about the approach with Match-LSTM [20], which was described in the baseline section, a higher F1 score (69%) was achieved. However, this F1 score was achieved with slightly different data, this approach was based on the SQuAD v1.0 Dataset. The main difference between the SQuAD v1.0 dataset and the SQuAD v2.0 dataset (that originated Dutch SQuAD) is that over 50000 questions shouldn't be answered, in such case, the answer should be empty, but Match-LSTM always provides an answer. When we look at the F1 score for the PolicyQA dataset (22.77%), we can see the score is lower than 28.88. Moreover, the F1 score for the Dutch SQuAD dataset (50.23%) from our results is lower than the score obtained in the original English SQuAD V1.0 dataset [20] (77%). A possible reason for the lower score is the machine translation. When the English SQuAD dataset was translated to Dutch, the translation was not perfect [16]. This influences the output of the model.
BERT-based. When we compare our results for the BERT-based models with the state-of-the-art approaches, we can see that Dutch SQuAD had a similar performance as previous research. For example, Rouws et al. [16] achieved similar scores using mBert on the Dutch SQuAD dataset [16]. With an F1 score of 71% that is only 10 points higher than our F1 score of 61%. Thus, we can assume our approach on mBert with training is successfully executed since the scores are similar. Since the results for the Dutch SQuAD are similar to previous research [16] we can presume the F1 score of 94.10% and EM of 83.50% for the PolicyQA dataset are also as expected. Mainly because of the similar results to the known dataset, the PolicyQA dataset is a new benchmark that can be used for future work.

Generally, a pre-trained large language model performs better on more data. Even for 100 samples in the PolicyQA dataset, the model scored high (79% of F1-score). Thus, we can confirm that by training on the PolicyQA dataset, a high score can be achieved with little data.

7 Conclusions and Future Work

This research focused on the lack of Dutch resources in the field of QA and the generalizability of pre-trained large language models. This paper provides a solution to the lack of Dutch resources, namely the PolicyQA dataset, and creates new insights on domain adaptation.

In this research, we tested and evaluated a baseline for extractive question answering in order to investigate the generalizability of pre-trained large language models and examine to what extent we can make a contribution to the field of extractive question answering.
SRQ1.1: By comparing the scores to existing research, we found that LSTM scored low (F1 of 22%) on the PolicyQA dataset. One possible problem is that

the supplementary annotations in the data are not enough to train a model from scratch, these lead to bias and thus lower scores. Also, LSTM scored low on the Dutch SQuAD dataset. A possible reason for this is the machine translation from English to Dutch.

SRQ1.2: We tested the three BERT-based models without any domain adaptation technique (e.g., fine-tuning) to assess their performance in answering Dutch questions. In general, the results were low (below 17% of F1-score) for any of the models.

SRQ1.3: To understand the effect of fine-tuning the models on domain data, all three BERT-based models (BERTje, RobBERT, mBert) were evaluated with and without fine-tuning. The results show that fine-tuning leads to significant improvements in performance, e.g., with fine-tuning, mBert improves from 16% to 61% of F1 score in the PoliciQA data. Moreover, if fine-tuning is performed first in the same task and language but in a different domain (i.e., Dutch SQuAD) and then on domain data, we verify also considerable improvements using the mBert model to 94% of F1 score in PoliciQA. This is a high score, according to Lipton et al. [7]. Thus, we can conclude that the BERT-based models with fine-tuning adapt well to the Dutch government policy domain.

SRQ1.4:

We investigated if further improvements are obtained by fine-tuning with a different learning rate per layer. For that purpose, we conducted experiments with the Layer-wise Learning Rate Decay (LLRD) technique. We verify that for all models, LLDR resulted in considerable score decreases, at least 10% less of F1 score when fine-tuned without LLDR. So, for this domain adaptation task, which involves a drastic change of domain from Wikipedia (SQuAD) to Policy writings, the bottom layers required higher learning rates potentially because specific linguistic cues learned on those layers have become harder to train.

In conclusion, we evaluated how three BERT-based models (BERTje, RobBERT, mBert) perform in answering questions of Dutch government policies compared to an LSTM model. And we found that all three BERT-based models outperformed the baseline model, LSTM, with significant scores on both the Dutch SQuAD dataset and the PolicyQA dataset. We also showed that by training mBert on the Dutch SQuAD dataset and the PolicyQA dataset higher F1 scores and EM scores were achieved and that the use of LLRD did not improve the performance.

Compared to previous research, this research adds a new domain dataset, namely Dutch government policies, to the field of extractive question answering.

For future work, we suggest increasing the annotated texts of PolicyQA because we observed that the increase in the number of samples positively influences the performance. Also, other ways of domain adaptation like tuning other hyperparameters or adding top domain-specific layers without affecting pre-learned representations can be investigated using the PolicyQA dataset.

References

1. Araci, D.: FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. Master's thesis, University of Amsterdam, the Netherlands (2019)
2. Chen, D., Fisch, A., Weston, J., Bordes, A.: Reading Wikipedia to answer open-domain questions. In: ACL (2017)
3. Delobelle, P., Winters, T., Berendt, B.: RobBERT: a Dutch RoBERTa-based Language Model. In: Findings of ACL: EMNLP (2020)
4. Hazen, T.J., Dhuliawala, S., Boies, D.: Towards domain adaptation from limited data for question answering using deep neural networks. arXiv:1911.02655 (2019)
5. Hoekstra, A., Hiemstra, D., van der Vet, P., Huibers, T.: Question answering for dutch: Simple does it. In: BNAIC (2006)
6. Isotalo, L.: Generative question answering in a low-resource setting. Master's thesis, Maastricht University, the Netherlands (2021)
7. Lipton, Z.C., Elkan, C., Naryanaswamy, B.: Optimal thresholding of classifiers to maximize f1 measure. In: ECML PKDD (2014)
8. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692 (2019)
9. Nadeem, M., Bethke, A., Reddy, S.: StereoSet: Measuring stereotypical bias in pretrained language models. In: ACL IJCNLP (2021)
10. Pasch, S., Ehnes, D.: StonkBERT: Can language models predict Medium-Run stock price movements? arXiv:2202.02268 (2022)
11. Pearce, K., Zhan, T., Komanduri, A., Zhan, J.: A comparative study of Transformer-Based language models on extractive question answering. arXiv:2110.03142 (2021)
12. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: EMNLP (2014)
13. Pires, T., Schlinger, E., Garrette, D.: How multilingual is multilingual BERT? In: ACL (2019)
14. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad (2018)
15. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: EMNLP (2016)
16. Rouws, N.J., Vakulenko, S., Katrenko, S.: Dutch SQuAD and ensemble learning for question answering from labour agreements. In: BNAIC (2022)
17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NIPS (2017)
18. de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., Noord, G.v., Nissim, M.: BERTje: A Dutch BERT Model. arXiv:1912.09582 (2019)
19. Wang, D., Nyberg, E.: A long Short-Term memory model for answer sentence selection in question answering. In: ACL IJCNLP (2015)
20. Wang, S., Jiang, J.: Machine comprehension using Match-LSTM and answer pointer. arXiv:1608.07905 (2016)
21. Yu, Y., Zuo, S., Jiang, H., Ren, W., Zhao, T., Zhang, C.: Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In: NAACL (2021)
22. Zhang, T., Wu, F., Katiyar, A., Weinberger, K.Q., Artzi, Y.: Revisiting few-sample bert fine-tuning. arXiv:2006.05987 (2021)