# Question Answering on Legal Software Document

Ernesto Quevedo Caballero, Mushfika Rahman

October 19, 2022

**Abstract**

The transformer-based architectures such as BERT have achieved remarkable success in several natural language processing tasks such as question answering domain. One of the standard practices is to utilize the power of these transformer-based language models is to fine-tune them on custom datasets. Our research focuses on different transformer-based language models' performance in software development legal domain specialized datasets for the question-answering task. It compares the performance with the general-purpose question-answering task. The work shows the efficacy of fine-tuning capability transformer-based models. We have experimented with the PolicyQA dataset, conformed of documents regarding user's data handling policies, which falls into the software legal domain. We have experimented and fine-tuned the BERT [4], ALBERT [10], RoBERTa [13], DistilBERT [19] and LEGAL-BERT [2] and compare their performance on the Question answering benchmark dataset SQuAD V2.0 and PolicyQA.

## 1 Introduction

Question Answering (QA) Systems are an automated method for retrieving correct answers to questions posed by humans in natural language [6]. It is one of the most critical challenges in the natural language processing task because understanding the question and the context in which a user forms the inquiry is one of the fundamental components of QA. Moreover, with the plethora of information, the need for automated question answering systems becomes more acute since the users struggle to navigate the correct direction of their inquiries. A subclass of question answering systems is known as machine reading comprehension, whose primary goal is to retrieve answers to a given question in a single paragraph of text. The task has achieved remarkable success in the general domain using transformer-based architecture. However, previous research has demonstrated that utilizing in-domain text can benefit more from general-domain language models in specialized disciplines such as biology [11]. Thus, one can infer that utilizing legal-domain text can leverage the question-answering performance regarding legal questionnaires.

Legal documents are challenging to understand appropriately without a legal background. The challenges also lie with software companies and software privacy policies and regulations. The exponential growth of applications worldwide and monitoring of our environments, decisions, tastes, and others make it more and more important to be aware of how the data is being managed, shared, and used. Companies must include the stated information in the privacy policies of every application. The challenges lie in the characteristics of software legal documents which are longevity, ambiguity, and complexity. A high-performance Question Answering system on legal documents of software systems (privacy, policy rules) can have various implementations. One example of possible implementation is that every person can check fast if their queries have an answer in such a document before signing or agreeing to the terms and conditions.

One recommended practice in the question answering system is to leverage pre-trained embedding and fine-tuning them. The big transformer-based BERT model has a great result on general-purpose dataset SQuAD version 1 (V1) and 2. (V2) [18, 17]. There is limited research on the performance of the BERT-model on legal software datasets such as PolicyQA [1]. Since the inception of the BERT model, researchers have introduced variants of BERT, which have achieved improved performance in the general-purpose dataset. However, research on the performance of BERT variants (i.e., LEGAL-BERT) on the legal document is insubstantial [14]. Moreover, we were unable to find a study on the performance of BERT and variants of BERT model on the PolicyQA.

In this paper, we provide a study of the performance of models like BERT[4], LEGAL-BERT[2], Albert[10], and RoBERTa [13] in the SQuAD and PolicyQA dataset. Since there is limited research on the model's performance on the legal dataset, we aim to put insight into the benchmark by training on the SQUAD V2.0 dataset by ourselves. Furthermore, we compare and analyze such results, allowing us to pick the best model to obtain the best results in the PolicyQA dataset using only pretrained models.

The following section of the paper outlines related works, methodology, experiments, and conclusions.

## 2 Related Work

In recent times, researchers have dedicated significantly to the Question Answering systems in the legal domain. According to the study by [14] the best results were achieved by Deep Learning models. The very first work of [9, 5] used Convolutional Neural Networks (CNN) for the Legal Question Answering (LQA) problem. Furthermore, the use of Long-Short Term Memory with Neural Attention [15], Multi-Task Convolutional Neural Networks [25], and Match LSTM with Pointer Layer [3] was proposed by researchers.

The most recent research success in Question Answering and LQA have came from in the Neural Attentive Text Representation. Few Shot Learning in the legal domain, and diverse applications of the successful BERT model [4, 14]. However, all the proposed methods have the limitations of being poorly interpretable and require a massive amount of data for training. Furthermore, few legal datasets to train neural models at a proper scale have shown that the performance expected for LQA systems is usually worse than for generic ones [14].

There are various datasets for working with Question Answering tasks in the general-purpose domain and the Stanford Question Answering dataset (SQUAD) is most recognized because of achieving benchmark result [17]. In the legal domain, JEC-QA [26], ResPubliQA [16], JRC-ACQUIS Multilingual Parallel Corpus [21] are well recognized. In the legal domain of Software Development of Question answering datasets, PolicyQA [1] is one of the most well-known datasets in the domain. However, the study on the performance of some significant benchmarks in general Question Answering in these domain-specific datasets is limited.

## 3 Methodology

We studied and compared the performance of several BERT-related models in two Question Answering datasets. One dataset is from general domains SQuAD V2.0 and a specific domain in legal text related to software development called PolicyQA.

### 3.1 Datasets

The SQuAD V2 dataset is a reading comprehension dataset consisting of more than 100,000 questions posed by crowdworkers on a set of Wikipedia articles. The answer to each question is a segment of text from the corresponding reading passage [18, 17].

The PolicyQA dataset is a reading comprehension dataset that contains 25,017 reading comprehension style examples curated from an existing corpus of 115 website privacy policies. PolicyQA provides 714 human-annotated questions for a wide range of privacy practices [1].

Both datasets have the feature of the extractive Question Answering, where the answer is a span of text in the passage or the passage is not related to the question and does not contain the answer.

### 3.2 Models

We selected a set of the most used BERT-related models with outstanding performance in the SQuAD dataset like ALBERT, RoBERTa, and classic BERT. Additionally, we utilized the DistilBERT model because it is a cheaper and smaller model with competitive capabilities compared to other bigger BERT-based models. Thus, it is feasible to choose the model for speed during inference and usability on devices [19]. Moreover, we tested the LEGAL-BERT model, a version of the original BERT model trained from scratch with legal documents. We compared it with the other general-purpose models to see if we could get better results using a legal-based BERT in the PolicyQA dataset than others trained in general text.

The data where each model was pre-trained is the following:

**BERT, DistilBERT and ALBERT** were pretrained originally on BookCorpus [1] and English Wikipedia (excluding lists, tables, and headers).

**RoBERTa** was pretrained in the same data than BERT and also CC-News [2], OpenWebText [3] and Stories [22].

**LEGAL-BERT** was trained on documents of EU legislation from EURLEX [23], the UK legislation portal [12]; 19,867 from the European Court of Justice (ECJ), also available from EURLEX; 12,554 cases from HUDOC, the repository of the European Court of Human Rights (ECHR) [20]; 164,141 cases from various courts across the USA, hosted in the Case Law Access Project portal [24]; 76,366 US contracts from EDGAR, the database of US Securities and Exchange Commission (SECOM) [7].

## 3.3 Architecture

Span labeling models the answer extraction process, where we locate a span of text in the passage that confirms the answer. Basically, given a question $q$ of $n$ tokens $q_1, ..., q_n$ and a passage $p$ of $m$ tokens $p_1, ..., p_m$ the goal is to compute the probability $P(a|q, p)$ for each possible span $a$ as the answer [8].

The standard algorithm and also the one used in this work is to pass the question and passage to an encoder like BERT as strings separated with a $[SEP]$ token, resulting in contextual embedding for every passage token $p_i$ [8] which is shown in Figure 1. The question is represented as the first sequence and the passage as the second sequence. Next, two particular vectors are added, a span-start embedding $S$ and a span-end embedding $E$, learned and fine-tuned. Finally, a linear layer is trained in the fine-tuning phase to predict the start and end positions of the span. To obtain a span-start probability for each output token $p'_i$, we compute the dot product between $S$ and $p'_i$ and then a softmax to normalize overall tokens $p'_i$ in the passage:

$$P_{start_i} = \frac{exp(S * p'_i)}{\sum_j exp(S * p'_j)} \tag{1}$$

Analogously is computed the span-end probability:

$$P_{end_i} = \frac{exp(E * p'_i)}{\sum_j exp(E * p'_j)} \tag{2}$$

The score of every candidate span from position $i$ to $j$ is computed as $S * p'_i + E * p'_j$, and the top-scoring span which $j \geq i$ is chosen. The training loss

[1] https://yknzhu.wixsite.com/mbweb

[2] https://commoncrawl.org/2016/10/news-dataset-available/

[3] https://github.com/jcpeterson/openwebtext

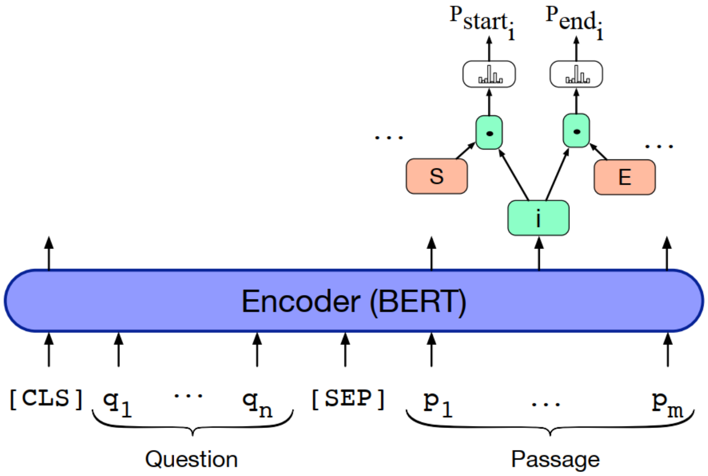

Figure 1: Standard Architecture for Question Answering based on BERT encoder

for fine-tuning is the negative sum of the log likelihoods of the correct start and end positions for each instance:

$$L = -log(P_{start_i}) - log(P_{end_i}) \tag{3}$$

A final point to address is that both datasets contain the case where the answer is not included in the passage. Therefore, estimating the probability that the answer to a question is not in the document is needed. Once again, we use the standard approach, which treats questions with no answer as having the $[CLS]$ token as the answer, and therefore the answer span start and end of the index will point at $[CLS]$.

# 4 Experiments

We conducted our experiment using the pretrained versions of the models BERT, ALBERT, RoBERTa, DistilBERT, and LEGAL-BERT. It is difficult to assess the success of a question answering task. The benchmark in the Question Answering (QA) task is evaluated on the EM(Exact Match) and F1 metrics. The EM metric refers to the percentage of predictions that exactly match any of the True Answers(labeled datasets for QA can have more than one span of text that is accepted as a correct answer to a given question). On the other hand, the F1 metric is computed over the individual words in the prediction against the True Answer. The number of shared words between the prediction and the truth is the basis of the F1 score which could be written as: $F1 = \frac{2*precision*recall}{precision+recall}$

Our selected models ran for 5 and 10 epochs on PolicyQA and SQUAD V2.0 datasets. The following table compares the models on 2 datasets and variations of epochs, and the metric for measurement is EM(Exact Match) and F1.

| | BERT | | ALBERT | | LEGAL-BERT | | RoBERTa | | DistillBERT | |
|---|---|---|---|---|---|---|---|---|---|---|
| Dataset Name | EM | F1 | EM | F1 | EM | F1 | EM | F1 | EM | F1 |
| SQUAD V2.0<br>5 epochs | 71.6 | 77.39 | 73.9 | 77.91 | 73.5 | 77.01 | **76.9** | **80.1** | 65.47 | 69.27 |
| PolicyQA<br>5 epochs | 29.5 | 56.11 | 28.76 | 57.36 | 28.08 | 54.66 | 27.23 | 54.88 | 25.42 | 52.34 |
| SQUAD V2.0<br>10 epochs | 71.7 | 75.39 | 72.71 | 77.21 | 73.95 | **77.79** | 75.18 | 74.30 | 64.79 | 68.93 |
| PolicyQA<br>10 epochs | 29.64 | 57.02 | 29.31 | 58.43 | 28.45 | 55.01 | 27.85 | 54.91 | 25.81 | 52.48 |

Table 1: Result of the models on SQUAD V2.0 and PolicyQA

## 5 Discussion

Our experimental results revealed some interesting aspects of the models. Firstly, The BERT and ALBERT model's pre-trained version performed better than the LEGAL-BERT model in SQUAD V2.0 and PolicyQA datasets. Since LEGAL-BERT was trained legal text, our assumption was that the fine-tuned version would perform better than all the models. However, our assumption did not hold. We believe that there are notable separations among subdomains even inside the legal domain. PolicyQA is legal text in the subdomain of Software Development and privacy in applications. Thus, it is feasible that a more generalized pre-trained model would produce a good result than a model trained in a legal subdomain with high separation from the Software Development legal domain. Secondly, we can observe that the increase in epochs from 5 to 10 did not significantly increase the performance of the models in SQuAD V2.0. On the other hand, in PolicyQA, the increase of epochs improves the performance of the models to some degree, which suggests that training for more epochs might still improve the results.

## 6 Conclusions

Transformer-based language models have significantly advanced the field of NLP tasks, including question answering. This work shows that BERT-related models, trained in different corpora, including legal documents (LEGAL-BERT), can also obtain good results in answering tasks related to software development legal documents using the PolicyQA dataset. Additionally, it compares the performance of each of these models in the SQuAD V2.0 and PolicyQA datasets. The results showed LEGAL-BERT did not perform better than general pre-trained models like BERT and ALBERT. Finally, the ALBERT achieved top results, which makes it a proper choice as a root and contextual embeddings encoder for a complex model design in the future.

# References

[1] Wasi Uddin Ahmad, Jianfeng Chi, Yuan Tian, and Kai-Wei Chang. Policyqa: A reading comprehension dataset for privacy policies. *arXiv preprint arXiv:2010.02557*, 2020.

[2] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*, 2020.

[3] Diego Collarana, Timm Heuss, Jens Lehmann, Ioanna Lytra, Gaurav Maheshwari, Rostislav Nedelchev, Thorsten Schmidt, and Priyansh Trivedi. A question answering system on regulatory documents. In *Legal Knowledge and Information Systems*, pages 41–50. IOS Press, 2018.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Phong-Khac Do, Huy-Tien Nguyen, Chien-Xuan Tran, Minh-Tien Nguyen, and Minh-Le Nguyen. Legal question answering using ranking svm and deep convolutional neural network. *arXiv preprint arXiv:1703.05320*, 2017.

[6] Sanjay K Dwivedi and Vaishali Singh. Research and reviews in question answering system. *Procedia Technology*, 10:417–424, 2013.

[7] EHUC. European court of human righths. `http://hudoc.echr.coe.int/eng/`, May 2013.

[8] Daniel Jurafsky and James H Martin. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

[9] Mi-Young Kim, Ying Xu, and Randy Goebel. Applying a convolutional neural network to legal question answering. In *JSAI International Symposium on Artificial Intelligence*, pages 282–294. Springer, 2015.

[10] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[11] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[12] UK legistration. Uk legal corpa. `http://www.legislation.gov.uk/`, May 2013.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[14] Jorge Martinez-Gil. A survey on legal question answering systems. *arXiv preprint arXiv:2110.07333*, 2021.

[15] Ayaka Morimoto, Daiki Kubo, Motoki Sato, Hiroyuki Shindo, and Yuji Matsumoto. Legal question answering system using neural attention. *COLIEE@ ICAIL*, 2017:79–89, 2017.

[16] Anselmo Peñas, Pamela Forner, Richard Sutcliffe, Álvaro Rodrigo, Corina Forăscu, Iñaki Alegria, Danilo Giampiccolo, Nicolas Moreau, and Petya Osenova. Overview of respubliqa 2009: Question answering evaluation over european legislation. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 174–196. Springer, 2009.

[17] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Association for Computational Linguistics (ACL)*, 2018.

[18] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[20] USA SEC. Sec. `https://www.sec.gov/edgar.shtml`, May 2013.

[21] Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *arXiv preprint cs/0609058*, 2006.

[22] Trieu H Trinh and Quoc V Le. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*, 2018.

[23] European Uninon. Legal corpa of european union. `https://eur-lex.europa.eu/`, May 2013.

[24] USCL. Usa case law. `https://case.law/`, May 2013.

[25] Guangyi Xiao, Jiqian Mo, Even Chow, Hao Chen, Jingzhi Guo, and Zhiguo Gong. Multi-task cnn for classification of chinese legal questions. In *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, pages 84–90. IEEE, 2017.

[26] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. Jec-qa: A legal-domain question answering dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9701–9708, 2020.