Using Transformers to Improve Answer Retrieval for Legal Questions

Andrew Vold
Thomson Reuters
TR Labs Research
andrew.vold@thomsonreuters.com

Jack G. Conrad
Thomson Reuters
TR Labs Research
jack.g.conrad@thomsonreuters.com

ABSTRACT

Transformer architectures such as BERT, XLNet, and others are frequently used in the field of natural language processing. Transformers have achieved state-of-the-art performance in tasks such as text classification, passage summarization, machine translation, and question answering. Efficient hosting of transformer models, however, is a difficult task because of their large size and high latency. In this work, we describe how we deploy a RoBERTa Base question answer classification model in a production environment. We also compare the answer retrieval performance of a RoBERTa Base classifier against a traditional machine learning model in the legal domain by measuring the performance difference between a trained linear SVM on the publicly available PRIVACYQA dataset. We show that RoBERTa achieves a 31% improvement in F1-score and a 41.4% improvement in Mean Reciprocal Rank over the traditional SVM.

CCS CONCEPTS

• **Information systems → Information Retrieval**; *Retrieval Tasks and Goals*; Question Answering; **Information Retrieval**; *Retrieval Models and Ranking*; Language Models; **Information Retrieval**; *Evaluation of retrieval results*; Relevance assessment.

KEYWORDS

Question Answering, Legal Applications, Deep Learning, Language Models, BERT Engines, Evaluation

ACM Reference Format:
Andrew Vold and Jack G. Conrad. 2021. Using Transformers to Improve Answer Retrieval for Legal Questions. In *Eighteenth International Conference for Artificial Intelligence and Law (ICAIL'21), June 21–25, 2021, São Paulo, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3462757.3466102

1 INTRODUCTION

Historically, when legal professionals performed natural language search, they would be required to sift through exhaustive lists of results, ranked by probability of relevance, in order to identify materials relevant to their search [18]. The task could be a time consuming and laborious effort. Over time, we began to see an interest in more focused question answering systems taking the place of traditional information retrieval systems. In the field of AI and Law, Quaresma and Rodrigues were among the first to implement a question answering system for legal documents [13], one that focused on Portuguese legal decisions. More recently, however, developments in deep learning-based approaches for tasks like open domain question answering have resulted in major gains in answer rate performance. They have also been responsible for comparable advances in closed domain question answering in fields such as Legal QA [1]. Such progress has resulted in performance gains for both factoid and non-factoid question answering.

Transformer architectures have delivered impressive performance gains over baselines for standard natural language processing (NLP) tasks. Open domain language modeling as a pretraining step, followed by domain specific fine-tuning on another domain has delivered state-of-the-art performance for tasks in a specific domain, including the legal domain. One should thus expect to see significant performance gains in legal question answer retrieval by utilizing the output of a transformer based classifier which has been fine-tuned on legal QA pairs.

It has been well observed that transformers are highly performant at answering factoid questions which typically have answers with one or a few words [5]. Transformer based research in the Legal domain has evolved toward more complex non-factoid questions which are more nuanced and may require several sentences to provide context and elaboration in order to answer the legal question at hand, for example, "When is a party entitled to a protective order?" The current work extends this research by processing a publicly available non-factoid QA dataset in an application workstream, while addressing the challenges of performance quality, speed and scale.

2 PRIOR WORK

The primary approaches employed to improve question answering search results fall into three categories: document-centric, query-centric, and ranking-centric (e.g., neural approaches). The works described below generally fall into one or more of these categories.

2.1 Open-Domain Question Answering

Open domain question answering is a task that answers factoid questions using large collections of documents [19]. Historically, retrieval in open domain QA was usually conducted using tf.idf or BM25 approaches, which match keywords with an inverted index, and represent the question and content in high-dimensional, sparse vectors [16]. In their 2017 report, Chen, et al. propose using Wikipedia for open domain question answering for factoid

ICAIL'21, June 21–25, 2021, São Paulo, Brazil

ACM ISBN 978-1-4503-8526-8/21/06.
https://doi.org/10.1145/3462757.3466102

questions [5]. The task is one of machine reading at scale, which addresses the challenges of document retrieval and machine comprehension (identifying text spans containing the answer). Their approach combines a search component based on bigram hashing and tf.idf matching with a multi-layer recurrent neural network model trained to detect answers in Wikipedia paragraphs. They use the SQuAD dataset for training and three other datasets for testing [14]. They obtain an F-score of 79%, which was within a point of the top performing method at the time.

In their work on dense passage retrieval for open domain question answering, Karpukhin, et al., show that retrieval can be effectively implemented using dense representations alone, where embeddings are learned from a small number of questions and passages via a simple dual encoder framework [9]. It has outperformed traditional QA baselines (top-20 results) by 9%-19%, while establishing new end-to-end baseline performance levels.

In their earlier work on Bidirectional Encoder Representations from Transformers (BERT), Devlin, et al. introduced a new language representation model which is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers [7]. BERT consequently can be fine-tuned with just one additional output layer to create state-of-the-art, highly performant models for a wide range of tasks, including question answering.

As an extension to BERT, Liu, et al. developed a "robustly optimized" pretraining approach to BERT known as RoBERTa [10]. They found that BERT was significantly undertrained. In their replication study of BERT, they carefully measured the impact of many key hyperparameters and training data size. They showed how hyperparameter choices have a major impact on final results. Their best model achieved state of the art results against such standard collections as GLUE, RACE, and SQuAD.

Because pre-trained language models are usually computationally expensive, and it is difficult to execute them on resource limited devices, researchers like Jiao, et al. have focused on transformer model distillation methods and proposed a novel method that was specially designed for knowledge distillation (KD). By leveraging their new KD method, while focusing on the knowledge already preserved in larger models like RoBERTa, they discovered that such knowledge could be transferred to a smaller TinyBert model [8]. The new framework captured in TinyBert performs transformer distillation at both the pre-training and task specific learning stages. They have shown that their framework ensures that TinyBert captures the general knowledge and task specific knowledge preserved in BERT.

In contrast with factoid question answering, Zhu, et al. pursued non-factoid question answering where the answers tend to be longer passages [22]. In this work, the authors determine that by generating synthetic training data of arbitrary volume and with well understood properties, the learning capacity of Knowledge Graph architectures can be better understood and characterized. Whether a given neural architecture for KGQA will train a model to generalize rather than memorize may depend on dataset properties.

2.2 Legal Domain Question Answering

In a recent work, the authors address a due diligence topic where lawyers review documents for indication of risk due to the prospect of a merger or acquisition [6]. They claim that what is novel in their approach is that the proposed system explicitly handles the imbalance in the data, by generating synthetic instances of the minority answer categories, using the Synthetic Minority Oversampling Technique [4]. This ensures that the number of instances in all the classes are roughly equal to each other, thus leading to more accurate and reliable classification. They use conditional random fields as their text selection algorithm. Each sentence in the contract under consideration is featurized into a tf.idf vector and fed into the CRF algorithm. The authors found a 13% improvement in accuracy due to the imbalance handling.

The recently published work on Legal BERT has reported on performance gains on an assortment of downstream NLP tasks [3]. The authors compare the performance of out of the box BERT with a version that benefits from additional pre-training with legal domain data, and finally with a version where the pre-training with legal domain data starts from scratch. The legal domain training data consists of UK and EU legislation, European Court of Justice and Court of Human Rights cases, and finally U.S. court cases as well as U.S. contracts. The authors show that the best strategy to transfer BERT to a new domain may vary, but that one may consider either further pre-training or pre-training from scratch on data from the new domain. Legal BERT achieved state-of-art results in three end-tasks, and, most notably, the performance gains were stronger for the most challenging end-tasks (i.e., multi-label classification in ECHR-cases and contract header & lease details in Contracts-NER) where in-domain (legal) knowledge is arguably the most important. The authors also released a version of Legal BERT-SMALL, which is 3 times smaller than Legal BERT, but quite competitive performance-wise to the other versions of Legal BERT.

Reports on question answering systems have also recently been published by researchers at Thomson Reuters and LexisNexis [2, 11]. The current work demonstrates the robustness of a Legal QA system deployed in a multi-stage workstream where the engine is fine-tuned on an application-specific dataset. The application and dataset are discussed below. The system is shown to significantly outperform the baseline using contemporary neural techniques.

3 METHODOLOGY

Transformer models have achieved state-of-the-art performance in many NLP applications such as text classification, text summarization, question answering, etc. Though transformers are highly performant, their generally large size make them difficult to deploy in production systems. Successful transformer model hosting in a production environment would be a major advance in natural language applications. For this reason, we developed a high performance question answering (QA) system based on the RoBERTa base architecture, but other transformer architectures could be used as well [10, 12]. The challenges and our strategies for handling these problems will be discussed later in this section.

QA system researchers do not frequently have access to evaluated QA pairs that are broad, balanced, and comparable to what a user would ask. Open sourced QA pairs tend to be either very general or belong to a niche domain. If one is fortunate to have access to labeled QA pairs in the working domain, it is unlikely that there is enough data for broad topic coverage. To address this issue, subject matter experts (SMEs) can be hired to procure quality QA pairs. Yet

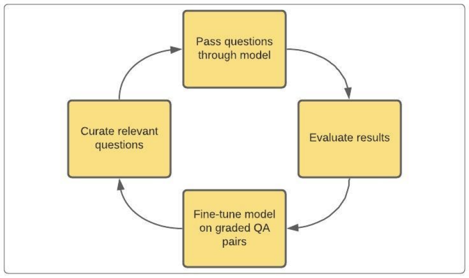
Pass questions through model
Evaluate results
Fine-tune model on graded QA pairs
Curate relevant questions

Figure 1: QA System Development Cycle

SMEs often experience fatigue when producing examples, even if the queries are from user query logs. This phenomenon often manifests itself in the form of weak question-answer pair generation where examples differ by only a few words. To address such limitations, natural language user queries are identified, run through the classifier, and the highest scoring QA pairs are evaluated. The resulting data can then be used to train the model, resulting in a cyclic data curation, model training process as seen in Figure 1.

Given the QA system that we developed was intended for application to sets of in-house legal documents, many of which are not freely available to the general public, for the purposes of this research report, we have opted to apply our techniques to the publicly available legal questioning collection described in section 3.2. Though it covers a subdomain of the legal space, it is nonetheless a broad ranging and complex dataset that contains an array of topics, question and answer lengths and types. It is a nuanced and challenging set of data which is indicative of the kinds of question and answer types one can expect to see in the legal domain. The findings we obtain apply specifically to the PRIVACYQA dataset, but are also representative of the kinds of issues and challenges one encounters with wider-ranging legal datasets as well.

3.1 Training Targets

In order to assess the performance of the QA classifier, natural language user log queries and their retrieved answers are presented to an SME. The SME then must determine whether or not the top answers returned by the classifier satisfy what was being asked. The grade by the SME can be a binary "pass/fail", a letter grade, or even a score on a continuous scale. In our case, the grade is converted into a label or regression target to be used for model fine-tuning.

For our internal QA classifier, we utilized a multi-label grading criteria which determined whether or not the answer satisfies the requirements and to what degree it answers the given question. In order to avoid grader bias, we have two SMEs grade each QA pair, and the average is taken. Disagreements of more than one grade may be adjudicated by a senior SME. A similar approach was used by the creators of the PRIVACYQA dataset, which will be explained later in this section.

3.2 Data

The dataset used in these experiments comes from the PRIVACYQA dataset described by Ravichander, et al. in [15]. It is a corpus consisting of 1,750 questions about privacy policies associated with mobile applications [20], and more than 3,500 relevant answers that have been annotated by experts. From the data provided, we have obtained approximately 130K passages for our training set, of which about 25% was used in our validation set. The goal of the collection was to achieve broad coverage across a spectrum of application types. The researchers collected privacy policies from 35 mobile applications representing different categories in the Google Play Store [17]. Another goal of the creators was to include both policies from well-known applications, which are likely to have carefully-constructed privacy policies, and lesser-known applications with smaller install bases, whose policies might be considerably less sophisticated. They set a threshold of 5 million installs to ensure each category includes applications with installs on both sides of the threshold. All policies in the corpus are in English, and were collected before April 1, 2018, predating many companies' GDPR-focused revisions.

3.2.1 Answer Identification. In order to identify legally valid answers, seven subject matter experts with legal training were recruited to formulate answers to the Amazon Mechanical Turk questions. They indicated relevant material within the given privacy policy in addition to supplying relevant metadata regarding the question's relevance, subjectivity, OPP-115 category [21], and how likely any policy is to containing the answer to the question.

Table 1 presents aggregate statistics for the PRIVACYQA dataset. 1750 questions are posed to an imaginary privacy assistant over 35 mobile applications and their associated privacy documents.

Dataset	Train	Test	All
No. of Questions	1350	400	1750
No of Policies	27	8	35
No. of Sentences	3704	1243	4947
Avg. Q Length	8.42	8.56	8.46
Avg. Doc. Length	3121.3	3629.13	3237.37
Avg. Ans. Length	123.73	153.44	139.62

Table 1: Statistics of the PRIVACYQA Dataset

4 EXPERIMENTS

To demonstrate the quality of answer retrieval performance of a transformer in comparison with traditional ML models, we fine-tune an open domain pretrained RoBERTa classifier and train a linear SVM with tf.idf features on the PRIVACYQA dataset. Training models on this dataset is challenging for several reasons. First, the dataset is largely unbalanced with negative examples occurring 25 times more often than positive examples. In addition to this, there exists considerable noise in both the queries and the answers. Finally, the number of unique questions and answers are far fewer than the total counts of QA pairs in the dataset.

Class imbalance is a common problem in real world machine learning applications. For this reason, there are many methods to effectively combat the adverse effects of training on an imbalanced dataset. These can include over/under sampling, class weighting on the loss, external or generated training data augmentation, and more. For our experiments, we apply a simple class weighting scheme to give more weight to the underrepresented positive class.

The PRIVACYQA data is quite noisy. The queries and answers are riddled with misspellings, URLs, improper grammar, fragmented sentences, lack of punctuation, and more. In order to have the data resemble the data existing in our internal system, significant data cleaning and filtering is applied. This includes capitalizing sentence beginnings, removing URLs, removing queries or answers with more than 4 non-english words, and additional cleaning and filtering steps. Even after all of this data preprocessing, the data remains far from perfect, but is sufficient to meet the requirements of our experimental conditions.

The original PRIVACYQA paper split the training and testing datasets by privacy category, rather than by unique queries. Therefore, original PRIVACYQA dataset contains data leakage. Several queries from the test set can also be found in the train set. In order to rectify this, we identify the queries which exist in the training set, and reassign those QA pairs as training data, which can be seen in Table 2.

Set	Positives	Negatives	Total
Train	6,950	152,903	159,487
Test	5,276	45,493	50,720

Table 2: Dataset Split Statistics

We perform tf.idf fitting on the unigrams and bigrams from the corpus of unique answers, and use it to vectorize the QA pairs which are then used as inputs to a linear SVM. The hyperparameters of the SVM are found by performing 5-fold cross validation via grid searching with maximizing the validation set F1-score as the objective. This process leads to optimal hyperparameters for the SVM model and a consistent training-validation split to be used for training RoBERTa.

Due to the large number of parameters in RoBERTa, it is trained by gradually unfreezing the layers, starting with the classification head. The learning rate and the batch size are decreased as layers are unfrozen, as to avoid overloading the CUDA memory. After each epoch, the validation F1-score is measured until a plateau is reached, at which point the model loses generalizability.

5 RESULTS

After training both RoBERTa and SVM classifiers, the models are run over the test set to determine the performance gain of using a transformer based QA classification engine. The results can be seen in Table 3.

Metric	SVM	RoBERTa
Precision	0.212	0.470
Recall	0.480	0.326
F1-score	0.294	0.385*
MRR	0.074	0.105**

Table 3: Classifier Performance on the Test Set

As seen in the table, RoBERTa outperforms the SVM for all metrics, except for recall. This makes sense because the SVM looks for exact token matches between the query and answer to assign a positive label. RoBERTa, however, uses the latent representation of the tokens to identify potential answers. In any QA application, it is important to serve an expansive set of quality answers, for this reason, RoBERTa is preferable to the SVM for its 31% improvement in F1-score over the SVM ($^{*}\, p < 1 \times 10^{-5}$).

One of the most important metric for QA classification systems is the Mean Reciprocal Rank (MRR). This simple metric is the average inverse position of the true labeled examples in the answer pool. MRR is a useful metric for ensuring that the highest quality answers make it to the highest rank in the list. This is especially important for applications like question answering which may return a few or even one answer for a particular query. Due to the importance of MRR, RoBERTa is the better choice for a QA model with a 41.4% improvement in MRR over the SVM baseline ($^{**}\, p < 1 \times 10^{-5}$).

It is interesting to see that a simple, traditional ML model operating on sparse word vectors achieves performance relatively similar to that of a transformer. One explanation of this could be due to the fact that the data is very messy and lacks uniqueness. A simple ML model doesn't get distracted by nuances of this dataset such as: fragmented sentences, misspellings, and the frequent use of URLs and company names. A simple ML model is also less prone to overfitting than a transformer, especially considering the redundancy of the text in the dataset. Overfitting was a challenge during experimentation. For this reason, one can expect even higher RoBERTa performance if the experiments are repeated with a more sophisticated strategy for combatting overfitting. A major lesson learned from running this experiment is to ensure that the data used for training a transformer QA classifier is clean and without redundancy. In addition to this, more careful domain adaptation could be applied before fine-tuning on the experimental dataset.

6 APPLICATION PIPELINE

Developing a strong QA classifier is only one piece of deploying a scalable QA application. It is not feasible to simply concatenate all passages from a corpus to a user's query and sequentially feed them to a classifier. Instead, there needs to be a way to quickly filter out obvious negative passages, yielding a smaller pool of potential answers to be fed to the classifier. An additional challenge of using a transformer based classifier like RoBERTa is its size and latency. In order to address these challenges, we propose a solution consisting of a parallel cluster for candidate retrieval (Stage 1) and RoBERTa operating on a GPU endpoint (Stage 2). In addition, in order not to overwhelm the user of the application, we typically return the top n answers as predicted by RoBERTa, where n is small.

One of the most important requirements for a powerful QA classification engine is to have a sufficiently large corpus of passages against which a query can be compared. Oftentimes, this can be on the scale of hundreds of thousands to millions of passages. The overwhelmingly vast number of passages is irrelevant to a particular query, and these are not difficult to identify. For this reason, it is a good idea to have a computationally efficient method of removing the obviously irrelevant passages before performing any QA inferencing. In addition, due to the scale of the data, it is imperative to perform this filtering in parallel. To accomplish this, we employ a parallel data cluster in the cloud with our data spanning several nodes (See Figure 2). The cluster functions by serving up the top-n most relevant passages as determined by properties such as term overlap between the query and passages. It is up to the application designer to determine the appropriate number of passages to include in a candidate pool. Most often, a candidate pool size between 100 and 1000 suffices. Increasing the number of nodes decreases latency but increases cost, so application engineers must decide in advance on how many nodes to include in their cluster.

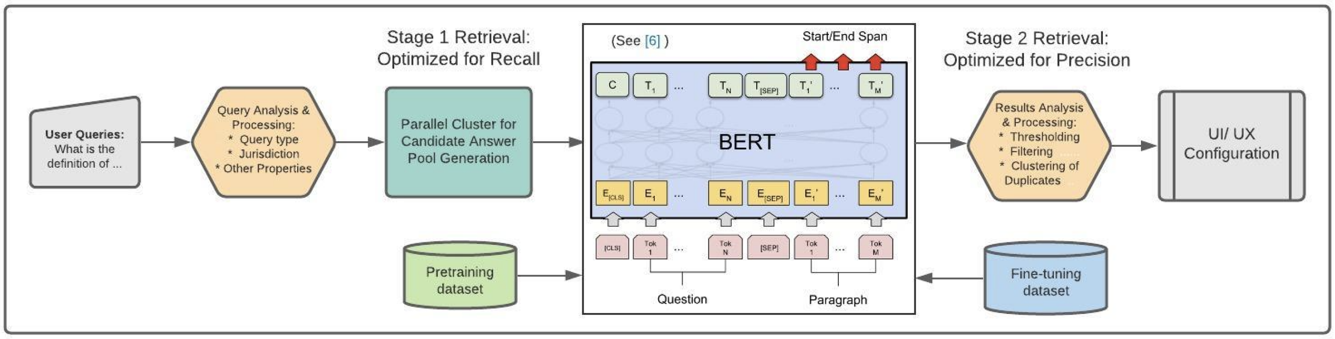

Figure 2: QA Application Pipeline

After a more narrow candidate pool has been retrieved, the QA pairs are then tokenized, pushed to the CUDA device, and fed to the classifier. The classifier returns a list of prediction scores of the relevance of the passage to the answer. The passages associated with these predictions are then sorted, and the top-n are returned, where n is determined by the application development team. One may wish to apply a RoBERTa score threshold, so that very low predictions, which are very often negative, are not shown to the user. If executed properly on the appropriate hardware, the entire answer serving process can take a second or less to perform.

7 CONCLUSIONS

Question answering is a challenging task which has been in development for many years. Question answering can take on different forms such as answer generation, answer snippet retrieval, and question answer classification. We propose an end-to-end pipeline which combines the speed of a parallel data retrieval mechanism with the classification power of a fine-tuned RoBERTa Base classifier. Our observations from our internal data and the data discussed in this paper indicate that transformer architectures can achieve greater classification performance than traditional machine learning methods in legal QA classification tasks.

We have discussed the efficacy of transformer models in text classification tasks. We observe a significant increase in F1-score and MRR of a RoBERTa classifier over a linear SVM on the PRIVACYQA dataset. Our experiment has shown that transformer models can achieve superior performance over traditional machine learning techniques in legal question answer classification.

We have also also discussed some of the challenges and solutions associated with developing and operating a transformer based question answer classification system. With a large set of content, subject matter experts, and sufficient computing power, it is possible to train and operate a transformer based system in a cost effective manner.

REFERENCES

[1] S. Badugu and R. Manivannan. A study on different closed domain question answering approaches. *Int. J. Speech Technol.*, 23(2):315–325, 2020.

[2] Z. Bennett, T. Russell-Rose, and K. Farmer. A scalable approach to legal question answering. In *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law*, ICAIL '17, pages 269–270, New York, NY, USA, 2017. Association for Computing Machinery.

[3] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos. Legal-bert: The muppets straight out of law school, 2020.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002.

[5] D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions, 2017.

[6] R. Chitta and A. K. Hudek. A reliable and accurate multiple choice question answering system for due diligence. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, ICAIL '19, pages 184–188, New York, NY, USA, 2019. Association for Computing Machinery.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351, 2019.

[9] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. t. Yih. Dense passage retrieval for open-domain question answering, 2020.

[10] Y. Liu, M. O., N. Goyal, J. Du, M. Joshi, D. Chen, O. L., M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[11] G. McElvain, G. Sanchez, S. Matthews, D. Teo, F. Pompili, and T. Custis. West-search plus: A non-factoid question-answering system for the legal domain. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1361–1364. ACM, 2019.

[12] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling, 2019.

[13] P. Quaresma and I. Rodrigues. A question-answering system for portuguese juridical documents. In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, ICAIL '05, pages 256–257, New York, NY, USA, 2005. Association for Computing Machinery.

[14] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[15] A. Ravichander, A. W. Black, S. Wilson, T. B. Norton, and N. M. Sadeh. Question answering for privacy policies: Combining computational and legal perspectives. *CoRR*, abs/1911.00841, 2019.

[16] S. Robertson and H. Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, apr 2009.

[17] P. Story, S. Zimmeck, and N. Sadeh. Which apps have privacy policies? In M. Medina, A. Mitrakas, K. Rannenberg, E. Schweighofer, and N. Tsouroulas, editors, *Privacy Technologies and Policy*, pages 3–23, Cham, 2018. Springer International Publishing.

[18] H. R. Turtle. Text retrieval in the legal world. *Artif. Intell. Law*, 3(1-2):5–54, 1995.

[19] E. M. Voorhees. The trec-8 question answering track report. In *Proceedings of TREC-8*, pages 77–82, 1999.

[20] D. Weissenborn, G. Wiese, and L. Seiffe. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[21] S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. Giovanni Leon, M. Schaarup Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell, T. B. Norton, E. Hovy, J. Reidenberg, and N. Sadeh. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[22] M. Zhu, A. Ahuja, D. Juan, W. Wei, and C. K. Reddy. Question answering with long multiple-span answers. In T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 3840–3849. Association for Computational Linguistics, 2020.