Question Answering System for Legal Documents using Transfer Learning

Anonymous EMNLP submission

Abstract

Question Answering Systems (QAS) have become more popular than the traditional search engines by providing relevant and concise answers to domain-specific questions. Moreover, the Legal Domain comprising of long and difficult-to-comprehend Legal Acts, benefit significantly from such answering systems. Therefore, we intend to present a closed domain QAS for Legal Documents, particularly the Indian Legal Acts, using the Transfer Learning approach. The proposed system is designed to pre-process the overlong Legal Acts into small yet significant contexts, for answering the posed questions using the state-of-the-art BERT Model, fine tuned on an Indian Legal Dataset that consists of human-generated and system-generated questions, containing around 2250 questions. Additionally, the Question Generation (QG), which is an auxiliary component of the system, developed using Google's T5 model, demonstrates the automatic generation of system-generated questions based on the Indian Acts for the Legal dataset. The Legal QAS produces reliable results by answering the questions from the Legal test set, containing human generated questions, with an Exact Match (EM) of 77.38 and a F1-score of 76.26.

1 Introduction

The Question Answer System (QAS) is one of the key tasks of Natural Language Processing (NLP), with wide applications in language based AI Systems. In recent years, a significant progress has been made in QA automation through the advancement of end-to-end DL techniques (Hermann et al., 2015). Transfer Learning is one such technique that attempts to adapt the knowledge gained from one task and apply it to another related task in order to either increase efficiency or reduce the size of the domain-specific dataset. Recent state-of-the-models (Vaswani et al., 2017) like the BERT (Devlin et al., 2018) and its variants have proved quite evident and useful for Transfer Learning. Question Answering Systems provide a solution to various complex domains making them easy-to-understand. Legal Domain is one such complex domain that comprises Legal Documents with intricate terminologies and subtle language which are difficult to comprehend even for legal practitioners. This paper presents a solution by building a QAS for Legal Documents, particularly the Indian Legal Acts.

The paper reviews various BERT variants for the Answering System and compares them based on an evaluation on a test set of the Legal Dataset, consisting of both machine-generated questions and human-generated questions. The pre-trained BERT Model, fine-tuned on this Legal Dataset is ultimately used in the Answering System. The legal dataset, which uses a format similar to the SQuAD Dataset (Rajpurkar et al., 2016, 2018), is created using Legal Acts that are amended by the Indian Constitution and Judiciary.

2 Methodology

The proposed system has three components: Preprocessing and Context Creation, Question Generation (QG) and Question Answering.

2.1 Pipeline

2.1.1 Preprocessing and context creation

Indian legal acts are available in PDF format at India-Code [1], a database of all Central enactments. A typical legal act consists of the title of the act, index pages, chapters, sections, subsections and footer notes. The relevant information is found in the sections and subsections which serve as good context. As a result, pre-processing is needed to filter the data and ensure that only the most relevant content is extracted.

The steps are as follows : 1) The PDF format data is converted into plain text. 2) The first main content page number is then determined by comparing the title on the index page with the title on

the first main content page. This is done in order to remove the index pages and keep only the content pages as they don't provide any significant information. 3) The title, footer notes and the chapter headers are discarded using regular expression matching. 4) The content from all of the pages is then concatenated to create a single text data entity and this text is filtered to delete any unnecessary symbols. 5) The text is then divided into chunks sections of the act. If the context is longer than 350 words, we break it again into chunks of less than or equal to 350 words, retaining the subsections boundary for each chunk..

2.1.2 Question Generation

Question Generation is an auxiliary component of our system to demonstrate generation of machine-generated questions in absence of the actual user query in real time and also to build a SQuAD-like legal dataset required for fine-tuning. Google's T5-base model (Raffel et al., 2019) is used for this purpose [2][3]. The pre-processed contexts generated as an output from the preprocessor are given as input to QG which finds possible answer spans and generates questions for those spans by considering the specified contexts. These machine-generated question-answer pairs together with human-generated question-answer pairs are used for generating the SQuAD-like Legal Dataset.

2.1.3 Question Answering System

The BERT-large Model [4], fine-tuned on the Legal Dataset, is the main component in the QAS. The answer to a question is fetched using the technique of batch processing. A batch of 32 contexts along with a parallel list of questions is provided as input to the system. If there are more than 32 contexts, then a series of batches are processed by the system. For a batch input, the question is repeated to form a list of questions such that for each context in the batch of contexts, there is one question item to process parallely. The model processes a batch and generates a list of start and end scores for each context. Based on the output scores, the system finds the best answer from every context. The best context answer is determined using a final score which is calculated as the sum of the start and end scores for the answer : 1) The token indexes giving the top 10 start and end scores are fetched. 2) All start-end combinations between these top 20 indexes are applied and the final scores of each valid answer is recorded. 3) The answer with the highest final score is returned (along with the final score) as the answer from the context.

The system combines all such best context answers to find the best answer from among all contexts from all batches, using their final scores. The answer is reported only if the final score of the best answer crosses the score threshold(+2.5) set for the system. The threshold score determines the minimum score required by an answer to be relevant to the question and shows a confidence in its content.

2.2 Fine-Tuning

To achieve better results on the task of Answering Questions on the Legal Acts, the pre-trained model was fine-tuned on the Legal Dataset consisting of about 2500 questions. The Legal Dataset was split in the percentage of 80% train set, 10% validation set, and 10% test set. The BERT Large Model was trained using the Trainer Module with a batch size of 16 dataset items. The training proceeded in 4 epochs at a recommended learning rate of $2e-5$. The training and validation loss were recorded after each epoch completion which are discussed in the Results section.

2.3 Evaluation of Fine-Tuned Model

After fine tuning, the model was evaluated using three performance metrics as discussed below. The performance is calculated with reference to a test set containing an equal measure of Machine Generated and Human Generated Questions. The answers predicted by the model are compared against the reference answers in the test set and based on the number of common words between the two, the model is evaluated.

3 Review of alternatives

In this section we review the alternatives, primarily the BERT variants that we tested for choosing the best model for QAS task. In particular, alternatives were :

3.1 ALBERT

ALBERT (Lan et al., 2019) is a simplified version of BERT that has fewer parameters than BERT. To minimise the number of parameters, it employs the cross-layer parameter sharing and factorized

[1] https://www.indiacode.nic.in/
[2] T5-based Model
[3] T5-based fine-tuned Model
[4] BERT Model

embedding layer parameterization techniques. It is presented in 4 different configurations, with the parameter size and number of hidden layers increasing with each configuration.

3.2 RoBERTa

RoBERTa (Liu et al., 2019) is a BERT variant that has been robustly optimised using a pre-training approach. RoBERTa uses dynamic masking and is trained only with the MLM (Masked Language Modeling) task. Also, RoBERTa is pre-trained with a large batch size [160 GB] and it uses byte-level BPE (BBPE) as a tokenizer. RoBERTa is presented in 2 configurations, RoBERTa-base and RoBERTa-large, where each BERT configuration has been pre-trained with this approach.

3.3 DistilBERT

DistilBERT (Sanh et al., 2019) is a smaller, faster, cheaper, and lighter version of BERT. The ultimate idea of DistilBERT is to take a large pre-trained BERT model (teacher BERT) and transfer its knowledge to a small BERT (student BERT) through knowledge distillation (Hinton et al., 2015).

Evaluating these models with each configuration, it was clearly observed that, even though the variants were trained to improve the BERT's performance by applying various architectures and approaches , these models did not show as promising results as the BERT-large model. So, we adopted BERT LARGE for the Question Answering component of our system. Each model had different runtime, depending on its number of parameters. The average runtime for fine-tuning BERT-large was about 30 minutes on Google Colab platfrom.

4 Results

Experiments were carried out by fine-tuning the BERT model and its mentioned variants, in all of their configurations using the generated legal text dataset. The performance of each model was evaluated using the metrics discussed in next sections.

4.1 Evaluation metrics

4.1.1 Weighted Accuracy (A_w)

Weighted Accuracy defines how accurate the model works. It is defined as the ratio of sum of weights of the answers of the questions to the total number of questions, expressed as a percentage. For each question, a weight between 0 and 4 is chosen on the matched percentage. The weight allocated to each answer is determined by how closely the predicted and actual answers align.

1. **Weight 0**: *Irrelevant Answer* - If match percentage is less than 25%.
2. **Weight 1**: *Wrong Answer* - If match percentage is greater than or equal to 25% and less than 50%.
3. **Weight 2**: *Correct Answer* - If match percentage is greater than or equal to 50% and less than 75%.
4. **Weight 3**: *Specific but Incomplete Answer* - If match percentage is greater than or equal to 75% and less than 100%.
5. **Weight 4**: *Complete Answer* - If match percentage is equal to 100%.

The weighted Accuracy (A_w) can be calculated as :

$$A_w = \frac{w_1 + w_2 + ... + w_n}{max(w_i) * N} \quad (1)$$

where: w_i = Weight assigned to an answer i
N = Total number of questions

4.1.2 F1 Score

F1 is defined as the harmonic mean of precision and recall.

4.1.3 Exact Match (EM)

Exact Match (EM) calculates the accuracy based on the number of exact answers provided by the model. It is defined as the percentage of questions which are answered perfectly.

4.2 Results and discussions

The results obtained on experimenting various BERT variants, after fine-tuning them on the Legal Dataset, are represented in Table 1. Three performance metrics values were calculated for all the models using the test set. For every evaluation metric, the BERT-Large model outperformed all the other models and answered most of the answers completely (See Appendix). It showed reliable results with an EM of 77.381%, Weighted Accuracy of 84.2262% and a F1-score of 76.259.

The reasons for such results may be because the variants of BERT were developed with an aim to optimize or to get a light weight version of BERT while trying to attain the similar accuracy as that

Model	W0	W1	W2	W3	W4	A_w	F1	EM
ALBERT-base	30	22	11	9	96	67.7083	57.8014	57.1429
ALBERT-large	29	15	15	15	94	69.3452	66.8684	55.9524
ALBERT-xlarge	24	21	9	5	109	72.9167	68.0631	64.881
ALBERT-xxlarge	23	16	9	6	114	75.5952	69.2978	67.8571
BERT-large	**14**	**9**	**8**	**7**	**130**	**84.2262**	**76.2579**	**77.381**
DistilBERT	30	22	11	6	99	68.1548	60.078	58.9286
RoBERTa-base	34	22	13	8	91	64.881	62.6461	54.1667
RoBERTa-large	32	18	10	6	102	69.0476	62.9649	60.7143

Table 1: The table summarizes the exact numerical metrics values achieved while evaluating the fine-tuned BERT model and its variants on Legal dataset, were tested on the test dataset. Note : The A_w and EM metric values are mentioned in percentage.

of BERT. But in doing so, the performance or accuracy of those models is compromised to a some extent depending on the methodologies used.

As we go from ALBERT-base to ALBERT-xxlarge, the evaluation metrics values go on increasing because number of parameters increases. BERT-Large performs better than ALBERT-xxlarge. This can be because BERT-Large has more number of parameters (340M) than ALBERT-xxlarge and also due to the methodology followed by ALBERT while training as reviewed above. DistilBERT shows relatively bad results compared to other models due to usage of knowledge distillation methodology which works by extracting only essential and predominant aspects leaving the minute details which is most required in legal domain. Further, RoBERTa also shows unsatisfactory results as it is an optimized BERT and due to the training methodology it follows.

Parameters	Pre-Trained	Fine-tuned
Precision	0.9364	0.9353
Recall	0.6429	0.6437
F1-score	76.235	76.2579
A_w	83.9286	84.2262
EM	76.1904	77.381

Table 2: The table mentions values when BERT, pre-trained on SQuAD dataset and BERT, fine-tuned on Legal dataset were tested on the test set.

Fine-Tuning is a process which updates and adjusts the parameters of the model to increase the performance of the model in solving a particular problem, making them more task-specific. The fine-tuning of the BERT Pre-Trained Model displayed improved results in answering the questions of the Legal Domain as shown in the table 2. The F1 score increased in the evaluation of the Fine-Tuned BERT, largely due to the increase in the Recall of the answers by the model. The model learnt to answer questions with a few more details as in the reference answer, as expected.

For example, for a question - "*In what form can we obtain the information under the RTI Act?*", the reference answer was - *diskettes floppies tapes video cassettes or in any other electronic mode*. The BERT Pre-trained model answered - '*diskettes floppies tapes video cassettes*' while BERT Fine-Tuned answered - '*diskettes floppies tapes video cassettes or in any other electronic mode*'. The number of words common between the reference and predicted answer were more for the Fine-Tuned Model as it learnt to answer the questions specific to the Legal Domain during its training.

5 Conclusion

This work aimed to build a robust Question Answering system for legal documents, particularly the Indian legal acts. The modern neural models, advanced in the way they are trained, and the massive datasets on which they are trained, enable them to understand the general English language model and apply it to various downstream tasks. The system employs the Transfer Learning approach to solve the issue by using a pre-trained BERT model that has been fine-tuned on a Legal Dataset. The generated legal dataset consisting of a pool of machine-generated and human generated questions, incorporating a diversity in the question types gave an additional advantage for fine-tuning. BERT variants with all the configurations were also analyzed quantitatively and the BERT Large model proved to be the best choice for the system, providing correct and relevant answers to the legal questions.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.