Legal Domain Question Answering System

NLPProjectReport

FALL2023

By

Shashank

Nandaki

12/5/2023



# Abstract

Although Q&A systems have become commonplace in many businesses, the legal domain has proven challenging because of the abundance of specialised information. Retrieval-based methods have demonstrated potential as more substantial pretrained language models become available. Creating a retrieval-based system to answer medical questions is the aim of this research. We do this by extending our knowledge using massive language models and graphs. First, we efficiently obtain a huge yet coarse set of replies using Elasticsearch. Next, we integrate semantic matching with pretrained language models to obtain a fine-tuned ranking, leveraging knowledge graphs and named entity recognition to leverage the relationship between the entities in the query and answer. We provide an extensive analysis of the legal reading comprehension test as well as this dataset.

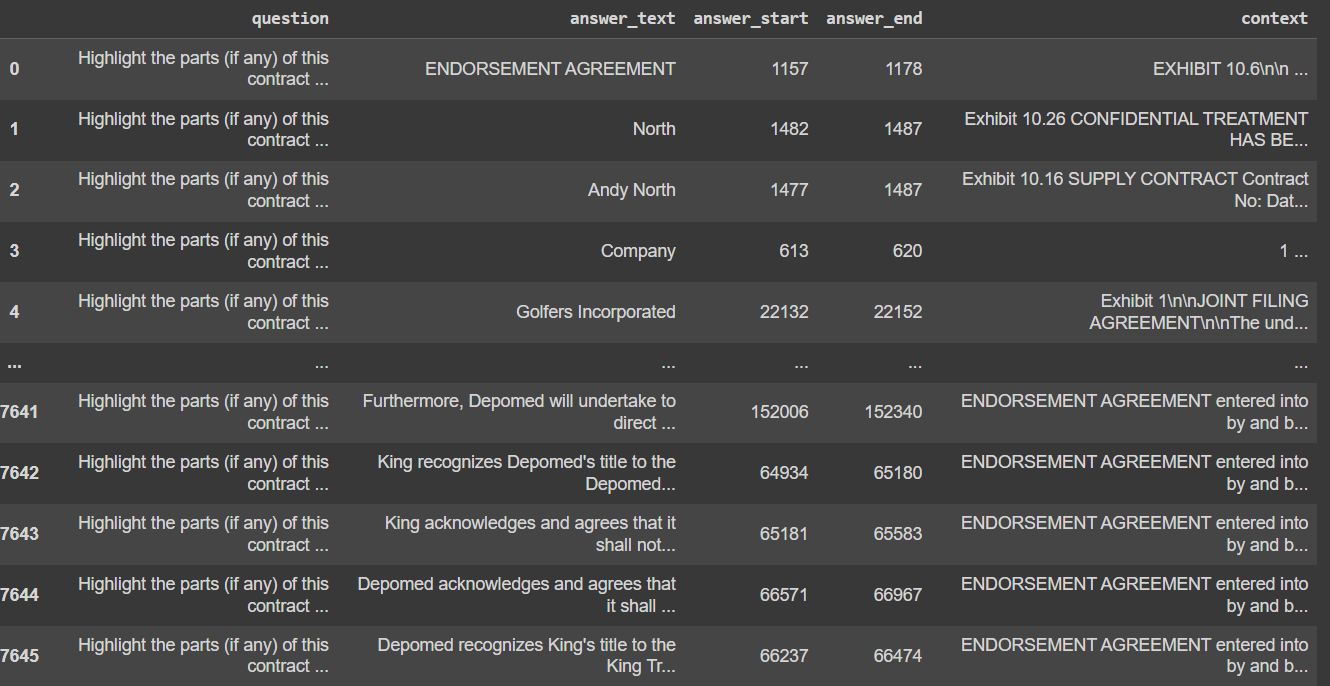
# Introduction

# The goal of question answering (QA) is to automatically provide answers to user inquiries based on information found in external sources including the Web, knowledge bases, and free text. Reading comprehension is a crucial form of quality assurance that aims to provide a response to a query after the passage is read. Building deep neural models to perform RC problems has been made easier by the recent availability of large-scale RC datasets, such as the Stanford Question Answering Dataset (SQuAD) and CNN & Daily Mail. Contextualised word representations and pretrained language models—like ELMo, GPT, and BERT—have proven to be especially helpful in a variety of natural language processing tasks, including RC, in more recent times. These pretrained language models can capture the complex semantic content and generate more exact and accurate representations for words given by seeing a variety of settings across big datasets.

# Method

This project's aim is to develop a question answering system in the legal domain. For the implementation of a T5-based model which returns “an answer,” given a user question and a passage which includes the answer to the question. For this question answering task, we used CUAD dataset tuning a model that is trained on SQUAD 2.0. Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable.Contract Understanding Atticus Dataset (CUAD) v1 is a corpus of 13,000+ labels in 510 commercial legal contracts that have been manually labeled under the supervision of experienced lawyers to identify 41 types of legal clauses that are considered important in contact review in connection with a corporate transaction, including mergers & acquisitions, etc.. I started with the t5-base pretrained model “t5- base-uncased” and fine-tuned it to have a question answering task. For Question Answering we use the t5 for Question Answering class from the transformer's library. The T5 model is a transformer-based neural network architecture. Transformer models are known for their parallelizable and scalable architecture, making them suitable for a wide range of NLP tasks..T5 is pre-trained on a large corpus of text data using a denoising autoencoder objective. It learns to generate the target text from a corrupted or noisy version of the same text.The model is then fine-tuned on specific downstream tasks by providing task-specific input-output pairs.. The Transformers library has a diverse collection of pre-trained models which we can reference by name and load easily. The vocabulary of this model is identical to the one in t5-base-uncased. The T5 tokenizer is responsible for converting raw text into a format suitable for input to the T5 model. It breaks down the input text into smaller units called tokens.

Sample of the dataset:



Our Implementation

* First Step:

Evaluating performance of the CUAD finetuned t5QAmodelonlegalQA.

* Second Step:

T5 finetuned t5-QA model onlega lQA.

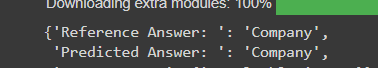
* Evaluating model performance on Fine-tuned model with legal QA dataset

# Training

The trained model is stored as a Pytorch checkpoint file, which may be used for testing or further training with more datasets for improved performance. We examined the dataset again using word piece tokenization in dataset t5 after training. Rare words in T5 are divided into smaller words or phrases. To clean up and tokenize the sentences and produce word embeddings, we employ a pretrained tokenizer. Using a pre-trained distillbert model, we trained our model for ten iterations. We then assessed and measured its performance using test and validation datasets.



Output:



# Evaluation

We adopt our model two metrics including Exact Match (EM) and F1 scores to evaluate our model. The EM score determines the percentage of predictions that perfectly match the ground truth answer,and the F1score demonstrates the average overlap between the prediction and the ground truth answer.

|  |  |
| --- | --- |
| Measure | TestScoresonlegalQA |
| ExactMatchScore | 61.20 |
| F1 | 67.39 |

# Conclusion

On domain-specific datasets, models trained on the general domain dataset perform poorly. One of the most crucial phases in adapting to the legal domain is task-driven fine-tuning with legal domain-specific QA dataset. Only clinical notes or paragraphs from the QA dataset are available for language model training when adapting to the legal domain; this results in a negligible performance gain. It can be beneficial to fine-tune the t5-QA model using a large legal domain QA dataset before fine-tuning on a domain-specific QA dataset if the domain-specific dataset is small.

# References

* <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2760988.pdf>
* <https://towardsdatascience.com/the-definitive-guide-to-bi-directional-attention-flowd0e96e9e666b>
* <https://web.stanford.edu/class/cs224n/materials/CS224N_PyTorch_Tutorial.html>
* <https://arxiv.org/pdf/2005.00574.pdf>
* <https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/>
* <https://ieeexplore.ieee.org/document/9762943>
* https://www.atticusprojectai.org/cuad

# GitHubRepository:

* https://github.com/shasha612/nlp