Question Answering over Legal Documents

Ahmed Ahmed
ahmedahm@seas.upenn.edu

Danqi Luo
danqiluo@seas.upenn.edu

Hannah Gonzalez
hannahgl@seas.upenn.edu

Jiani Huang
jianih@seas.upenn.edu

Karan Jaisingh
karanj@seas.upenn.edu

Abstract

Machine reading comprehension is a useful tool to extract information from massive online resources in natural language form. However, with the domain shift from generic natural language to legal documents, the reasoning complexity increases steeply. We studied the performance of the generic language model in the legal document domain and propose extensions such as date calculation and period extraction that are specifically important for question answering in this field. Our extensions improves the completeness and correctness of the date and period related questions as well as the predictive capabilities on boolean questions.

1 Introduction

Existing approaches such as Deep Learning have been useful in advancing reading comprehension, however, it is mostly limited to extractive questioning answering (text pattern matching). Additional limitations to current question answering systems include considering syntactic and semantic roles, selecting exact answers, better answer ranking, numerical augmentations, and providing an answer justification. As opposed to strictly pattern matching, this task involves integrating distributed representations with symbolic operations to enhance multi-step reasoning in a specialized domain application of MRC. The use of transformers such as BERT (Jacob Devlin, 2018) and DeBERTa (Pengcheng He, 2020) offer a method to begin to apply MRC to specific domains, however, just the application of these and similar models are not sufficient to achieve a good performance.

In order to discover more about numeric reasoning and try to improve some of the existing MRC baselines, we decide to work with the Contract Understanding Atticus Dataset(CUAD) (Hendrycks et al., 2021) which consists of 510 legal documents, and 41 questions. Our main task is to train a model to answer questions with regard to legal contracts. A sample data point is shown in Table 1.

Legal Contract
THIS CO-BRANDING AGREEMENT (the ”Agreement”) is made as of May 22, 2000 (the ”Effective Date”), by and between WOMEN.COM NETWORKS, INC., . . . , shall remain effective for two (2) years from and after the Effective Date (the ”Initial Term”). . . . ”
Question and Answer Pair
Question: Expiration Date Answer: 5/22/02 Explanation: [’shall remain effective for two (2) years from and after the Effective Date’]

Table 1: Sample data point of CUAD dataset

From the example we show, we can see that in order to answer the question, we need to perform date extraction(’May 22, 2000’), period extraction(’two(2) years’) and date addition(05/22/2000 + 2yrs) to obtain the final answer.

We chose this task and the CUAD dataset for several reasons. First of all, the CUAD dataset provides questions in various categories including numeric reasoning, date extraction, span extraction and boolean questions which makes it challenging enough and suit our purpose well. In addition, the current baselines built for CUAD dataset only perform span extractions which are unable to directly answer the questions and the performance is far from idea. This means there is enough room for us to work on and improve the baselines.

2 Literature Review

DROP dataset. To achieve our goals for numeric reasoning in CUAD, we want to find some related work done in similar fields. Although the

CUAD baselines cannot perform mathematical calculations, we can draw inspiration from the DROP dataset (Dua et al., 2019), which stands for Discrete Reasoning Over Paragraphs. The DROP dataset contains approximately 96k questions which require the MRC system to perform some kind of Discrete Reasoning Over Paragraphs to obtain the correct answers. There are various types of questions that can be asked in the DROP dataset which includes different arithmetic operations.

Compared to the current DROP baselines and question-answer pairs, the CUAD dataset has a stronger focus on Yes/No questions (33 out of 41 categories). However, the remaining 8 categories share very similar characteristics with the DROP datasets, which require answers in the form of exact numbers, dates, spans, and names, etc. To achieve better accuracy on questions that require numeric reasoning, we would like to draw inspiration from the NeRd and QDGAT models which are tested on the DROP datasets to see if we can apply them to the CUAD dataset. Neural Symbolic Reader: Scalable Integration of Distributed and Symbolic Representations for Reading Comprehension

NeRd. The Neural Symbolic Reader (Xinyun Chen, 2020) approaches reading comprehension through the use of symbolic representations, as existing approaches which utilize neural modules are found to be difficult to generalize to other domains. It proposes an architecture that encompasses both a reader which encodes a passage and question and a programmer that generates the program which is utilized for the reasoning process. The decoder's output is a program in their domain-specific language that can be executed directly. The DSL introduces four span selection operators (PASSAGE_SPAN, QUESTION_SPAN, VALUE, KEY-VALUE) on top of the basic arithmetic, counting, and sorting functions to allow for the tokens to be extended to the text samples as well. The paper concludes that the NeRd architecture provides better scalability relative to existing models as the identical neural architecture can be generalized across domains and the programs can be generated via a compositional application of symbols and operators.

QDGAT. (**?**) The experiments within this paper are focused on DROP. Compared to the existing baseline approaches including NAQANet (F1 of 47.77), MTMSN (F1 of 79.85), and BERT-Calc (F1 of 80.53), NeRd garners a testing F1 score of 81.71. In particular, when considering questions that require multiple-choice selections, we find that NeRd outperforms the most optimal baseline models by over double in terms of accuracy, with an increase in F1 score of 15Question Directed Graph Attention Network for Numerical Reasoning over Text This paper attempts numeric reasoning in machine reading comprehension(MRC) through the use of a heterogeneous typed graph network model called question directed graph attention network(QDGAT) which encodes relationships between entities and numeric values as well as relationships among numeric values themselves (Kunlong Chen, 2018) . The model mainly includes four parts: Word Representation Extractor, Graph Construction Module, Numerical Reasoning Module, and Prediction Module. The Extractor transforms input passages and questions into their representations; the Numeric Reasoning module reasons over the representation of inputs and the graph from previous modules conditioned by the embedded question command; the Prediction Module will then employ the output from the QDGAT network for final prediction in the form of Span Extraction, Count or Arithmetic Expression.

3 Experimental Design

3.1 Data

This paper deals with the curation and analysis of the CUAD dataset (Hendrycks et al., 2021) which is a contract review dataset annotated with 41 label categories that lawyers pay particular attention to when reviewing a contract. This dataset considers clauses that would warrant lawyer review and tries to create a question answering system for this specific domain. Refer to Table 1 to look at a sample data point of the CUAD dataset.

The researchers used QuestionAnswering models in the HuggingFace Transformers library (Wolf et al., 2020) as a baseline. It performs well on simple questions such as Governing Law and Document name but performs badly on more complex questions such as a covenant not to sue. The experiment shows that DeBERTa has the best performance among BERT, ALBERT, RoBERTa, and DeBERTa. Further, model size had little impact on performance, suggesting that data is a major bottleneck. There is still a great deal of room for improvement for machine-reading comprehension in a highly specific domain.

Extension 1 The data used for the first extension is a subset taken from the CUAD data set. It only considers the 33 categories that are of boolean questions. An example of the data set can be seen on Table 2, where 0 is no and 1 is yes.

Text	Output
Most Favored Nation	0
Non-Compete	0
Change of Control	1
Competitive Restriction Exception	0

Table 2: Example of data used on first extension

The distribution over the labels is shown in Table 3.

Label	# of instances with such label
0	12,883
1	3,617

Table 3: Distribution over the labels

Extension 2. The second extension converts the data calculation and period calculation problem into a classification problem. Thus we need to preprocess the dataset to obtain classification labels.

Following the QDGAT work, we first pass the document through the NER system in Stanford Core NLP package.(Manning et al., 2014) We therefore obtain a set of phrases with their named entity labels, such as DATE, DURATION, TIME, ORG. For date calculation, we convert all of the phrases with labels DATE, TIME into structured dates, and all of the phrases with labels DURATION, TIME into structured periods. Then we enumerate all possible combination of valid date and periods, and perform $+$ and $-$ over these combinations to obtain calculated dates using the date package in python. We validify whether the ground truth lies in the union of the structured dates and the calculated dates, if the ground truth can be calculated, we add it into the processed dataset, otherwise we discard the data point. Similarly, we check whether the structured periods contains the ground truth answer, if not, we also discard the datapoint. Due to the inaccuracy in the NER step, we only have 169 valid datapoints left. To obtain an statistically sensible result, we split the dataset into training (80%) and evaluation (20%). We report the evaluation result rather than the test result.

3.2 Evaluation metric

The selected evaluation metric for the given problem is a Macro-Averaged F1 Score (Rajpurkar et al., 2016). For each question in a given document, an F-1 based scoring metric is computed based on the ground truth answer and the predicted answer. For questions involving dates, we first parse the date string to obtain numeric representations of day, month and year, which are then compared. For questions involving time periods, we parse the time period to obtain numeric representations of time and unit, which are then compared. This metric is then averaged over all questions for the given document to obtain a per-document F-1 score, which is then again averaged over all documents to obtain an overall F-1 score for the model itself. The Macro-Averaged F1 Score can then be calculated as follows:

1. For a given document, iterate through each question.

2. For each question, obtain both the predicted answer (refer to this as ’pred’) and the ground truth value (refer to this as ’true’).

3. For questions involving dates, we use a custom date parser to numeric representations of day, month and year for the answers. We indicate a score of 1 if there is a match in dates, and 0 if not. We do the same for questions involving time periods, obtaining representations for time and unit for the answers. However, we relax the period comparison to include a range of values, wherein at the upper end a 1 indicates an exact period match and at the lower end a 0 indicates a period mismatch of anything greater than or equal to the period of time itself.

4. For other questions, compute the string-based F-1 score between these two answers as follows by finding the Precision (numCommon / number of tokens in pred) and Recall (numCommon / number of tokens in true), and then via: F-1 = (2 * Precision * Recall) / (Precision + Recall).

5. Obtain the average of the scores for each question within the document, and then calculate the Macro-Averaged F-1 Score by obtaining the average F-1 score over all documents.

3.3 Simple baseline

Our weak baseline utilized the question answering transformer model in hugging face, then we post-process the output into a structural format, such as date and time. The first part is the baseline described in the original CUAD dataset paper, while the second part is for measuring the real answer accuracy rather than just measuring the explanation extraction accuracy. The pipeline is shown below. There are two components we add into the pipeline, the date parser, and the period parser. Depending on the question, we choose which parser to use. For example, the "effective date" expects an answer in date type, while the "term period" expects an answer in period type. Both the date parser and the period parser rely on the off-the-shelf date parsing library and regex extraction to obtain the month, day, year information. Further, the period parser also relies on a unit conversion library, so that it can unify different units into the same measurement, such as 1 month to 30.44 days. The transformer output is far from ideal. Especially in the date-related questions, hardly can this pipeline capture the numerical related information. Here are a few sample answers:

4 Experimental Results

4.1 Published Baseline

As described in previous literature, Question directed graph attention network (QDGAT) is a heterogeneous typed graph network model which encodes relationships between entities and numeric values as well as relationships among numeric values themselves. Our strong baseline utilized the model from QDGAT and we applied the model on the eight questions that require numeric reasoning and span extraction from CUAD. In order to do this, since the QDGAT model requires data in DROP format and annotation before processing, we add a data preprocessing and annotation step to the pipeline as well as a prediction generation process.

According to QDGAT, we are only getting around 0.2 F1 score. However, from the prediction results, we can see that the model performs exceptionally well on 'Document Name' extraction and some Date predictions. Although it cannot give the correct answer to many of the questions, it gives the correct types most of the time, i.e. numbers for date and spans for other questions. Table 4 contains some examples of the predictions.

Some reasons that are causing the low performance could be the length of passage and question and answer format. For example, the model cannot extract enough context information from the questions being asked since they are short and do not include any entities. In addition, for questions like Renewal Term which expects answers like 'successive 1 year', it is difficult for the model to extract such information from the span.

4.2 Extensions

Extension 1 The first extension of QDGAT was motivated by the desire to better answer the boolean questions in the dataset, which in comparison to the date related questions, should have a different approach. Firstly, QDGAT outputs a span for these types of questions, rather than a yes/no answer. Therefore, our extension to this model was to use the output of the span QDGAT predicts for the boolean questions and then use a bidirectional LSTM to predict the final output. Our assumption was that the output of QDGAT should theoretically encode all important information associated with a question and would be sufficient context to let the LSTM accurately predict the final yes/no result. In the first extension, we split the dataset into training (80%), evaluation (10%), and testing (10%) sets. We then cleaned the data set by removing conjunctions and replacing them with their full name, as well as tokenizing the text. After that, we train the bidirectional LSTM using an adamgrad optimizer, binary cross-entropy loss, 30% dropout after each layer, and a final sigmoid activation function at the end to get the output.

The output of extension 1 is much better than the original without the baseline. The original QDGAT model had a .61 f1 score on the boolean questions whereas this extension had an f1 score of .83, which is a very sizable increase in the predictive capabilities.

Extension 2

We follow the notation of QDGAT, and implement a date calculation module and a period extraction module. Given that P is the passage, and Q is the question, by passing them through the RoBERTa network yields us with $\hat{\mathbf{Q}}$ and $\hat{\mathbf{P}}$. Further, the QDGAT network constructs a graph using heterogeneous type information. The graph $G = (V, E)$ contains numbers N and entities T as the nodes $V = N, T$, and its edges E encode the information of the number type and the re-

Document Name	Agreement Date	Effective Date	Expiration Date	Renewal Term
WEB HOSTING AGREEMENT	9/9/1997	9/9/1997	9/9/1997	correct any technical problems (Server being down or inaccessible) 24 hours per day , 7 days per week . . . .
BROKER DEALER MARKETING AND SERVICING AGREEMENT	2013	2013	2013	2013
GLOBAL MAINTENANCE AGREEMENT	3/3/2017	3/3/2017	3/3/2017	BRASILEIRAS S / A (as Company) and AVIONS DE TRANSPORT REGIONAL , G.I.E . (as Repairer) 2015 03 09 AZUL - ATR Global Maintenance Master Agreement DS / CS - 3957 / 14 / Issue 7 Page 1/110 Source : AZUL SA , F - 1 / A , 3/3/2017

Table 4: Examples from QDGAT Baseline

Precision	Recall	F1 score
0.78	0.88	0.83

Table 5: Performance on test set of first extension

lationship between the numbers and the entities. Then we have the embedded graph representation $U = QDGAT(G; \hat{\mathbf{Q}}, \hat{\mathbf{P}})$. Then we train two classifiers with the input $I = \mathbf{W_i}(U : \hat{\mathbf{Q}} : \hat{\mathbf{P}})$.

In the CUAD dataset, only addition and subtraction operations are involved. Our date calculation is achieved by classifying each date into one of (0, +1), and each period into one of (-1, 0, +1) and our period retrive module is achieved by classifying each period into one of (0, +1), which is then used as the coefficient of the number in the date calculation expression to arrive at the final answer.

We thus, are able to incorporate structured calculations in the learning pipeline. With these two new modules, we have improved the original QDGAT baseline on the date and period related questions by 0.1 F1 score.

	w/o extension2	extension2
F1	0.12	0.22

Table 6: Performance on test set of second extension

4.3 Error Analysis

Extension 1,

As mentioned before, the output of extension 1 is much better than the original without the baseline; the original QDGAT model had a .61 f1 score on the boolean questions whereas this extension had an f1 score of .83.

Our original assumption was that the output spans of QDGAT should encode all contextual information required in order to correctly answer the question. However, we can clearly see that is not the case since we are only getting an f1 score of .83. This model tends to favor choosing no as an answer which can be seen through the relatively lower precision than recall. The number of false negatives are higher than the number of false positives meaning that the model is relatively unsure when predicting the positive class. This can be due to a wide number of factors, however we hypothesize that it is largely due to the class imbalance in there being about 4 times more negative instances than positive instances. However, as a whole, this model is far better than the original baseline. Future improvements on this extension could leverage some type of attention for increased context impact as well as a better way to encode the original context from the QDGAT model.

Extension 2. The output of extension 2 is significantly more readable and complete than without

w/o extension2	extension2	ground truth
6th	4/6/1999	4/6/1999
13	None	9/1/2004
10th	1/10/2018	1/18/2010

Table 7: Example outputs for extension 2.

using the extension. As represented in Table 7, the answer without extension 2 cannot output a valid date, but usually a part of the date.

The errors that extension 2 makes mainly falls into two categories:

1. The classified output cannot formulate a valid equation. For example, `-2 years` is not a valid answer for date calculation. This issue will lead to `None`s in the output.

2. The date has swapped digit in recognized month, year, and dates. For example, the predicted answer is "1/10/2018" while the ground truth is "1/18/2010". This is because QDGAT network has preprocess the text, and substitute all the natural language representation to a numerical representation. For example, "January 18, 2010" will be processed into "1 18 2010". Sometimes, this step will create ambiguity for the date parsing library, and the final output may have misplace date, month, and year.

5 Conclusions

Overall, this project has been a success. We were able to improve on the baseline QDGAT model with both of our extensions, increasing the f1 score of the 33 out of 41 boolean questions from .62 to .83 and increasing the f1 score on the remaining 8 questions from .12 to .22. Given the complicated nature of this hightly specialized domain of machine reading comprehension and question answering, the models are still far from good. Due to the relatively recent creation of this dataset, there is no real State of the Art, only current applications of traditional models like BERT and DaBerta, when perform about as well as our models. If we had access to more computational resources, we could alter and use transfer learning to train Bert and DaBerta using the same extensions as this project which we now know would result in performance improvements. Finally, there is still a lot of work to be done on date relateed questions given that .22 f1 is not very good in general. This is a much harder challenge since the understanding of language that requires semi-complex and complex logic is still a progressing topic in machine learning. However, we are very satisfied with the findings of our project.

6 Acknowledgements

This paper is an extension of Jiani Huang's research at the University of Pennsylvania. We would also like to thank our TA for this project, Qing Lyu, for her continued help and assistance with the entirety of this semester long project.

References

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. CUAD: an expert-annotated NLP dataset for legal contract review. *CoRR*, abs/2103.06268.

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

Xingyi Cheng Zou Xiaochuan Yuyu Zhang Le Song Taifeng Wang Yuan Qi Wei Chu Kunlong Chen, Weidi Xu. 2018. Question directed graph attention network for numerical reasoning over text. *EMNLP*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Jianfeng Gao Weizhu Chen Pengcheng He, Xiaodong Liu. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *CoRR*.

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

Adams Wei Yu Denny Zhou Dawn Song Quoc V. Le Xinyun Chen, Chen Liang. 2020. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. *ICLR*.