M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Exploring CUAD using RoBERTa span-selection QA models for legal contract review

Gustav Selfort Hartz (s174315)

Supervisor: Søren Hauberg
Co-supervisor: Nicki Skafte Delftsen

b

Abstract

Deep learning models for span-selection question and answering have, with the advent of transformer models such as BERT and RoBERTa, shown excellent performance on established tasks like SQuAD and GLUE. Training these models requires large annotated datasets, making them expensive to use in highly specialized domains.

The Contract Understanding Atticus Dataset (CUAD) is a new public dataset by The Atticus Project, Inc., suitable for training span-selection models for legal contract review. This thesis presents an in-depth analysis of the performance obtained from the RoBERTa models presented by the authors of CUAD. It examines the models' robustness to multiple different new formulations of the original 41 questions, in addition to methods for reducing the number of model parameters by upwards of 20% and model inference time by close to 50%, with little to no effect on performance. Moreover, ways of boosting model performance based on a thorough analysis of the CUAD dataset in terms of annotation quality, question category annotation distribution, and potentially unmarked answers were tested. Additionally, a new proposal for pre-processing CUAD is presented to mitigate ambiguity in training data. The thesis also argues for using other evaluation metrics than CUAD. These new metrics show that performance across all 41 question categories is better than indicated in the original paper and likely usable for a human-in-the-loop legal contract review system.

The thesis improves the `top 3 Has Ans metric` from 92.5% to 94.1%, but cannot conclude that this is due to augmentations made to the dataset. Additionally, it suggests that performance gains can be found by adjusting objective functions to suit the performance metrics better.

Decleration

This thesis was prepared at the Technical University of Denmark, Department of Applied Mathematics and Computer Science in fulfillment of the requirements for acquiring a Masters of Science degree in Human-Centered Artificial Intelligence. The project was prepared by Gustav Selfort Hartz. The workload of the project is 30 ECTS.

The reader of this thesis is expected to have basic knowledge of deep learning, natural language processing, and recurrent neural networks.

Kongens Lyngby, 2nd August 2022

Gustav Selfort Hartz

Contents

CHAPTER 1

Introduction

In recent years transformer models have made significant improvements in Reading Comprehension (RC) and Question Answering (QA) tasks that extract information from real-world documents, even surpassing human performance on challenges like SQuAD[51, 53] and GLUE[71]. To achieve this state-of-the-art performance large quality annotated datasets are needed, making it expensive and challenging to utilize the models for specialized domains. One industry that could benefit dramatically from these models is legal. Law firms use upwards of 50% of their time on contract review [23]. This contract review process can include assessing risky clauses, liability, and financial risks. Additionally, a major time consumer in legal processes are extracting preamble information such as *Parties*, *Document Type*, *Agreement Date*, and *Effective Date*. These pieces of information are often used for organizing the documents internally in folder structures with hundreds of folders, as explained by an assistant project manager at a major danish law firm in an interview for this thesis. This technology can potentially reduce the cost of legal services as a direct consequence of the increased efficiencies rendering legal services more accessible to companies and people that might have had limited access before.

The amount of research published within the area of legal has steadily increased since 2011, as seen in Figure 1.1, with an increasing number of legal contract understanding datasets being published, such as ContractNLI [31], and LEDGAR [66]. According to Google Scholar, these datasets have seven citations each at the time of writing, with little analytical derivative work. This field of research within natural language processing is still young, and there are no well-established performance measures that quantify the performance of legal understanding models, though they are emerging[11]. This thesis will seek to contribute to the discussion and share insights into the Contract Understanding Atticus Dataset, that hopefully will benefit future research within the field, both for QA, and other areas of NLP within the legal domain, that benefit from deeper data insights than what is currently available. Additionally, it will look at improving the state-of-the-art models based on the Transformer Architecture[68] for span-selection Question and Answering (QA) in a legal context.

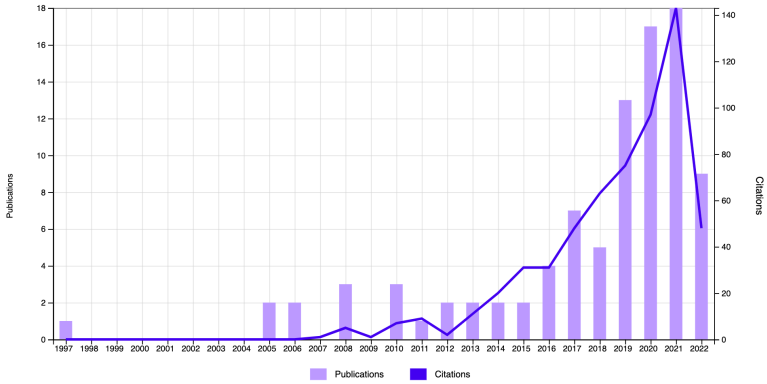


Figure 1.1. Publications and citations of material related to Natural Language Processing and Legal[1]

1.1 Related work

Recurrent neural networks and advanced information retrieval systems has show decent performance[3, 12] on Reading Comprehension (RC) challenges (see Section 2.1) using static word embeddings such as glove [48] and word2vec [42]. Since the publication of the large labeled dataset such as the Stanford Question Answering Dataset[51] and the CNN/Daily Mail dataset[24], the transformer models have dominated the leaderboards of the standard RC challenges[53].

The area of span-selection QA within a legal context and for legal element extraction can be classified as a closed or specialized domain RC challenge. Chalkidis et al. [9] investigated contract element extraction for elements such as title, effective period, governing law, and more using manually written rules, linear classifiers such as logistic regression and support vector machines, along with custom word embeddings and parts of speech tagging. The paper evaluates performance using precision, recall, and f1 and shows promising results from their `SW-SVM-ALL` model. The paper's approach is to classify contract elements on a token level, determining whether or not it is part of the answer. Every token is evaluated using a sliding window of 11-13 tokens. The extractions zones, where to search for potential questions answers, are identified using a rule-based approach, but it is unclear how long these extractions zones are in terms of tokens.

The authors later published another paper where they explore bidirectional long short-term memory (BILSTM) models on a similar dataset[8], showing that deep learning models "operating on word, POS tag, and token embeddings outperform

[1] Web of Science - Query: Legal OR LAW OR Contract (All Fields) and Natural Language Processing (Topic) and Computer Science Artificial Intelligence (Web of Science Categories)

the linear sliding-window classifiers of (their) previous work, without any manually written rules"[8]. In this paper, the authors also evaluate the models' ability to predict the entire contract element as a sequence, instead of on a token level, as it drastically decreases the complexity of adding new tasks due to the extensive requirements of pre- and post-processing rules.

Both papers use the same dataset of 3500 contracts[8, 9] of which 1000 are labelled with 11 types of contract elements. Due to privacy concerns, these tokens are provided in an encoded format to the public.

Within the industry, Atrium, a law firm based in San Francisco, has been developing machine learning models to "save time for lawyers and paralegals by automating some of the boring stuff they have to do every single day"[76]. The company has used similar approaches with deep learning models and sliding window techniques and taken efficiency within intake, a category of tasks in legal companies, from 30 to 150 documents an hour[76].

In 2021 the Contract Understanding Atticus Dataset (CUAD) v1 was published (see Section 3.2), which is the largest public annotated contract dataset to date[23]. It includes over 13.000 annotations across 500+ contracts and with 41 different types of clauses. The paper experiments with transformer models (see Section 2.2) using the Huggingface `Transformers` Library [75]. The task structure in this paper is defined in the same format as SQuAD [51], where the models are tasked with predicting start and end token positions using a sliding window approach with a configurable step size (see Section 3.1). The paper assesses model performance using AUPR due to the large class imbalance since most contract parts do not have an annotation. The primary findings are that the amount of data greatly impacts performance by decreasing the dataset size by a factor of 10, AUPR is roughly reduced by ten percentage points from 45-35%. Furthermore, the paper finds a strong correlation between model performance and model size, with large versions of models outperforming the base version across the line, similar to the findings of Chalkidis et al. [11].

Hegel et al. [22] showed that performance on the CUAD dataset could be increased by incorporating visual queues into model inputs. Chalkidis et al. [11] introduced the Legal General Language Understanding Evaluation (LexGLUE) benchmark to standardize an answer on whether current state-of-the-art models can generalize across various tasks in the legal domain.

Vaswani et al. [68] initially presented the transformer, but the architecture has since then been developed for many different use-cases such as the encoder-only models BERT[14] and RoBERTa[37] and decoder-only models like GPT2[49] and GPT3[6]. Central for the performance of transformer models are the attention mechanism, which has been the subject of much research[13, 26, 67, 69, 74]. Sajjad et al. [58] showed that models like BERT, RoBERTa and XLNet can be reduced in size "up to 40%, while maintaining up to 98% of their original performance".

These works forms form the foundation of the thesis, which has the objectives set forth in Section 1.2

1.2 Research question and objective

This thesis aims to recreate, analyze, and enhance the closed-domain extractive Question and Answering model proposed by Hendrycks et al. [23] based on the Contract Understanding Atticus Dataset (CUAD) with the perspective of using it in a human-in-the-loop legal contract review system. This is done by working with the following sub-problems:

- Setup a modern single-node multi-GPU training environment that allows for efficient experiments with both model architectures and data manipulations
- Thoroughly examine the CUAD dataset to interpret and understand model behavior and seek performance increases through data quality improvements
- Analyse and interpret attention weights from the model outputs on legal domain task to enlighten model performance
- Assess model performance metrics from CUAD and litterateur to establish reasonable evaluation metrics for the task.
- Explore the inclusion of other data sources to improve model performance
- Review methods and possibilities for boosting the usability of the RoBERTa-based span-selection QA system in a production setting

Additionally, it is a clear goal of this thesis to provide easy access to the training framework presented and the trained models.

1.3 Overview

This section gives an overview of what can be expected from the different thesis sections. The associated source code can be found on the thesis authors' Github page, with a link in Appendix, Section A.1. Appendix, Section A.1 also contains links to the Weights and Bias projects with in-depth data about model performance and results, along with instructions for obtaining the best performing model(s) presented in this thesis.

The following is an overview of the chapters in the thesis:

Chapter 2: A theoretical motivation for Question and Answering within natural language processing, along with fundamental theory about the transformer model architecture and its training.

Chapter 3: An overview of the datasets used in this thesis, along with the machine learning operation tools used to facilitate the training and monitoring of the models described.

Chapter 4: In-depth descriptions of the initial modeling based on the original proposal by CUAD implemented in a new framework along an evaluation setup

Chapter 5: Deep dive into the analysis of model performance based on the architecture and evaluation framework presented in **Chapter 4**

Chapter 6: Attempts to enhance the values of the models presented in **Chapter 5** from a production perspective, including dataset manipulations, transfer learning from other datasets, and experiments with dataset size reductions.

Chapter 7: The conclusion of thesis

CHAPTER 2

Background

This chapter presents background knowledge covering question answering (QA) and the transformer architecture. It provides an overview of the mechanisms within transformers and QA needed to understand the thesis but does not include all details of their implementation and the theory behind them. For this, the reader is referred to the papers written about the subjects in the bibliography.

2.1 Question Answering

Question Answering (QA) is a task from computer science within NLP and information retrieval (IR)[12]. Information retrieval covers processes such as identifying documents and paragraphs with high similarity to a search query. Due to the scope outlined in Section 1.2 IR will not be covered in this thesis, as it falls outside the scope, even though it is a central part of most production QA systems.

There are multiple QA task classifications based on the models' goals, inputs, and outputs, but most can be covered by the two major categories: open-domain QA and closed-domain QA.

Definitions of the two terms vary across papers[4, 15, 43, 73], but in general open-domain systems refer to systems where the input (question) does not include any context, and the systems have a large corpus of documents that it processes to identify relevant documents and paragraphs as shown by Chen et al. [12] on Wikipedia data. This often entails document retrievers, a concept from IR, but it has been shown that state-of-the-art performance also can be obtained from large transformer models that "memorize" facts during pre-training[6, 54, 73]. Systems with access to a non-fixed database that relies on document retrievers or other techniques are termed open-book, whereas a system with the entire corpus of knowledge "memorized" is considered a closed-book system. The system presented by Brown et al. [6] is an example of a closed-book system, whereas Chen et al. [12] is an open-book system.

Closed domain systems are designed only to answer questions about a specific domain. This could be mathematics, physics, chemistry, or something completely different and it might very well entail only operating with a fixed set of questions in that domain [4, 15]

Reading comprehension, also referred to as extractive QA or span selection QA, is a category of QA tasks where the answer is found in a provided context. Extractive

QA is, by some researchers, seen as a subset of the open-domain system[43], though the terminology used in this thesis falls into both categories. Open-book closed-domain RC is the focus of this thesis, and models presented in Chapter 5 and 6 would constitute the process after IR in a more extensive system that extracts the answers after the relevant subset of the corpus has been identified.

2.2 Attention is all you need

Natural language processing has received much attention in recent years due to the advance of the transformer model. Models based on the transformer architecture are at the time of writing among the top performers[53] on the most well know NLP tasks such as SQuAD v1[51] and SQuAD v2[52]. Furthermore, it has shown massive improvements in the task of machine translation, generally outperforming Recurrent Neural Networks (RNN) [68], and transformer-based models are currently powering Google Translate[21].

The transformer architecture was originally presented by Vaswani et al. [68]. This paper presents an alternative to complex recurrent and convolutional neural networks (CNN) by reducing sequential computation as seen done with Extended Neural GPU[27], ConvS2S[18], and ByteNet[28]. The proposed transformer architecture is solely based on attention, a concept explained in Section 2.2.2, and thus more parallelizable than the models mentioned earlier. This is due to the way it processes data. Unlike the RNN, which needs to process input data sequentially, meaning that the computation of hidden states h_i of an input x_i, is dependent on h_{i-1}, the transformer architecture is non-sequential and thus processes the entire input sequence at once instead of word by word. This is also why transformer models are better at long-range context dependencies, as they can evaluate the entire context simultaneously, whereas RNNs and Long short-term memory (LSTM) models depend on previously computed internal hidden states. To mitigate some of the problems with long-range context dependencies, RNNs and LSTMs have been extended to be bi-directional, meaning that the input is encoded from both start and end direction, allowing for words at the end of the sentence to have an effect on the hidden states in the beginning, but the transformer-based architecture still outperforms these alterations[53]. Compared to CNNs for sentence classification, Transformers also has advantages due to the many different kernel sizes needed to handle a wide variety of text dependencies, based on the notion of a 2x2 kernel modeling word duplets, 3x3 kernel handling triplets, etc.[29, 68]

2.2.1 The architecture

As presented in its original form, the transformer architecture uses an encoder-decoder structure, where the encoder consumes a sequence of symbols; in the case of text input, this would be tokenized words or embedding. In general, the input can be described

as $[x_1, x_2, ..., x_n], n < D_{inputlimit}$, where n is the length of the input and $D_{inputlimit}$ a parameter of the model that defines the maximum inputs span. This parameter is often fixed at 512[14, 37, 68] as it provides a good balance between model complexity and input size because the complexity of attention scales quadratically in the input size. The encoder then produces a continuous output $[z_1, z_2, ..., z_n]$. Given $\mathbf{z}$ the decoder then produces an output sequence $[y_1, y_2, ..., y_n]$ of symbols one by one in an auto-regressive manner also consuming the previously generated symbols alongside the output of the encoder for each step [56, 68].

Though it varies across implementations, the fundamental logic of the encoder-decoder structure is that the encoder, in the text case, constitutes a language model converting the tokenized input sequence and transforming it into a language representation in latent space. The decoder part of the model then takes this representation and generates an output sequence. Vaswani et al. [68] showed this for a translation, as the model is trained on the WMT 2014 English-German dataset[5]. In general, some auto-regressive structure is needed in the model for the translation task else; the entire output should be predicted at once, somewhat equivalent to translating a sentence by translating each word individually without consideration to the rest of the translation, which does not yield a good translation for apparent reasons. Transformer models can also exist as encoder-only and decoder-only forms, which will be explained in Section 2.2.4

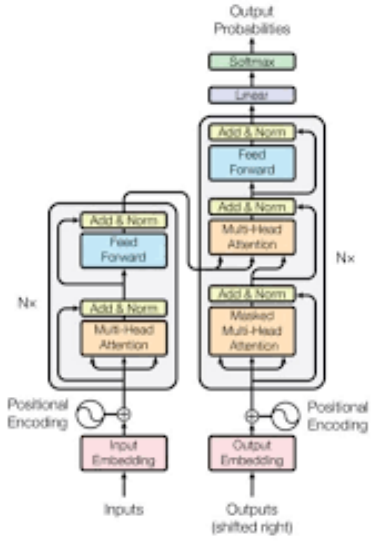


Figure 2.1. Visualizations from Vaswani et al. [68] of the transformer architecture

The encoder (left side of the model in Figure 2.1) consists of a stack of $N = 6$ identical layers. Each layer receives the same input of symbols with a positional encoding applied. This is then passed to the first of the two sub-layers, the multi-

headed self-attention. Dropout is applied to the sub-layer's output and then combined with the residual connection and normalized. The normalized data is then passed through a fully connected layer with a residual connection where the input is added and normalized. As with the first sub-layer dropout is also applied here. Please note that the output of each layer is the same dimensionality to facilitate the residual connections. This size is referred to as D_{model}.

The decoder part of the model also consists of a stack of $N = 6$ layers. The decoder implements an extra sub-layer in addition to the ones implemented in the encoder. This is a Multi-Headed Attention on the outputs of the encoder that also has a residual connection. The self-attention layer of the decoder is also slightly modified to "prevent positions from attending to subsequent positions"[68]. This ensures that the predictions for position j cannot attend to position $j + 1$.

2.2.2 Attention

Attention is a mechanism for models to decipher what it should focus on in a given input. "It is inspired by the biological systems of humans that tend to focus on the distinctive parts when processing large amounts of information"[45]. It has been applied across many tasks in deep learning with various implementations [2, 40, 45].

The transformer architecture presented by Vaswani et al. [68] uses an attention mechanism, doubt "Scaled Dot-Product Attention" by the authors, which is identical to the dot product attention presented by Luong et al. [40] with the addition of a scaling factor of $\frac{1}{\sqrt{d_k}}$ where d_k is the dimension of the queries and keys.

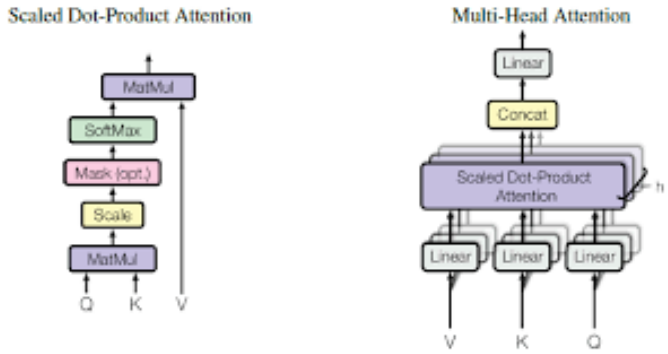


Figure 2.2. Visualizations from Vaswani et al. [68] of the multi-headed attention mechanism

The full multi-head self-attention architecture can be seen in Figure 2.2. The input to the "Scaled Dot-Product Attention" is the query (Q) and key (K) vectors of dimension d_k and value (V) vector of dimension d_v, which are calculated from the product between the input embeddings and the respective weight matrices W^Q, W^K, W^v. In

the base model these matrices have the size $d_k = d_v = \frac{D_{model}}{hiddensize=64}$ and their parameters are part of the training process. The actual implementation of the attention is compacted into a set of matrix multiplications described in Equation 2.1.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

The multi-head attention is h stacked Scaled Dot-Product Attention layers which allow the model to "attend to different representation subspaces at different positions"[68], improving the capability to focus on different positions. As mentioned earlier the output of each sub-layer in the model should be D_{model} which is why the h attention heads are concatenated and passed through a linear layer. This can be expressed as

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(head_1, ..., head_h)W^O \\ \text{where } head_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2.2}$$

Having the projections W_i^y being matrices of shape $W_i^y \in \mathbb{R}^{d_{model} \times d_y}$ where $y \in \{Q, K, V\}$

2.2.3 Tokenization, embeddings, and positional encoding

Transformer models cannot work directly with text; thus, a tokenization strategy must be used to break down the text into smaller discrete information elements. This can be achieved by methods such as the SentencePiece algorithm[33] implemented in ALBERT[35], XLNet[77], and T5[50] and the WordPiece algorithm[60] used in BERT[14].

Both are subword-based tokenizers, which lie between a word tokenizer that splits any sentence into words, usually on space, and the character-based tokenizer that splits a given text into characters. The benefit of a word tokenizer is that the model receives a different input for every word, which lets it better distinguish between elements, but this comes with the drawback of a considerable vocabulary size since the English language has 170k words, but only 256 characters[57].

The character level tokenizer resolves the issue but is faced by another. Since each token is passed to the model and transformer models have a limit on the length of input sequence accepted, a word like "contract" would have a tokenized size of 7, making the text span a model can see at any given time minimal. This thesis will not detail it, but the subwords-based tokenizers seek to combine the best from both approaches. The output of all these tokenizations is a list of discrete values and a lookup table to decode the encoding.

Transformers use learned embeddings comparable to other sequence transduction models that convert the input tokens to vectors of size D_{model} [68].

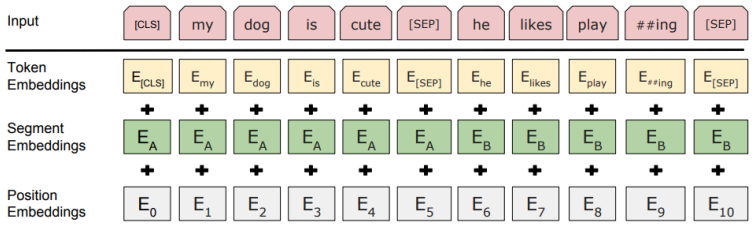


Figure 2.3. BERT Embedding visualization from Devlin et al. [14]

Since the transformer architecture does not process data sequentially with a recurrence or using convolutions, it does not have any sense of understanding of the ordering of words. Thus we encode information about the absolute and relative positioning of the words into the embedding. This is done using "positional encoding"[68] of the same shape D_{model} as the input embeddings, so they can be summed and not affect the input size, as seen in Figure 2.3. This is done using sine and cosine functions of specific frequencies as seen in Equation 2.3, but other methods also exist[18].

$$\begin{aligned} PE(pos, 2i) &= sin(\frac{pos}{1000^{\frac{2i}{d_{model}}}}) \\ PE(pos, 2i+1) &= cos(\frac{pos}{1000^{\frac{2i}{d_{model}}}}) \end{aligned} \tag{2.3}$$

"where pos is the position and i is the dimension"[68]. A graphic representation of the encoding can be seen in Figure 2.4

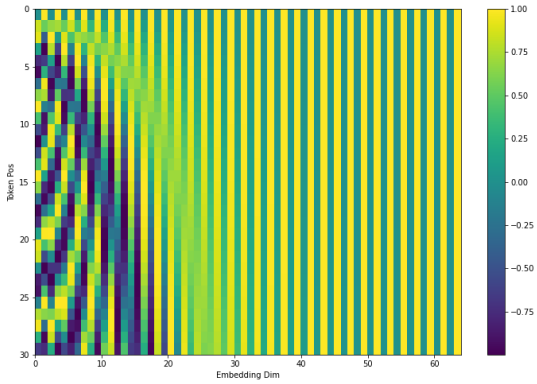


Figure 2.4. Positional encoding across embedding dimensions generated with code from Tensorflow[1]. Shows the token position along the y-axis and embedding dimension along the x-axis

2.2.4 BERT & RoBERTa

The transformer model used in this thesis, is based on the Bidirectional Encoder Representations from Transformers architecture (BERT) first presented by Devlin et al. [14] specifically the BERT-base model (see Section 5.1). This model uses an encoder-only architecture similar to the encoder introduced by Vaswani et al. [68].

This is because the pre-training of the model does not require an auto-regressive decoder structure. Instead, BERT is pre-trained using the unsupervised learning techniques masked language modeling and next sentence predictions (see Section 2.2.4) on 3,300 million words[14] and it achieved state-of-the-art performance when it was initially published on a large set of tasks within NLP[14]. The BERT model uses a 30,000 token WordPiece[60] embedding with a special set of tokens, to indicate elements such as separation of inputs and a classification token called [SEP] and [CLS]. Most transformer model a more extensive set of special tokens, some of which are shown in Appendix Table A.1.

It was later shown by Liu et al. [37] (RoBERTa) that the BERT model performance can be further optimized, extracting much more performance. As mentioned, BERT-based models are encoder-only architectures, but decoder-only architectures also exist, with the most well-known being the GPT-3 model presented by Brown et al. [6], which is often used for tasks such as text generation. This thesis is limited to experiments with encoder-only style architectures.

2.3 Training BERT Transformers

Learning strategies within machine learning can commonly be segmented into two groups: supervised and unsupervised learning related to labeled and unlabelled data. Unsupervised learning is often associated with outlier detection and clustering, whereas supervised is often used for classification and regression.

A large part of the dominance of the transformer-based model on NLP tasks has been due to the excellent results obtained from deep unidirectional architecture with rich unsupervised pre-training[14]. RoBERTa has been pre-trained on over 160GB of uncompressed unlabelled text data, and with a relatively small fine-tuning using supervised methods, it has shown great results[37] across well know NLP challenges such as SQuAD, GLUE[71], and RACE[34]. The implementation of this learning strategy can be segmented into two phases: the unsupervised pre-training using Masked Language Modelling and Next Sentence Prediction and the supervised fine-tuning. This second phase of the process often includes transfer learning where fine-tuned models or model segments are further optimized and fine-tuned for other tasks than the original purpose, such as a model developed for the SQuAD dataset being used as a starting point for the extraction of contract elements as seen in Chapter 6.

2.3.1 Pre-training

The MLM training procedure is focused on training a deep bi-directional representation within the models. It is done by taking a fraction of the tokens in the model input, masking them, and then predicting the original label of those masked tokens. The objective function of this task is a cross-entropy loss on the labels of the masked tokens[37], that is obtained by passing the transformer output through an MLM-head, similar to the logic presented for extractive QA presented in Section 4.1.2. However, instead of having an output shape equal to the input span, the output size is equal to the vocabulary size. The task looks like the one presented in Equation 2.4 for a model using a word-based tokenizer.

$$\text{[CLS] The name of this [MASK] is Supplier Agreement [EOS]} \tag{2.4}$$

NSP is focused on incorporating cross-sentence relationships between two sentences. It is a trivial training process where the model is provided with two sentences A and B where there is a 50% chance that B is the correct sentence following A in the corpus and 50% that it is not. Both these tasks have been shown to be very beneficial for RC and Natural Language Inference (NLI) tasks[14], but Liu et al. [37] later proved that removing NSP from the pre-training actually can be beneficial for performance on some tasks. Like MLM training, a transformer on NSP also requires a particular *head*, but in this case, it just has two output logits indicating if sentence B is the next sentence. This is at the core of the versatility of the transformer models, as adapting to a new task often only requires changing the model *head*.

2.3.2 Fine-tuning and transfer learning

Transfer learning is a learning strategy where performance on a target task is improved by utilizing representations learned on a source task[55, 72] and it is part of the reason why transformer models have performed so well on a wide range of NLP tasks[72] along with the easy access to state-of-the-art pre-trained models like BERT and RoBERTa through open-source frameworks like `Transformers`.

For usage of these pre-trained models in span-selection QA, the most straightforward approach is to pass the output of the transformer through a fully connected layer, having the output produce two values for each position in the input related to the likelihood of a given position being start- and end position of the answer (see Section 4.1.1). Other tasks include spam classification, where the simple implementation has a single or two output logits after a fully connected layer for predicting spam or non-spam based on the entire input. Typical for these tasks is that they can often be trained with very few resources compared to the unsupervised part, as will be seen in Chapter 5 and 6 and shown by Liu et al. [36] for the binary classification problem related to spam filtering. Due to computing restraints specified in Section 3.3 this thesis will not investigate the improvements that can be obtained from the pre-training.

2.4 Performance metrics

In QA tasks like SQuAD 2.0, rankings and performance measures are based on the exact match `EM` and `F1` scores. Exact match is defined as predicting the exact answer for a given context, often with the answer and prediction cleaned for whitespace characters and punctuation and converted to lower case. `F1` is a less strict metric related to the average overlap between question ground truths and predictions. It is calculated by treating the prediction and ground truth as a bag tokens and computing their similarity as in Equation 2.5 where tp is the number of tokens that are shared between the prediction and ground truth, fp is the number of tokens in the prediction that is not in the ground truth label, and fn is the number of tokens that are in the ground truth but not in the prediction[51].

$$\begin{aligned} F1 &= \frac{2 \cdot precision \cdot recall}{precision + recall}, \text{ where} \\ precision &= \frac{tp}{tp + fp}, \; recall = \frac{tp}{tp + fn} \end{aligned} \tag{2.5}$$

Evaluating a given model's performance is done using a Macro EM[51], termed `EM` in this thesis, which is the total percentage of prediction that matches any of its ground truth labels across the dataset. The macro F1 score, termed `F1`, the maximum F1 score between the predictions and the answers associated with a given question, averaged over all the questions.

If the predictions have confidence probabilities, one can seek to tune the model's precision/recall trade-off by thresholding the confidence probability that determines whether something is a prediction. In these cases, measures such as Area Under the Precision-Recall, "Precision at %X Recall," and the AUPR curve are often used. The measures used to assess model performance will be further discussed in multiple sections throughout the thesis.

CHAPTER 3

Data and tools

This section presents central elements concerning the training setup, machine learning frameworks, and data sources. As this thesis only experiments with the fine-tuning stage of the transformer model training, the pre-training dataset will not be discussed. For detail surrounding that, the reader is referred to the original papers BERT by Devlin et al. [14], Attention is all you need by Vaswani et al. [68], and RoBERTa by Liu et al. [37].

3.1 Stanford Question Answering Dataset

The Standford Question Answering Dataset (SQuAD) is one of the largest and most popular publicly available datasets for reading comprehension tasks within machine learning[51, 52, 53]. It exists in two versions 1.1 and 2.0. The 1.1 dataset is a collection of "questions posed by crowdworkers on a set of reading passages from Wikipedia articles, where the answer to every question is a segment of text, or span from the corresponding passage. The 2.0 dataset expands this by adding questions that are unanswerable[53]. Together, the dataset has 100.000 answerable and 50.000 unanswerable questions, where crowdworkers have written the impossible questions to look similar to the answerable ones.

From table 3.1 one can see the general structure of questions posed to the reading passage found in Appendix B.1. The reasoning for including unanswerable questions is to make the models applicable to use cases where the answer is not guaranteed to exist in the text[51]. In the remainder of this thesis SQuAD will reefer to version 2.0 of SQuAD and the dataset will only be used in Chapter 6 for experiments with transfer learning and its effect on performance, thus for a more in-depth understanding of

Question	Answer
Who was the Norse leader?	Rollo
What century did the Normans first gain their separate identity?	10th century
What is France a region of?	<EMPTY>

Table 3.1. Question examples from the SQuAD dataset. The associated text paragraph can be found in the appendix B.1

the dataset the reader is referred to the original papers presenting the dataset or derivative analytical work.

3.2 Contract Understanding Atticus Dataset

The Contract Understanding Atticus Dataset (CUAD) is a collection of contracts with more than 13.000 annotations created by legal experts related to 41 different questions[23]. The dataset and material associated with the paper[23] consist of the 510 contracts obtained from the U.S. Securities and Exchange Commission (SEC) through the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system. The dataset provides these documents in PDF and text formats, where the text version likely is obtained using OCR since it contains Unicode characters one would not expect from a human transcription, such as zero width space (\u200b) and no-break space (\xa0), but this is not mentioned in the paper or any associated resources[23]. The paper also includes 8.6 gigabytes of unlabelled contract data from SEC filings useful for MLM or NSP.

The authors of CUAD have provided the contracts in a pre-processed format similar to the one presented with SQuAD[51, 52]. This is a JavaScript Object Notation (JSON) file that consists of a list of data entries where each entry corresponds to a contract with the associated questions and their respective answer spans and a binary variable indicating if the question is impossible to answer. This allows for the utilization of the large ecosystem around the training of models on the existing SQuAD dataset.

The 41 questions used for annotations are formulated in the same way as seen below. The "<Label Category>" is one of the 41 categories seen in Appendix Table B.1 such as "Non-Compete", and the "details" are definitions of that label category created by the authors also found in Appendix Table B.1.

> Highlight the parts (if any) of this clause related to "<Label Category>".
> Details: <Label CategoryDescription>[23]

Every contract thus has 41 questions, each with ≥ 0 answers, that can be located anywhere in the contact.

3.2.1 Quality of annotations

According to Hendrycks et al. [23], the CUAD dataset might be the best high-quality publicly available dataset for the training of deep learning models for extracting legal contract elements, which is also evident from the paper title "An Expert-Annotated NLP Dataset for Legal Contract Review"[23], but it does have some caveats. During an analysis of answer spans, it was discovered that the actual text marked as the answer sometimes occurred multiple times in the contract text, but with only the first occurrence marked as the answer. As seen in Figure 3.1 the total number of marked

answers in the training dataset is around 11k, but exact and fuzzy matching on the answer spans reveals approx 110k and 920k matches, respectively. Correcting for categories with the significant class imbalances, along with filtering out dates where it is difficult to say with high confidence that the text should have been marked as an answer, the exact match on answers is 7227, fuzzy matching answers are 7617, and the original dataset has 7103 marked answer.

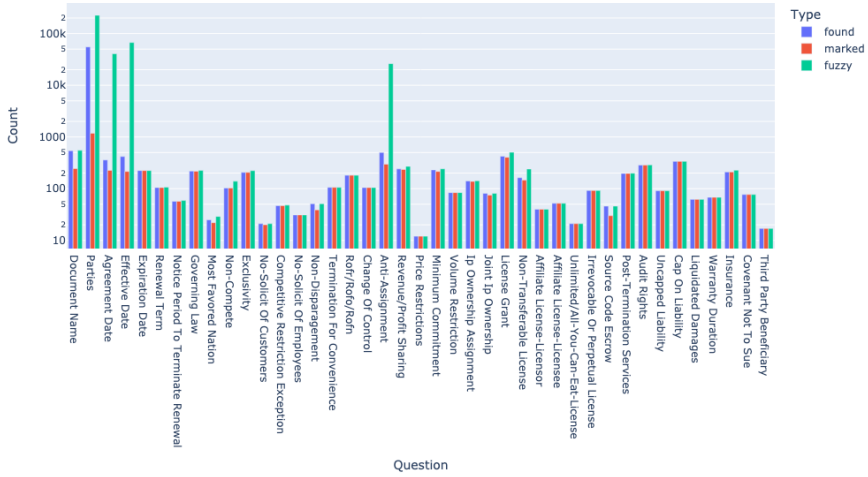


Figure 3.1. Distribution of marked answers across categories on the training dataset. Exact match is the answers as marked in the dataset. The fuzzy matching is the same, but with the text cleaned of spaces and special characters.

There is an argument to be made that there will be some of these exact and fuzzy matching answer spans that are false positives, where one cannot infer that this is the correct answer to the question. An example is the same date occurring in a different context. Another example is with acronyms for one of the parties used where it is not immediately clear if it is a contractual party, a third party, or something else. That being said, the dataset does contain examples like the following: The contract named "XinhuaSportsEntertainmentLtd_20070221_F-1_EX-99.4_645553_EX-99.4" is between "CHINA ECONOMIC INFORMATION SERVICE OF XINHUA NEWS AGENCY" and "XINHUA FINANCIAL NETWORK LIMITED".

> "...THIS AGREEMENT is made the 15th day of December 2001.\nBETWEEN \n \n(1) CHINA ECONOMIC INFORMATION SERVICE OF XINHUA NEWS AGENCY, the organisation within the Xinhua News Agency that is responsible for news and information operations and business, registered in the People\'s Republic of China with offices at 57

Xuanwumen Xidajie, Beijing, the People\'s Republic of China (" CEIS "); and\n\n(2) XINHUA FINANCIAL LIMITED LIMITED ..."

From the snippet above, only "CEIS" is marked as an answer to the *Parties* question, even though the dataset has the full company name marked as an answer elsewhere. In addition to this Chalkidis et al. [11] also raise the following concerns:

- Some answers can be semi-redacted such as "_____, 1989" as an answer for the agreement date
- Indirect mentions can be part of the answers such as "supplier" as answers to the *Parties* questions
- The dataset contains both short entity-level and long paragraph-level answers such as "Service agreement" for the *Document Name* and "Imprimis Pharmaceuticals, Inc. for *Parties* and paragraph-level (long) answers (e.g., If any of the conditions specified in Section 8 shall not have been fulfilled when and as required by this Agreement, or by the Closing Date, or waived in writing by Capital Resources, this Agreement and all of Capital Resources obligations hereunder may be canceled [...] except as otherwise provided in Sections 2, 7, 9 and 10 hereof." for *Termination for Convenience*

These are all elements one must consider when evaluating training performance and optimization. They will be further investigated and experimented with in Chapter 6 to search for improvements in model performance. Additionally, it should also be noted that the unanswerable part of this dataset is not created in the same way as SQuAD. The unanswerable questions in SQuAD 2.0 are written adversarially to look similar to the answerable questions, but where the context does not contain the actual answers[53]. In the SQuAD example in Appendix Section B.1, this would be a question like "What is France a region of?" [52], which seems related, but is unanswerable from the context. The unanswerable questions from CUAD are simply a random selection of contract spans with one of the 41 different questions associated, which arguably is less informative and should be a more straightforward challenge.

3.2.2 Question type occurrence

There is a difference in where one can expect to find the answers to the posed questions dependent on the question category. The question types can be segmented into three different categories. The ones most commonly found on the contract's first couple of pages in what is termed the *preamble* by Chalkidis et al. [9]. These are elements such as the *Document Name*, *Parties*, *Agreement Date*, and *Effective Date*. The small group of elements that look to be located towards the end of the contracts, including *Governing Law*, and *Third Party Beneficiary* and the remainder of questions that look to be distributed more uniformly across pages such as *Audit Rights*, *Cap On Liability*,

and *Non-Compete*. In Figure 3.2 one can see the distribution of the relative position of the answers in their respective contracts using the segmentations mentioned above. The answer location used in the figure is defined at the position of the first character marked as the answer. The figure clearly shows that the marked answers for some question categories have a noticeably higher probability of being in particular sections of the contract. The plot is only based on the originally marked answers, as this thesis does not have the resources or competencies to check and validate the potential answers presented in Section 3.2.1.

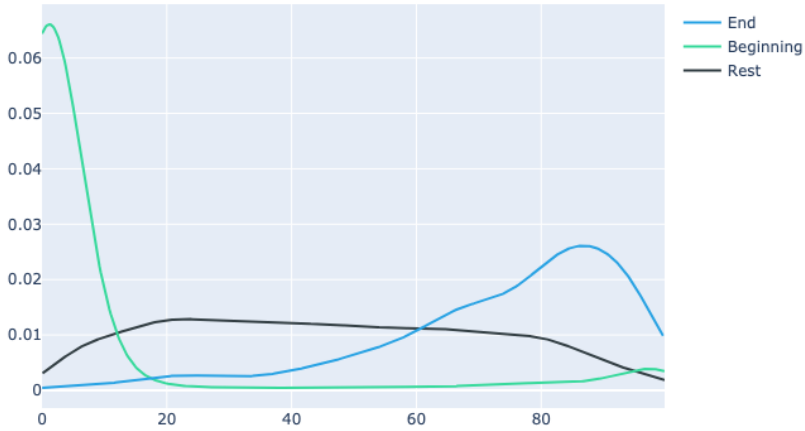


Figure 3.2. Answer segmentation distribution using segmentation indicated in Appendix Table B.1

3.2.3 Preprocessing of data

The preprocessing of the dataset takes the aforementioned JSON file and converts it into three elements: a list of `examples`, a list of `features`, and a `dataset`. Each example constitutes a single question, contract $(\mathbf{q}, \mathbf{C})$ pair. This pair consists of the answers (if any), the context, which in this setting is the full contract text, an unique id, and the actual question. As one `example` is created for each question and contract pair a total of questions $\times$ contracts $= 41 \times 510 = 20,910$ `examples` exists.

The `features` are created from the `examples` and each `feature` constitutes a data point for training/testing/validation and is a section of the contract that can fit into a transformer model, meaning that the feature's total tokenized size $\leq D_{input\_size}$.

This also means that the `features` are dependent on a given model's tokenizer. Thus BERT and RoBERTa would be unable to use the same dataset due to the different tokenization strategies. The number of `features` is dependent on the hyperparameter termed `doc_stride`, which controls how a context that is larger than D_{input_size} is tokenized. The context is split into multiple `features` using a sliding window, where the sliding window is moved `doc_stride` tokens on each step. Thus each `feature` is some section of the contract that potentially overlaps with other `features`, a question, a boolean value indicating whether or not this section contains an answer, the answer locations if there are any, and multiple helper methods for converting back and forth from the tokenization and original contract.

The `dataset` is simply the `features` in a format that can be processed by the deep learning framework consisting of the following: start and end positions of the answer spans, the token type ids, the [CLS] token index, the p mask containing information about where the tokenized answer is located, the binary indicator variable showing if there is an answer the question, and finally an index variable for associating the data point with the original `feature` object.

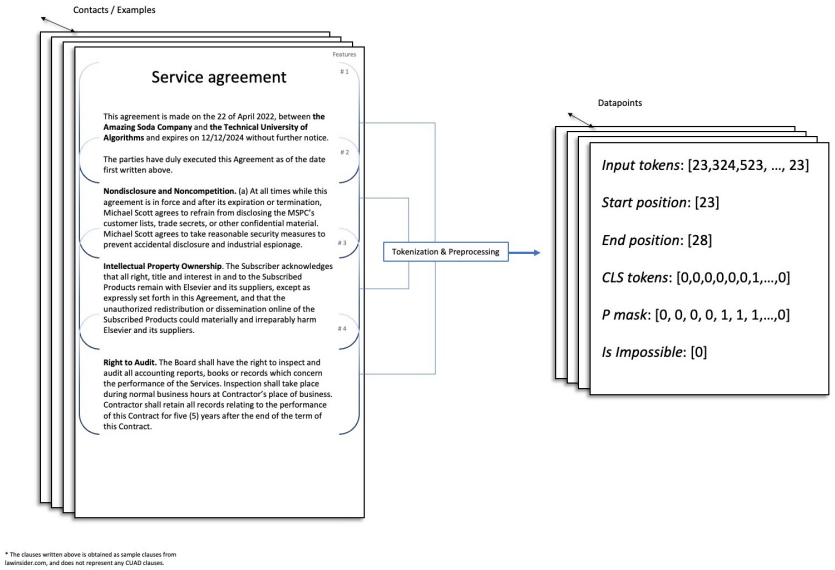


Figure 3.3. Contract processing flow for converting a single contract into a series of `features` and data points

With more than 99% of sections not containing an actual answer using a sliding window size of 256 tokens and a maximum question size of 64 tokens, the dataset is heavily imbalanced towards non-answerable questions. This can have unintended effects on performance[32], and consequently, methods for extracting a balanced dataset have also been implemented that randomly removes a portion of the impossible an-

swers.

Initially, the entire logic for preprocessing the data described above was implemented from scratch, but to ensure better comparability between the work done in this thesis and the work done by Hendrycks et al. [23], their framework, originating from the `transformers` library, was eventually adopted.

This thesis does not experiment with the actual sliding window size thus it can be assumed from hereinafter that the `doc_stride` = 256 and `max_query_length` = 64. After pre-processing, which currently requires around 300GB of RAM for 60 workers, the list of `examples` in its pickled size is 7.5GB, the list of `features` 12.5GB, and the `dataset` 16.3GB. The train `dataset` has a total size of 1.134.631 datapoints with 15.519 being answerable and 1.119.112 being unanswerable. The test set has a total size of 154.627 datapoints with 2.494 answerable questions and 152.133 unanswerable questions using the same notion. Please note that the number of answerable datapoints is larger due to the sliding window procedure and for reasons explained in Section 6.1.1.

3.3 Experiment Environment and MLOps

Prior to and throughout the modeling phase of the thesis, a large amount of resources went into creating a reproducible and capable machine learning setup that could allow for proper training and testing of the models in different configurations. This also includes experiments with distributed data processing to speed up training and validation. This thesis will not cover these experiments, which are evident from the source code.

All machine learning experiments listed in the report have been performed on Google Cloud AI Workbench, a hosted instance of JupyterLab[1], using the open-source machine learning framework Pytorch[46]. This was chosen as it provided a controlled environment for experiments and allowed for easy infrastructure scalabilities such as memory and GPUs. The compute instance was an n1-highmem-32 instance hosted in the europe-west4-b zone with 4 Nvidia Tesla T4 GPUs. The details of the compute instance and python library versions can be found in the thesis source code repository. To facilitate the training of the models and streamline the process of setting up and maintaining the multi-GPU environment, the Pytorch Lightning framework[17] was chosen in combination with the Weights & Biases developer tool[2] for tracking progress and sharing results.

[1] `https://jupyter.org/`
[2] `https://wandb.ai`

3.4 Hugging Face Transformers

Hugging Face is a leading NLP company that has built a large community around its open-source NLP frameworks and it is the company behind the popular Python library Transformers[75]. Transformers provides pre-built and pre-trained standardized implementations of popular transformer models such as RoBERTa[37], BERT[14], and XLNet[77] built directly on top of the Pytorch ecosystem. Furthermore, it allows for the loading and sharing of model checkpoints in a well-documented and simple way.

As part of the initial phase of research into the transformer architecture, the first transformer model was built from scratch in Pytorch, based on the architecture presented by Vaswani et al. [68] and implemented it as by Rush [56]. It was trained and tested on the SQuAD RC challenge to have a known benchmark to confirm correct workings.

This approach was discarded for multiple reasons. Firstly, the goal of this project, as described in Section 1.2, is to improve the model performance on the CUAD dataset and share the results, with the aim of letting it be used by others. Thus it does not make sense to invest resources into developing a new transformer framework, when one already exists that is widely adopted. Furthermore, the `Transformers` library comes with the benefit of better optimized vectorized operations, rapid experimenting with multiple model architectures, and most importantly pre-trainied model checkpoints for BERT and RoBERTa. This thesis is limited to a single host multi-GPU setup, rendering training a full model impossible as the authors of RoBERTa used "1024 V100 GPUs for approximately one day"[37]. Consequently, `Transformers` was chosen for the transformer architecture part of the model. In Appendix Figure A.1 is an example of how simple an extractive QA prediction can be made using the library.

CHAPTER 4

Methodology

This chapter covers the initial modeling choices used in the experiments and analysis presented in Chapter 5. The modeling is based on the RoBERTa-base model from Hendrycks et al. [23] along with the associated CUAD source code. In addition, the chapter presents how the task-specific heads have been modeled and the logic for calculating predictions along with the loss function and evaluation strategy. The architecture and framework presented here will be evolved in Chapter 6 to enhance model performance.

4.1 Model architecture

The answers to the dataset question are not common knowledge that the models can internalize. Thus we formulate the problem of obtaining contract elements as an extractive QA task where the models should mark the correct answer span in the context if there is one.

The model architecture is as seen in Figure 4.1 with the transformer output of shape $D_{input\_size} \times D_{model}$. To use the model for extractive QA, the output of the transformer model gets passed through a simple, fully connected layer termed the QA *head* to obtain the prediction logits, a process explained in Section 4.1.1. A post-processing procedure is applied to these output logits to extract the actual model predictions.

The learned tokenizations will not be discussed in this thesis as Liu et al. [37] showed, based on the work of Radford et al. [49], that changing the tokenization from the learned character-level Byte-Pair Encoding (BPE)[61] in the original paper transformer paper to a larger byte-level BPE only slightly altered performance on some NLP tasks for the worse. Furthermore, the limitations outlined in Section 3.3 hinder experiments with other tokenizations as it could require rerunning parts of the unsupervised learning described in Section 2.2.4, which is something outside the computational budget of this thesis.

4.1.1 RoBERTa input and output

Inspired by Devlin et al. [14]'s approach to solving the SQuAD v1.1 challenge, the models tested by Hendrycks et al. [23] take the question and context $(\mathbf{q}, \mathbf{c})$ pair as a

single packed sequence model input. This utilizes the model's learned structure from NSP, where the input is split an A and B embedding separated by a special $[SEP]$ token and with each segment having a learned embedding added to every token to indicate which segment it belongs as seen in Figure 2.3. The question and answering head QA_h implemented uses a `in_features` : D_{model} and `out_features` : 2 fully connected layer. The notion is that each vector constituting the transformer output of an input token is converted into two logits, with the first dimension termed $S \in \mathbb{R}^H$ related to the likely hood of the answer starting at this token position and the second termed $E \in \mathbb{R}^H$is related to the likelihood of the answer ending at this location. Thus a $(\mathbf{q}, \mathbf{c})$ pair results in a $2 \times D_{input\_size}$ output.

It should be noted that any $(\mathbf{q}, \mathbf{c})$ sequences that do not have a length of D_{input_size} in its tokenized state is padded or truncated to the D_{input_size} mentioned in Section 3.2.3, as the model does not have a mechanism for handling other sizes of inputs. This thesis used $D_{input\_size} = 512$ as this is the shape used in RoBERTa-base and BERT-base [14, 37].

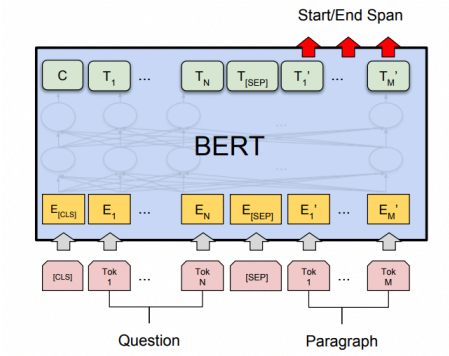


Figure 4.1. Illustration of the QA task adaptation of BERT presented by Devlin et al. [14]

4.1.2 Making predictions

Using the notation from Figure 4.1 where T_i' is element i of the output sequence corresponding to the context segment B and T_i the element i in the question segment A. Converting the transformer outputs to probabilities is then done as shown in Equation 4.1.

$$PS_i = \frac{e^{S(T_i)}}{\sum_j e^{S(T_j)}} \text{ and } PE_i = \frac{e^{E(T_i)}}{\sum_j e^{E(T_j)}} \tag{4.1}$$

, where $S(T_i)$ and $E(T_i)$ are the logits obtained from $QA_h(Tok(\mathbf{q}, \mathbf{c}))$

The probability of a prediction for a candidate pair is defined as $PS_i \times PE_j$. This implementation uses the same logic as CUAD, where the prediction score is defined

as the sum of logits S_i and E_j for a candidate pair where $i \leq j$ because S and E are treated as independent and since softmax is a monotonic increasing function. In the general $top_k = 1$ case we predict $\hat{s} = \max_{i \geq j} S_i + E_j$, where $\hat{s}$ is not the empty/null answer when $\hat{s} \geq s_{null} + \tau$ with τ being a threshold values for the confidence needed for predicting a non-empty answer. This values is often selected on a dev set. How the model actually makes an empty prediction is presented in Section 4.2. This is the theoretical notation underlying the idea of the prediction logic. However, the implementation, as seen in Section 4.2 differs a bit from it due to implementation choices and data processing considerations, which often leads to the set of possible values for i, j being drastically reduced. It should also be noted that $i, j \notin T$ as we never predict the answer to be part of the question.

4.1.3 Objective function

The objective function of the training is the average cross-entropy loss between the S and the start position of the answer plus E and the end position of the answer. The loss is implemented using Pytorch[46] and can be described as

$$\ell(x, y) = L = \{l_1, \ldots, l_N\}^\top, \quad l_n = -w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \cdot 1\{y_n \neq \text{ignore\_index}\} \tag{4.2}$$

where x is the predicted logits, y is the target logits, w is a weight matrix for scaling class loss, N is for the batched setting and spans the data points, and the *ignored_index* is used for the case where the start/end positions are outside the model inputs (these terms are ignored). In the batched cases, as done in this thesis, a mean reduction is applied to the output $\ell(x, y) = \sum_{n=1}^{N} \frac{1}{\sum_{n=1}^{N} w_{y_n} \cdot 1\{y_n \neq \text{ignore\_index}\}} l_n$. The target logit for an empty prediction is the [CLS] token at index 0. This is because the "final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks" [14] and has been used extensively in the model pre-training.

The optimizer used for the fine-tuning is the weight decay variant of the Adam[30] named AdamW[39]. The configuration is the same as in CUAD[23] with $\gamma = 0.0004$(lr), $\beta_1 = 0.9$, $\beta_2 = 0.999$, (betas), $\lambda = 0.01$(weight decay), $\epsilon = 1e-08$(epsilon). The optimizer is implemented using a learning rate scheduler that decreases γ linearly from the initial setting to 0 across the entire training. In this thesis's training loop, a warmup period of 100 steps was used where γ increases linearly from 0 to the configured γ. A setting that seems to be close common in transformer fine-tuning[23, 75]. The full list of specific training hyperparameters used can be found from the WandB interactive plots in Appendix Section A.1. The selection of these hyperparameters is based on what has shown good performance in literature [14, 22, 23, 37] and a few experiments. This thesis will not cover model optimizations through RoBERTa training hyperparameters, though the author acknowledges the potential for performance increases it has.

4.2 Evaluation framework

The evaluation framework is based on the notion that a paralegal or lawyer should verify all predictions and that the model should be presenting a thresholded top_k or *n-best* set of elements for each $(\mathbf{q}, \mathbf{C})$ pair. This differs from the objective function, but it is assumed that the task is close enough for the objective function to be a good proxy for optimization, something that will be investigated in Chapter 5 and 6.

The first step of the evaluation framework is computing the *n-best* predictions from the model outputs. The outputs are passed through Algorithm 1 that calculates the *n-best* predictions for each `example`, which consitutes a $(\mathbf{q}, \mathbf{C})$ pair, as presented in Section 3.2.3. It should be noted that *n-best* is a parameter determining the number of predictions to extract for a given example while N_{best} is the set returned by Algorithm 1 containing the actual predictions with meta-data. In the algorithm, the term *valid* predictions reefers rules determining what candidates are allowed. For example, a prediction where the start position is after the end position would be discarded. All inputs to the model are padded or truncated to the size D_{input_size}. This also means that the actual context inside the tokenization might stop at token $\frac{D_{inputsize}}{2}$. These padded predictions are also eliminated. All example predictions get their highest likelihood empty prediction added to N_{best} if it is not already a part of N_{best}. The probability of each prediction is then calculated by computing the softmax over the prediction scores in N_{best}. For additional information on the filtering of candidates, the reader is referred to the source code repository that can be found through the link in Appendix Section A.1.

$$\mathrm{P}(s_i) = \frac{\exp(s_i)}{\sum_{j \in N_{best}} \exp(s_j)} \ , \tag{4.3}$$

Where s_i and s_j are candidate scores $\in N_{best}$

Algorithm 1 Calculate top-n predictions from the model output

The algorithm makes use of function calls to improve readability. For a more in-depth explanation, the reader is referred to the associated source code.

procedure COMPUTE-PREDICTIONS(Examples, outputs, n-best)
    N_best ← *array[]*
    feature-to-output ← *lookup table from feature to associated output*
    for *each example in Examples* **do**
        prelim-preds ← *array[]*
        for *each feature in example* **do**
            result ← *feature-to-output(feature)*
            $start\_pred = get\_n\_best(results.start\_logits, n\text{-}best)$ ▷ Sorts logits and outputs n-best of them
            $end\_pred = get\_n\_best(results.end\_logits, n\text{-}best)$
            $prelim\text{-}preds.append(extract\_valid(start\_pred, end\_pred, feature))$ ▷ Obtain all valid permutations of start and end logits
        end for
        sorted_preds ← *top-n-preds(prelim-preds,n-best)* ▷ n-best predictions determined by sum of start and end logits
        N_best.append(postprocess(sorted_preds)) ▷ Add empty prediction if not present and include metadata
    end for
    return *N_best*
end procedure

The model performance is evaluated based on the N_{best} predictions by the AUPR metric, AUPR @ 80% recall, and AUPR @ 90% recall making the output dependent on the *n-best* parameter. This parameter is fixed at $n\text{-}best = 20$ in this thesis, as this is the default value in the CUAD source code, and the value is not mentioned in the paper. First precision and recall are calculated at 100 uniformly distributed confidence steps between 0 and 1, meaning that at each step, all predictions in N_{best} with lower confidence than $conf_i$ are not considered. Using the filtered predictions, precision and recall are then calculated for every question and using these values, the area under the curve is calculated using the trapezoidal rule implemented in scikit-learn[47]. This is the evaluation approach presented as relevant for the problem at hand by Hendrycks et al. [23]. This thesis extends the evaluation to some of the metrics used in SQuAD, such as evaluating the top prediction using `EM` and `F1` on both unanswerable and answerable questions. These approaches uses the methodology presented in Section 2.4 with a normalized answer for `EM` and the bag of token approach for `F1`. This is included as it provides a more nuanced picture of model performance across a wider set of tasks, in the authors' opinion.

CHAPTER 5

CUAD Reimplementation and Analysis

This section will on the basis of Chapter 3 and 4 explore the performance of the CUAD RoBERTa-base model presented by Hendrycks et al. [23]. It seeks to reproduce the results presented in the paper. From there, it will do a thorough analysis of the model performance. This analysis will dig into the flexibility of the model in question formulations, along with explorations and interpretations of the attention weights produced by the models in a qualitative way. Furthermore, it will look at the evaluation metrics presented in the paper along with the proposed ones from Chapter 4 to assess the model's performance and flexibility. It also seeks to understand the performance difference across the different question categories put forth in CUAD.

5.1 Establishing the baseline

The RoBERTa-base model is the smaller of the two RoBERTa models presented in CUAD, and the only model experimented with in this thesis even though it has been shown that task can benefit from training wider and deeper models and then compressing them. It has the following parameters (L=12, H=768, A=12, Total Parameters=110M) where L is the number of layers, A is the number of attention heads, and H is the hidden size.

Table 5.1 shows the paper listed performance for RoBERTa-base, RoBERTa-large, the performance of the paper RoBERTa-base model checkpoint obtained through the CUAD source code repository[23], and the best performance found from training a RoBERTa-base model from scratch using the CUAD source code hyperparameters. The original results table presented in CUAD can be found in Appendix Table C.2.

From the results, it is evident that we can reproduce the results presented in CUAD. The **RB-S** improves the performance in terms of AUPR and Precision @ 80 % Recall, compared to the paper listed performance. It is also clear that the provided

Metric	Eval on	Model/Measure	RB-P	RL-P	RB-P-C	RB-S
CUAD	n-best	AUPR	42.6	45.2	33.17	50.39
		PREC @ 80%	31.1	34.1	15.26	37.32
		PREC @ 90%	-	-	-	-
OWN	top 1	Has Ans F1 (1244)	-	-	81.26	80.73
		No Ans (2938)	-	-	55.07	81.62
		EM	-	-	58.92	77.60
		F1	-	-	62.86	81.37

Table 5.1. Comparison table from results obtained from model checkpoints, reported scores, and training of the RoBERTa base model from scratch. RB-P is the RoBERTa-Base paper reported score. RL-P is the RoBERTa-Large paper reported score. RB-P-C is the RoBERTa-Base paper checkpoint as the inference model in the thesis validation loop, RB-S is the RoBERTa-Base model trained from scratch using paper hyperparameters. - *indicates not applicable*

checkpoint has not been fine-tuned to the state of model performance listed in the paper, which is evident from the large discrepancy in AUPR between **RB-P** and **RB-P-C**. Looking at the results, it seems like the increase in performance of the model AUPR of **RB-S** is related to it being better, 81.62% vs 55.07%, at correctly predicting the empty span in the top 1 prediction. The **RB-S** checkpoint is the checkpoint obtained from training with the highest AUPR, but the `Has Ans F1` measure is lower for **RB-S** than **RB-P-C**. Other checkpoints from the same **RB-S** run, have a higher score for the non-AUPR measures, thus $A_{PREC@80\%} > B_{PREC@80\%} \not\Rightarrow A_{AUPR} > B_{AUPR}$, but a high AUPR generally seems to indicate to good performance measures by the remaining metrics as will be seen in Section 5.2.

It should be noted that the evaluation framework used in this thesis, although based on the same logic, can be performed a configurable amount of times through a training epoch, meaning it has the option to save checkpoints "mid" epoch. This is not the case for the CUAD source code. The CUAD training loop does not evaluate performance as part of training but saves the model weights every epoch and calculates the scores for checkpoints afterward.

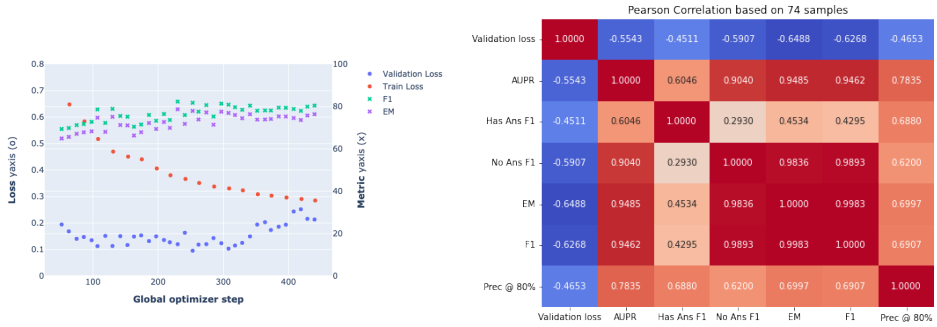
5.2 Validation loss as a proxy for test performance

As mentioned in Chapter 4, we do not directly train the models on the evaluation metrics, but instead on the cross-entropy loss presented in Section 4.1.3. This is the standard for training transformer models[14, 37, 51, 53, 75] for extractive QA. The CUAD paper does not argue that the objective function chosen is a good proxy for the evaluation metrics, but looking at the source code, it is directly using logic implemented in `Transformers` for the SQuAD extractive QA task. This is an implicit argument that the same objective function used for SQuAD will obtain good results

on CUAD as the task is fairly similar.

The Pearson Correlation Coefficient depicted in Figure 5.1(b), is a measure of linear correlation between two variables and is calculated as the relationship between the covariance of the variables and the product of standard deviations of the samples. The figure shows a clear linear correlation between the validation loss and the evaluation metrics, indicating that the objective function should embed some helpful knowledge into the model for the task at hand.

The correlation coefficient looks as expected with a negative sign due to the goal of maximizing the evaluation metrics by minimizing the validation loss. Extracting the R^2 values reveals that there still is much variance that the correlation model cannot explain, and though 0.55 is a decent correlation, this also means that there is room for improvements. The Spearman Rank Correlation Coefficient[16] showed similar patterns that can be found in Appendix Figure C.1.



((a)) Metrics and loss

((b)) Pearson correlation

Figure 5.1. (a) **RB-S** training run metrics and loss with signs of overfitting in validation loss, but little to none effect on metrics. (b) Pearson correlation between validation loss and the monitored metrics based on 74 samples obtained from 3 training runs on CUAD using the **RB-S** training setup.

It should be noted that Pearson correlation looks at a linear correlation between the variables; thus, the connection between minimizing the loss and improving model performance might be more robust than suggested by the plot.

In Figure 5.1(a) is a scatter plot of the training and validation loss along with the top 1 `F1` and `EM` metrics. This plot further supports the notion of room for improvements between the evaluation metrics and the model objective function, as the training and validation loss "curves" indicates that the models start overfitting around step 300, but the evaluation metrics seem to be affected little to none by it.

Looking at Figure 5.2 it is noticeable that there are many AUPR samples where a high AUPR does not necessarily lead to a low validation loss. These findings combined suggest that in the final steps of the training, where the learning rate is relatively low due to the linear scheduler, the model might not be finding a local minimum optimal for the metrics. This will be further discussed in Section 6.4.

An appropriate next step could be to investigate this correlation between the metrics and SQuAD, as the modeling used for CUAD has shown excellent results here[51, 52, 53], and SQuAD does not have the challenges faced by CUAD raised in Section 3.2.1 and by Chalkidis et al. [11], but this has been left for future work.

Figure 5.2. Evaluation metrics as a function of validation loss based on 74 samples obtained from 3 training runs on CUAD using the **RB-S** training setup

5.3 Evaluation framework considerations

As mentioned in the Introduction (Chapter 1) and in the Research Question and Objective (Section 1.2) an obvious implementation of these models with the current performance is to work in unison with lawyers and paralegals to speed up and improve processes like contract review or analysis. The metrics used to assess this performance are AUPR, Precision @ 80% Recall, and Precision @ 90% Recall. CUAD brings up the argument that precision of around 30% at these recall levels is roughly equivalent to $\frac{1}{3}$ clauses being relevant, meaning that a lawyer would need to read two irrelevant clauses/predictions for every correct one[23]. These metrics leave out some critical information about the performance.

In Table 5.1 it's evident that 1244 of the $(\mathbf{q}, \mathbf{C})$ pairs has an answer and 2938 does not. The same table also shows that performance increases can be obtained in terms of AUPR and Precision @ 80% Recall by improving the performance on the empty predictions.

Based on two interviews with digital transformation project managers at a major Danish legal company and a New York-based legal technology startup, this thesis will

argue that a better indicator of value created by this technology is in the precision at a low top n like top 3 or top 5 for the **Has Ans** predictions. This is because the **No Ans** predictions would need to be confirmed, as the model architecture does not present an argument for why the answer is not in the contract. Thus there is not an easy way of checking the model prediction in a human-in-the-loop system. This is due to the way the model predicts the empty span: this is "just" predicting the [CLS] token as seen in Section 4.1.2. Though it might provide an excellent indication to the lawyer or paralegal, model performance is not good enough to rely on. On the other hand, the **Has Ans** predictions include an argument for what the answer is - the predicted span, allowing the lawyer or paralegal to validate the prediction quickly. An analogy for this is finding and validating a solution to NP-complete problems.

In Figure 5.3 the relationship between recall and the number of predictions is shown. Though the standard deviation and number of predictions are relatively low on the high confidence threshold (left side) sections, one notices that around the 80% recall point, the number of predictions grows drastically. With a $\mu = 1.4$ and $\sigma = 2.7$ at 80% recall, the number of contracts requiring review might change drastically from case to case. The distribution of the number of predictions at this recall level can be found in the Appendix Figure C.2

Based on the this, it is in the opinion of the author that a better evaluation of extractive QA models intended to work with humans should be evaluated by the SQuAD-inspired metrics presented in Section 4.2 and seen in Table 5.1, but also evaluated on the top 3 predictions. This works by the same logic as top 1, but

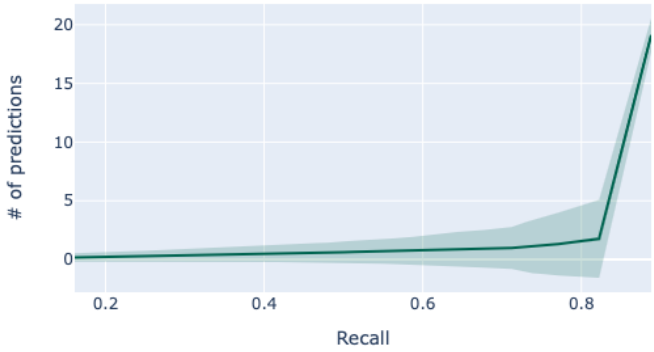


Figure 5.3. Mean number of predictions across recall levels with single σ marked, based on a **RB-S** model checkpoint on the test set using the standard confidence intervals defined in Chapter 4

instead of only using the best prediction, it uses the highest EM, F1, across the top 3. If one of them is correct, it will be a single TP; if all are incorrect, we will treat the predictions as a single FP. These metrics will provide a much clearer image of how the model performs across the border. It will also provide more insights into the expected performance in a production specialized QA system, where the user will likely be presented with three results as potential answers to a given question. top 3 was chosen as it seems a reasonable number of answers a person could be presented in a production system, but depending on the challenge, this value might vary. Additionally, top 1, top 3, and top 5 seems to be commonly used in NLP evaluations[11, 14, 51, 64]. That being said, the only way to truly determine a good value would be to run a large-scale human study presenting the solutions to lawyers and paralegals and investigating their interactions with them.

This additional evaluation metric will be included and calculated from the three highest probability predictions from the *n-best* = 20 output in the remainder of the thesis. Reusing the *n-best* = 20 instead of simply using *n-best* = 3 logic is done because on line 8 and 9 in Algorithm 1 the algorithm extracts the *n-best* start and end logits, but since some of them can be invalid due to line 10, we risk getting sub-optimal predictions, evaluated by the formula set forth in Section 4.1.2, for low values of *n-best*. Setting *n-best* = 1 with the **RB-S** model on the full test set results in AUPR$\approx$ 10; that being said *n-best* = 20 should overcome this problem and produce the best predictions as it is unlikely to have such a large set of invalid predictions. It also has the benefit of allowing to include both measures in model results.

5.4 Analyzing the baseline

5.4.1 Qualitative attention interpretation

Attention is at the transformer's core; hence interpreting the model's predictions ought to be rooted in the attention weights. This dissection of attention weights in transformer models has been the subject of many papers and the explanation it offers for model predictions is still debated [13, 26, 69, 70, 74].

Though it does not always provide explanations for model predictions, it can offer indications of the model's inner working, as has been shown by Vig and Belinkov [70] on the decoder-only GPT2 model and Clark et al. [13] on BERT.

Appendix Figure C.4 and C.5 shows the full Model View, a bird's-eye view of attention across all layers[69], between the first section of contract LIMEENERGYCO_09_09_1999-EX-10-DISTRIBUTOR AGREEMENT and the *Document Name* question seen in Appendix Section C.2. The patterns described by Clark et al. [13] are identifiable here, such as the precise attention to the next word in layer 0, head 9, the attention to the previous word in layer 0, head 11, and the aggregation of attention to the [CLS] token. This is as expected since the fine-tuning performed on the CUAD dataset is

very small compared to the pre-training; hence, it would be likely to see reoccurring patterns.

Clark et al. [13] hypothesized that high attention activation to the [SEP] token could be a sort of no-op for the model. Though there does not exist good evidence of this, comparing the SQuAD Model View in Appendix Figure C.6 and C.7 with the CUAD in Appendix Figure C.4 and C.5, it does look like there are fewer cases in SQuAD. This could indicate that we make better use of the model layers in the SQuAD trained model, which one would also expect due to the much larger dataset of 160k labeled data points with unique questions instead of 12k data points with a set of 41 different questions. These findings indicate that the CUAD fine-tuned model size could be reduced by removing layers and likely still obtain similar performance. Something that will be investigated in Section 6.3.

Inspecting the Head View, a visualization of attention for one or more attention heads in a particular layer[69], across the layers, an example of which can be found in Appendix Figure C.3, it looks like the attention between the question and context $(\mathbf{q}, \mathbf{c})$ is relatively low. Only the second part of the question, the details part described in Section 3.2, seems to have any connection with the answer span. Though the figure only shows a single layer of the model, this seems to be the case across all model layers. Looking at the same Head View with the SQuAD model, there is a much larger activation between the question and answer span. Knowing that the CUAD only contains 41 different questions, an explanation could also be that the model has memorized the questions and that it is maybe not learning properly to connect the question and context, something that will be tested in Section 5.4.3

The CUAD paper shows that pre-training increased the RoBERTa-base model's performance measured by AUPR from 42.6 to 45.2, using 8GBs of unlabelled contract data. One could expect to find a stronger connection between the legal terms using a model pre-trained for this specific case, as legal formulation and terms are not a large part of the pre-training dataset of BERT or RoBERTa. Using the LEGAL-BERT[10] obtained through `Transformers`[1] and inspecting the Head View using the same $(\mathbf{q}, \mathbf{c})$ pair did not reveal noticeable different patterns by visual inspection. Furthermore, the results in Table 5.1 have shown that the model can achieve at least the same level of performance without it.

5.4.2 Flexibility in formulations

As mentioned in Section 2.1 closed-domain QA systems can operate with a fixed set of "allowed" questions or a flexible set. In Section 5.4.1 we hypothesized that the model might have memorized the set of 41 questions, thus not making it useful for other questions than the original set or different formulations of the original questions. Investigating the model on a new type of question should be done by increasing the labeled test dataset, as removing a question from the training set would decrease the dataset's size, which has been shown to affect performance across all categories[23]

[1] **nlpaueb/bert-base-uncased-contracts**

severely. This thesis does not have the resources for that and has instead chosen to look at the adaptability to new question formulations of the existing set, as poor performance here would support a hypothesis that the model would not work well on an unseen question. Thus, three new formulations with varying degrees of "similarity" have been created for every question.

The three new formulations are depicted in Table 5.2. The full-text versions can be found in Appendix Table B.2 and B.3 or in the source code repository.

Question Set Name	Description	J(A, B)
Reversed	Reversed order of the "details" and "highlight" parts of the question	-
Altered Small (Alt-S)	Small alterations of the wording	0.48
Altered Large (Alt-L)	Larger alteration of the wording, validated to have the same meaning by a senior law student	0.685

Table 5.2. Dataset descriptions and the average Jaccard distance between the tokenized and original questions. Since the tokenization is case and order insensitive, the score is zero for the reversed question set

The model flexibility has been tested by running the test inference using the **RB-S** model checkpoint with all questions converted to new formulations. The results of this are depicted in Table 5.3.

The results clearly show that the model does generalize decently to these new question formulations with no significant degradation in performance measures in the "OWN" category. The new questions formulations do contain a high similarity to the original question posed, and hence one would also expect the reformulation to have a close distance to the original one in the embedding vector space discussed in Section 2.2.3, but even the most different one, Alt-L, obtains good scores. This indicates that the model might perform well in a specialized domain as a search engine with flexible question formulations where the user's end goal is to obtain the answer to one of the original questions. To further investigate the model's flexibility, an experiment could be to provide the question details for each question to crowdworkers and have them come up with question formulations that would obtain that piece of knowledge if they could as a question to a contract. This experiment will be left for future work, along with the analysis of the adaptation to new question types.

The Degradation in Alt-L measures by the CUAD metrics can be explained by the fact that Alt-L only contains 34 questions of the original 41 questions. This is because the author could not formulate the questions differently enough from the original formulation within the limited resources of the thesis without imposing a risk of changing the question's meaning.

Metric	Eval on	Model/Measure	Original	Reversed	Alt-S	Alt-L
CUAD	n-best	AUPR	50.39	50.53	50.59	44.83
		PREC @ 80%	37.32	36.86	-	-
		PREC @ 90%	-	-	-	-
OWN	top 1	Has Ans F1	80.73	79.74	80.52	73.12
		No Ans	81.62	82.44	83.39	83.49
		EM	77.60	77.80	78.76	77.07
		F1	81.37	81.64	82.54	80.41
	top 3	Has Ans F1	93.27	93.01	92.83	88.12
		No Ans	91.73	92.58	92.85	92.65
		EM	89.86	90.39	90.41	88.71
		F1	92.19	92.71	92.85	91.30

Table 5.3. Comparison table of performance one the test set with variety of different question formulations. Results obtained from the **RB-S** model checkpoint. - means not applicable

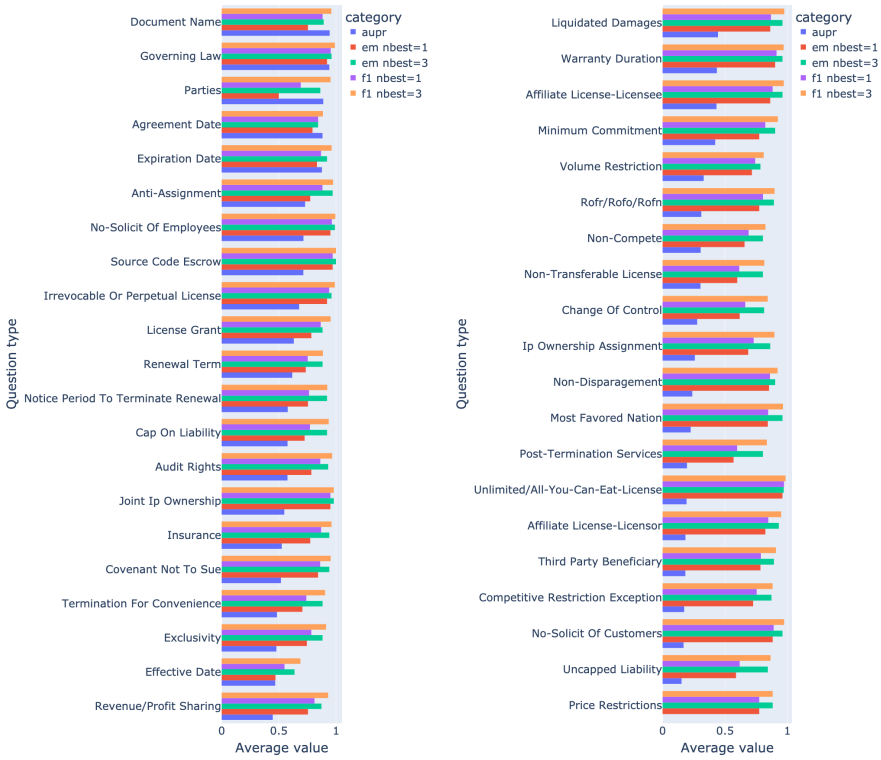
5.4.3 Performance across categories

The best performing model assessed by AUPR found by the CUAD paper is the **DeBERTa-xlarge** [23], and in Appendix Figure C.11 the papers reported results by question category are seen using this model. However, as argued in Section 5.3, the author believes that the implementation of AUPR in CUAD does not provide a complete picture of the model performance. Thus the performance across question categories using the **RB-S** model checkpoint has been investigated.

In Figure 5.4 the model performance across categories assessed by AUPR, along with top 1 and top 3 based `EM` and `F1` scores is reported using the previously discussed implementations.

Looking at the AUPR metric, the picture painted is very much the same in Figure 5.4 as in the CUAD paper, with a clear difference in performance across categories and with preamble measures at the top. However, the other measures paint a very different picture.

Take *Uncapped Liability* as an example. This question category has an AUPR score of 15.9%, whereas the *Document Name* has a score of 94.4%. This could lead one to believe that model performance is much better for *Document Name*. The reason for the low score of *Uncapped Liability* is that AUPR is calculated across $n - best = 20$ predictions as presented in Section 5.3. There are 102 examples in the test set, but only 13 of those have non-empty predictions, and with 20 predictions for each, the model can get a high FP rate easily as there can only be a single null prediction among the top 20. One could argue that the model should learn to give these incorrect predictions a low confidence score so that they will only be included close to the final confidence interval of the AUPR calculation, but the AUPR measure implemented by CUAD handles empty predictions as follows: if the ground truth is



((a)) 20 best performing questions by AUPR **((b))** The remaining 21 questions

Figure 5.4. Model performance across questions categories using the **RB-S** model checkpoint sorted by AUPR (Blue)

empty then any prediction by the model, including the empty prediction, will be a false positive [2], making the evaluation of question category very dependent on the test set $\frac{\text{impossible questions}}{\text{answerable questions}}$ ratio.

Looking at the `F1` and `EM` scores, it looks like the model is performing well across all categories. The lowest `F1` score for the Top 3 predictions is 68.9% and 54.9% for the top 1. Another interesting result is that for the *Parties* category, the model has the lowest `EM` top 1 score of 50.0% indicating that the model might be good at picking that a text section is referring to a contractual party, but not the precise location of where it starts and ends. It could also be due to unmarked answers in the dataset, as indicated in Section 3.2.1, something that will be further investigated in Section 6.1.3.

[2] `https://github.com/TheAtticusProject/cuad/blob/67faa0e6023b04fcaae6cc09497ab00e5d63a2a2/evaluate.py#L90`

5.5 Summary of findings

This chapter aimed to provide the reader with a good understanding of the baseline performance of the RoBERTa-base models on CUAD. It showed that it was possible to obtain results in a comparable range to Hendrycks et al. [23] using the same training approach implemented in a different framework.

It looked at the correlation between the objective function, the validation loss, and the different evaluation metrics using Spearman and Pearson correlation. Here it was shown that there exists a decent linear correlation between the measures, but with room for improvement as the optimal validation loss does not necessarily result in the best performance across all metrics, something that will be further discussed in Chapter 6.

The chapter discussed the relevance of AUPR as a quantifier for model performance in relevant industry tasks. It was concluded that top 1 and top 3 `EM` and `F1` provided a better picture of the quality of a given model's predictions. This was due to multiple reasons. First, metrics like Precision @ 80 % used in CUAD, tells something about the model performance on the average case, meaning that it can have scenarios where a lawyer or paralegal would need to look through many more predictions. Furthermore, it was argued that the model performance in the unanswerable question category is not of significant interest. This is because the model does not provide an argument for its decision that a human can easily verify as with the answerable prediction. It was also shown that the AUPR metric does not paint the complete picture of how well the models were performing on particular question categories, as it was very dependent on the question category distribution of answerable and unanswerable questions due to the implementation of the AUPR measure chosen in CUAD. Based on this argument, it was also shown that performance across all question categories was very similar, with good scores across the board measured by the newly proposed metrics. Furthermore, a hypothesis based on attention patterns extracted from the **RB-S** model on a CUAD `feature`, about the model not being generalizable to different formulations of the same set of 41 questions was investigated. It was shown that the model performed well on three new sets of question formulation applied to the test dataset, with a decreasing degree of similarity to the original questions measured by the average Jaccard distance between the tokenized questions. The significant finding was that the model worked well on all three formulations, but the results did not constitute a basis for concluding the model's generalizability to new questions within the legal realm. However, it did raise the question if the model would be able to adopt new question categories with a reasonably little amount of training data as question categories like *Third Party Beneficiary* obtained close to the same level of scores as the top marked answers categories with only 17 marked answer in the training dataset. This has much uncertainty associated as it similarly only has a few marked test examples.

CHAPTER 6
Enhancing CUAD

This chapter will, based on the findings in Chapter 3, 4, and 5 make alterations to the data, evaluation, and training methodology to enhance model performance measured by the SQuAD inspired top 1 and top 3 metrics and the general usability of the model in the industry. This will include rewriting the dataset creation logic, experiments with programmatically adding annotations to the datasets, altering that balancing between impossible and answerable questions on a question category level, and a complexity and optimization review of the RoBERTa-base model.

6.1 Dataset manipulations

6.1.1 Consistent dataset

At the time of writing, the latest release of the `Transformers` library was v4.19.2. This release, along with the one used in the thesis v4.18.0, has the same code for converting the raw data in a SQuAD format to `examples`, `features`, and a `dataset` as seen in Section 3.2.3. The code creates an `example` for each $(\mathbf{q}, \mathbf{C})$ pair, but this `example` only contains the first answer in the list of correct spans[1]. To mitigate this, CUAD uses a dataset where each question with multiple answers is split into separate identical questions with different answer markings, thereby creating an `example` for each answer associated with a $(\mathbf{q}, \mathbf{C})$ pair.

This approach faces other issues, as the model can be exposed to the same data point multiple times with different answer markings, thereby introducing ambiguity. Ignoring the balancing of the dataset that removes features randomly, the number of times the "correct" answer span is marked in a feature, in the multiple answer case, is $\leq$ the number where it is not. The extent of this problem is dependent on the question category, as some question categories have more answer-markings than others which is evident from Figure 6.1.

[1] `https://github.com/huggingface/transformers/blob/v4.19-release/src/transformers/data/processors/squad.py#668`

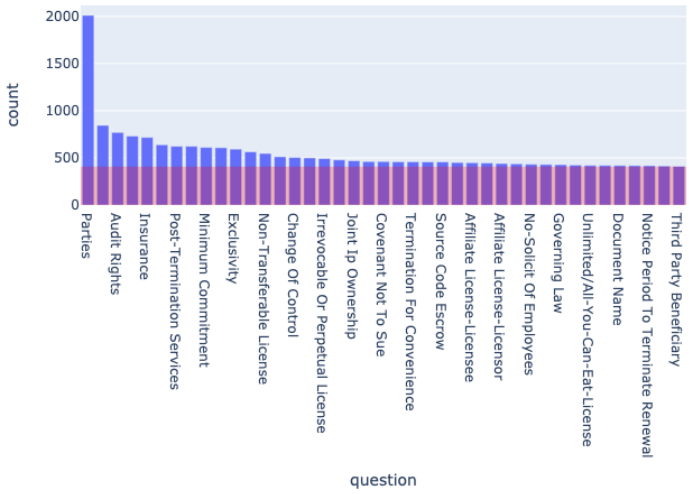
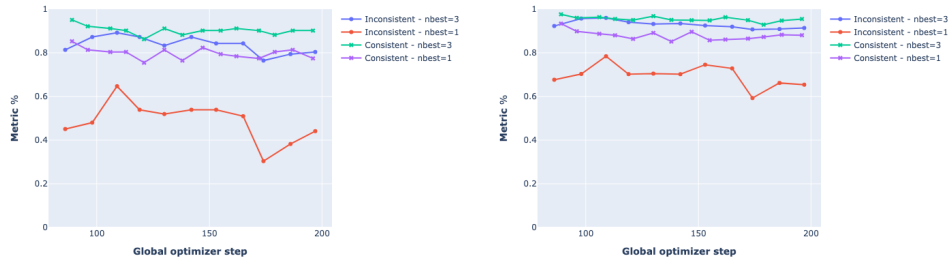


Figure 6.1. Distribution of the number of `examples` pr. question category. The purple zone indicates the number of contracts in the training set

To investigate the effect of this on the model performance, the pre-processing script from `Transformers`, has been rewritten to allow for `examples` to have multiple answers. This means that each question type now has the same number of `examples` and when it gets converted into `features`, multiple of them can contain answers for a given $(\mathbf{q}, \mathbf{C})$ pair.

Since the loss implemented is a cross-entropy loss, described in Section 4.1.3, that does not allow for multi-label predictions, the implementation of the new `feature` creation scripts is such that each `feature` only gets the first occurring answer, ordered by the answer start position if multiple are available for a given span. The question category most affected by this is *Parties*, as this is the category with the most answers, as seen in Figure 6.1. The new dataset creation methodology is termed `Consistent`, as it removes the probability of ambiguity in the `features` and the old methodology is termed `Inconsistent` for the opposite reason. The results from training two models using the same **RB-S** setup, with the two different dataset creation strategies, produce the results in Figure 6.2. The figure shows the models `EM` and `F1` scores of the *Parties* question, and it is clear that the top 1 prediction in the `Consistent` dataset outperforms the top 1 prediction in the `Inconsistent` dataset, which likely is due to the removal of ambiguity in the correct answer for a given span. Looking at the categories of *Audit Rights*, *Insurance*, and *Post-Termination Services*, the three question categories with the most answer markings after *Parties*, there does not seem to be a clear difference. The same is the case for overall model performance measures, as they do not indicate that one strategy outperforms the other, which will be covered in Section 6.4.



((a)) `EM` ((b)) `F1`

Figure 6.2. Model performance on the *Parties* question category measures by `F1` and `EM` metric on the `Consistent` and `Inconsistent` datasets trained using the **RB-S** hyperparameters

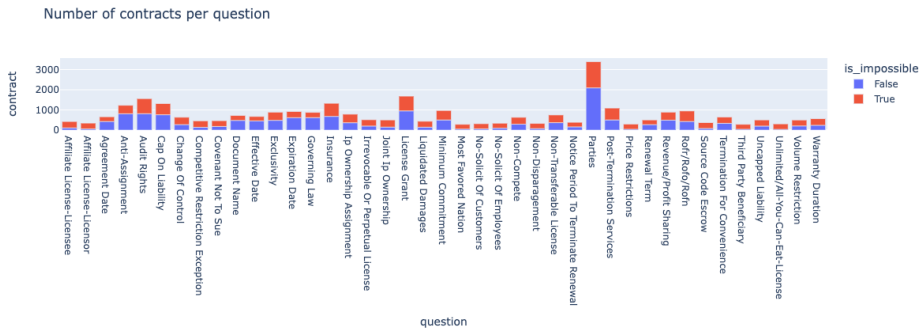
6.1.2 Truly balanced dataset

In Section 3.2.3, it was mentioned that a method is implemented to balance the empty and labeled predictions equally. This method achieves the goal by randomly removing empty data points from the dataset until there is a class balance without ensuring stability within the individual question categories. The issue is that for some seeds, the ratio between classes can be quite off from the desired $\frac{1}{1}$. In Figure 6.3(a) the distribution of these marked and impossible answers in the training dataset using the CUAD random seed.

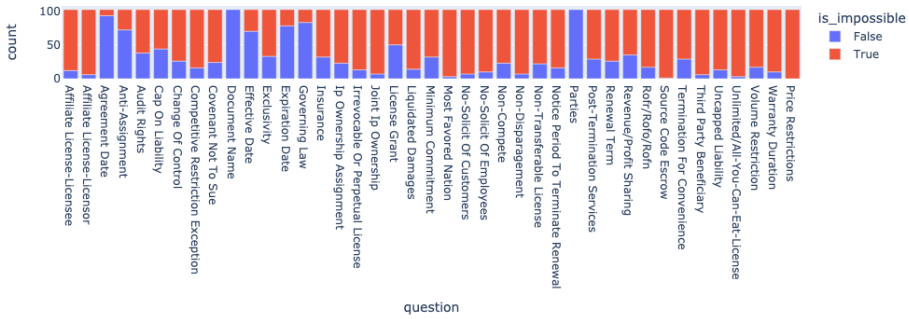
For the question category *Affiliate License-Licensor* the imbalance is upwards of 5x with 285 impossible questions and 62 answerable questions. This plot also shows why it is hard to conclude that some question categories are more difficult than others to predict, as there is a significant difference in the number of data samples and the ratio of class balance varies a lot.

To investigate the effect of this on performance, a new strategy that randomly balances all question categories was implemented. This strategy includes a hyperparameter δ for configuring the ratio. The results presented below is based on running the training using the `Consistent` dataset and using $\delta \in \{0.5, 1, 1.5, 2, 3\}$ parameter for the balancing producing training set sizes of $\{16.258, 21.694, 27.105, 32.541, 43.388\}$ respectively. This wide range of balancing is chosen due to the wide range of question ratios in the test set, which is not altered. In Figure 6.3(b) we see the distribution of the impossible and answerable questions in the test set, and there are roughly 2.3 times as many impossible questions as there are answerable.

In Figure 6.4 it is noticeable that *Most Favored Nation* seem to to benefit from a higher class imbalance towards impossible questions than *Parties*. This is expected since the test dataset for the *Most Favored Nation* has 99 impossible and 3 answerable questions, whereas *Parties* has 0 impossible questions and 102 answerable. Many



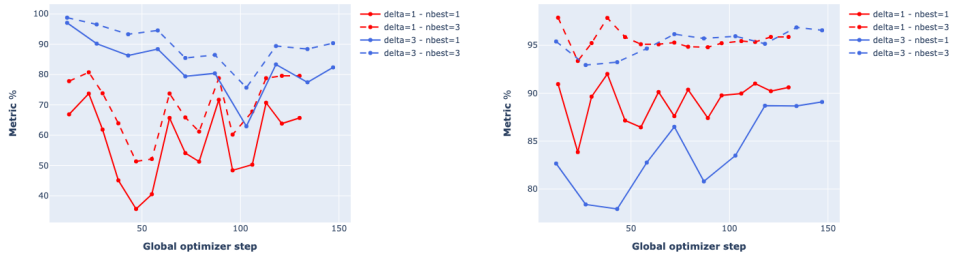
((a)) Distribution of marked and impossible answers in training dataset using the CUAD random seed



((b)) Distribution of impossible (2938) and answerable(1244) questions in the test set.

Figure 6.3. Distributions of impossible and answerable questions

other categories are inconclusive in performance change as a function of δ. As mentioned in Section 5.2, the variation in performance at different global optimizer steps seems more random than one approach working better than another. Looking at the `Top 3 Has Ans F1` measures for the run using $\delta = 1$, it achieves a slightly better score 94.06 at peak performance, but it does not outperform the $\delta = 3$ trained model at every checkpoint. As the results vary quite greatly across question categories, an aggregate results table has not been provided, as it does not provide a good representation of model performance across categories. However, the link to the WandB project has been provided in Appendix Section A.1 to allow for further analysis by the reader.



((a)) *Most Favored Nation* **((b))** *Parties*

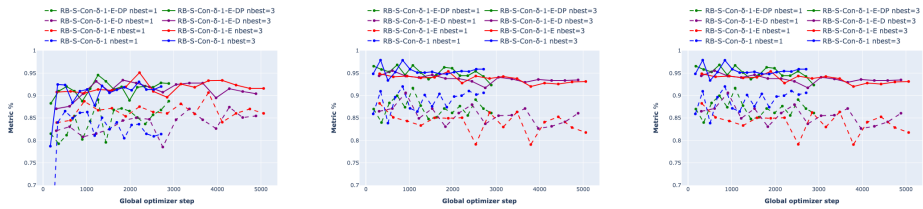
Figure 6.4. RB-S model performance in the *Parties* and the *Most Favored Nation* questions categories trained on the `Consistent` dataset with varying δ values

6.1.3 Adding answers

In Section 3.2.1, Figure 3.1, a presentation of the distribution of marked answers across categories was presented, along with potentially unmarked answers, defined as exact and fuzzy matches of the answer texts. These unmarked answers have only been probed for validity, but an experiment was done with the inclusion of the exact match answers during training to understand the effect (not)including them has on performance. As in Section 6.1.2 the impact this procedure has on question categories is varied. Three experiments were done using the `Consistent`, balanced dataset with $\delta = 1$ and the standard CUAD hyperparameters described with **RB-S**. Model one **RB-S-Con-δ-1-E** has all the exact matched presented in Figure 3.1 added. Model two, named **RB-S-Con-δ-1-E-D** has the additional answers added, excluding the ones related to dates as it is likely there are multiple dates in the contract where the answer cannot be inferred from the context. Model three, named **RB-S-Con-δ-1-E-DP**, is excluding extra answers related to both *Parties* and dates, with *Parties* being removed for the reasons presented in Section 3.2.1. Some questions also include indirect references to the company name, posing the potential issue that the model could learn to predict a specific set of pronouns or nouns referring to a contracting party.

In Figure 6.5 all three models are compared to the baseline model: **RB-S-Con-δ-1**. The categories initially chosen for analysis was *Parties, Document Name, Anti-Assignment, Agreement Date*, but during an investigation of the added answers, it was discovered that question "NEOMEDIATECHNOLOGIESINC... ...Anti-Assignment_1" had "." marked as an answer, meaning that the majority of additional 205 answers in the question category to the original 517 answers were useless. This was discovered after running the model inference for the three new datasets, and an interesting fact is that it did not seem to significantly worsen performance, as seen in Appendix Figure C.9. This finding raises the question of the general quality of the markings in other

categories, but revisiting Figure 3.1 no other question category seems to have signs of similar issues.



((a)) *Agreement Date* ((b)) *Parties* ((c)) *Document Name*

Figure 6.5. Performance across the question categories from models trained on three different datasets along with a baseline constructed by adding the exact match answers, described in Section 3.2.1, to the `Consistent` dataset with $\delta = 1$

In Figure 6.5(a) there does not seem to be a difference between the model performance across models. All seem to follow the same pattern, and looking at the larger perspective agreement date has an increased from 383 marked answers to 826, which might not be enough to improve the model understanding with an original total of 11180 marked answers. For *Parties* and *Document Name* the performance seems to be slightly worsened in **RB-S-Con-δ-1-E** and **RB-S-Con-δ-1-E-D** with their nbest=3 curves lower than the ones of **RB-S-Con-δ-1-E-DP** and the baseline **RB-S-Con-δ-1**. That being said, the variation in performance found on both question category and model level falls within the expected randomness range mentioned earlier, which will be further discussed in Section 6.4.

6.2 Transfer learning from SQuAD

The balanced CUAD dataset presented in Section 6.1.1 is approximately $\frac{150000}{20000} = 7.5$ times smaller than SQuAD, and in many NLP tasks, the amount of training data has a large impact on performance[11, 23, 51, 52]. The authors of CUAD also showed that this was the case for the RoBERTa-base model in their paper evaluated by the AUPR measure [23].

The market price for CUAD is estimated at $ 2.000.000 by the authors [23], making it a costly affair to increase the dataset size. As mentioned throughout this thesis, the CUAD modeling is very similar to the one used in SQuAD, and thus it's relevant to investigate whether utilizing the SQuAD dataset in combination with CUAD during training can improve model performance on CUAD.

To explore this, a RoBERTa-base model that has been fine-tuned on SQuAD was used as a starting point for training a balanced **RB-S-Con-δ-** model using $\delta \in \{1, 1.5\}$. These two runs are compared to the $\delta \in \{1, 1.5\}$ runs from Section 6.1.2. The results are reported in Table 6.1 from the model checkpoint that achieves the

best `Top 3 Has Ans F1`. The results indicate no difference in the performance when comparing the models using a SQuAD fine-tuned model[2] with a normal RoBERTa-base pre-trained model. The only noticeable difference is that the training loss starts much lower as seen in Appendix Figure C.10. Presumably this is due to the final fully connected layer, QA_h, not being randomly initialized for the SQuAD fine-tuned model, but that it instead can utilize previously obtained knowledge. These models also face the issue of models reaching a decent performance level very quickly, usually within 1-2 epochs, but then not improving much further, as seen in Appendix Figure C.8, and as discussed in Section 5.2 and 6.1.2. The results also show much variation in the scores across checkpoints for these models, and even though we pick the checkpoint with the maximum `Top 3 Has Ans F1`, the other reported scores might not be at their max.

Eval on	Model/Measure	SQuAD-δ=1.5	δ=1.5	SQuAD-δ=1	δ=1
top 1	Has Ans F1	84.18	85.28	85.69	85.47
	No Ans	69.30	69.71	53.64	54.70
	EM	70.70	71.28	60.19	60.33
	F1	73.73	74.34	63.18	63.81
top 3	Has Ans F1	93.53	93.63	93.88	94.06
	No Ans	84.28	81.11	67.50	66.34
	EM	84.79	82.59	73.41	72.17
	F1	87.03	84.83	75.34	74.58

Table 6.1. Model performance of the maximum `Top 3 Has Ans F1` checkpoint for SQuAD fine-tuned models compared against the baseline from Section 6.1.2

6.3 Model size

As outlined in the Introduction (Section 1) and the Research Question and Objective (Section 1.2), extractive QA within the legal domain is something both large law firms and startups are allocating vast resources to[65]. Since this thesis has a perspective of how the models presented could be used in a production environment and the considerations it requires, a relevant area to cover is the model size. Though Chalkidis et al. [11] have shown that the RoBERTa-large model leads to substantial performance improvement over the RoBERTa-base model on LexGLUE[11], a proposal for a standardized evaluation framework for legal domain deep learning models based on open-source data, it also comes with drawbacks. The size of RoBERTa-large (355M) is more than 2× the size of RoBERTa-base (110M)[14, 37] which leads to worse space and time complexity parameters for inference. Running inference on the SQuAD example from Appendix B.1 1000 times on a single Nvidia T4 GPU yields $\mu = 0.0197s, \sigma = 0.0044$

[2]Huggingface model: deepset/roberta-base-squad2

for the RoBERTa-base model and $\mu = 0.306s, \sigma = 0.0012$ for RoBERTa-large. Furthermore, the binary file containing the model state RoBERTa-base has a size of 501MB whereas the RoBERTa-large requires 1430MB using `Transformers`[3].

As compute expenses generally increase with larger space and time complexities[38], reducing model sizes is both interesting from a business and a machine learning perspective, as it can provide potential savings on computing resources and reveal underutilization of models and deeper insights into why it is performing as it is. This is illustrated in Figure 6.6, with the model inference time for the SQuAD example illustrated as a function of layers in the RoBERTa-base model.

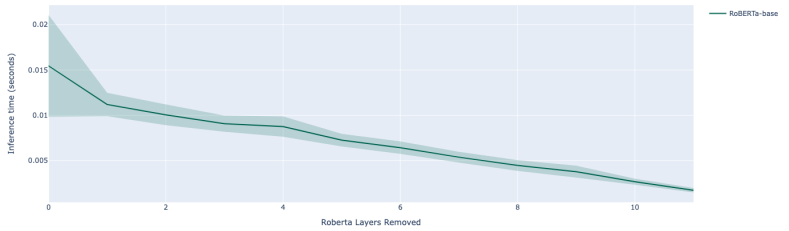


Figure 6.6. RoBERTa-base inference time as a function of model layers removed on the SQuAD example from Appendix C.2 with a single σ highlighted. Each data point is based on 300 runs on an Nvidia Tesla T4 GPU.

Investigating the significance of all layers in terms of the influence on model performance and their combinations requires $2^L = 2^{12} = 4096$ experiments, each one taking around 32 minutes to complete using the 4 Nvidia T4 GPU setup described in 3.3; thus only a subset has been analyzed. In Section 5.4.1 it was shown that there seems to be a strong connection between the question and correct answer span in layers 5-6 for in the **RB-S** model. It was also noted that the model seems to contain multiple layers with high attention activations to the [SEP] token, which has by Clark et al. [13] been hypothesized as a no-op for the model.

Based on this, and experiments on a smaller randomly selected test dataset, the model performance was tested with the $\delta = 1$ model from Section 6.2 using the minimum validation loss checkpoint obtained at global step 1864 with `Top 3 Has Ans F1`= 92.95 and `Top 1 Has Ans F1`= 84.04. This model then ran inference on the test set with $L \notin \{\{9\}, \{10\}, \{11\}, \{5\}, \{6\}, \{2\}, \{1\}, \{3\}, \{3, 6\}, \{9, 10\}, \{11, 3\}, \{2, 4, 6\}, \{5, 2, 4\}, \{1, 2\}, \{5, 9\}, \{8, 9\}, \{10, 9\}\}$ and based on those run $L \notin \{\{\}, \{9, 10, 1\}, \{9, 1, 3\}, \{10, 1, 3\}, \{10, 1\}, \{1, 3\}, \{10, 9, 4, 11\}, \{10, 9, 3\}\}$. A selected subset of results of running model inference with these layers removed is listed in Table 6.2.

Each removed layer reduces the model size by around 6%. The best-performing model with three removed layers, specifically $\{1, 9, 10\} \notin L$, was re-trained using

[3]Huggingface models: roberta-base & roberta-large

the **RB-S** hyperparameters. From this re-trained model, it is evident that model size can be decreased upwards of 20% without sacrificing much performance, as it potentially can be recouped by re-training the model using the original parameters. The results also show that different layers and layer combinations have very different effects on performance if removed. The model with $\{2, 4, 5\} \notin L$ has the same model size reduction as $\{1, 9, 10\} \notin L$ but a much lower performance score.

The approach to reducing model size presented here is simple, only removing entire layers but using the argument of no-op predictions as the basis of the size reductions. It would be more precise to remove individual attention heads instead of entire layers, but this requires more extensive model alterations as it alters the shape of the concatenated attention matrix described in Section 2.2.2. Michel et al. [41] showed that "a large percentage of attention heads can be removed at test time without significantly," making this a good candidate for model reductions. Other simple approaches could be by utilizing *quantization*, using a smaller precision of network weights, or *weights pruning*, simplifying some network connections as done by Suík [62] on BERT. A more advanced approach to reducing the model size could be distilling the model as done by Sanh et al. [59] with DistilBERT. DistilBERT is significantly smaller than BERT, reducing the number of parameters by 40% and inference time by 60% while preserving over 95% of the BERT performance on the language understanding challenge GLUE[59, 71]. This approach termed *knowledge distillation*[7, 25] or teacher-student learning, "is a compression technique in which a compact model - the student - is trained to reproduce the behaviour of a larger model - the teacher - or an ensemble of models"[59].

HA F1	NA F1	EM	F1	HA F1	NA F1	EM	F1	$\notin L$
Top 3				Top 1				-
93.27	**78.82**	**80.56**	**82.84**	**83.30**	**61.13**	**66.44**	**67.72**	**1, 9, 10**
92.97	85.60	85.27	87.79	83.83	76.96	75.47	79.00	9
92.94	83.56	84.00	86.35	84.04	73.49	73.24	76.62	
91.27	84.96	84.29	86.83	82.35	72.97	72.12	75.76	1, 10
90.12	86.93	85.13	87.88	80.23	74.06	72.24	75.90	5, 9
89.88	74.95	76.18	79.39	80.66	65.11	64.75	69.74	1, 9, 10
87.20	74.51	75.82	78.28	76.55	54.66	58.06	61.17	6
80.66	94.55	86.49	90.42	60.30	86.49	74.84	78.70	1, 3
75.04	97.79	86.90	91.02	52.86	93.09	78.43	81.12	1, 3, 10
69.19	8.20	13.39	26.34	54.49	3.95	7.60	18.98	4, 9, 10, 11
65.98	89.79	78.07	82.71	44.83	78.69	65.21	68.62	1, 2
38.94	20.18	19.99	25.76	27.18	16.13	14.97	19.42	2, 4, 5

Table 6.2. Subset of Appendix Table C.1, showing model performance for different configuration of deleted layers from the $\delta = 1$ model presented in Section 6.2 using the checkpoint from `global_step` = 1864. HA = Has Ans and NA = No Ans. The **bold** marking is the model with deleted layers retrained using the same parameters as the model checkpoint

6.4 Summary of findings

The goal of this chapter was to enhance the usability of CUAD by attempting to increase performance measured by the SQuAD top 1 and top 3 scores in the Has Ans category and by presenting optimizations such as decreasing model size, which can be of much value in production systems[59].

One way this was attempted was by creating a more consistent dataset, as it was discovered that some question categories had multiple identical data points but with different answer markings. Another alteration was balancing the ratio between impossible and answerable questions on a question level instead of on the dataset level and experimenting with different ratios between these two question types. Trails with utilizing models fine-tuned on SQuAD as a starting point revealed little to no difference in performance other than a faster convergence, which most likely is due to the final fully connected layer not being initialized randomly, but from a state similar to the challenge at hand. Experiments with adding the found unmarked answers from Section 3.2.1 were also done but did not increase model performance.

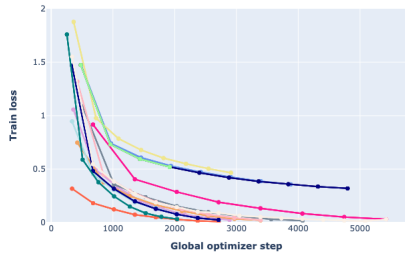
Though all these experiments improved the quality of the dataset, which constitutes an integral part of well-functioning machine learning systems [19, 20], little to no effect on the performance measures was found for all experiments on a dataset level. In Figure 6.7 one sees the training loss, validation loss, and model performance for all models discussed and tested in this chapter. Every single model seems to be facing the same issue of quickly reaching a decent level of performance, but the optimization that yields better scores appears to be more random than one the training approach embedding more helpful knowledge into the model than another after epoch 2. Thus it is in the author's opinion that it cannot be concluded that the performance increase from 92.5% to 94.1% in the `top 3 Has Ans` metric obtained by the best performing model compared to the CUAD RoBERTa-base model checkpoint is due to a better training approach. The authors' central idea for mitigating this issue is to increase the correlation between the objective function and the actual metrics, and a hypothesis for how this could be done is by an alteration of the loss function. This is left for future work, but there seems in general to be room for improvement in the loss function. The model is evaluated on a cross-entropy loss, but that does not encapsulate the way the model is used. In the evaluation metrics, it accepts all predictions with `F1` > 0.5 as correct predictions, thus, changing the loss function to reflect this through approaches like *label smoothing*[44, 63, 68] might be beneficial.

This chapter also showed that the current model performance could be obtained from a RoBERTa-base model reduced upwards of 20% in size and with a mean processing time close to 50% better, with little to no performance loss measured by `Top 3 Has Ans F1`.

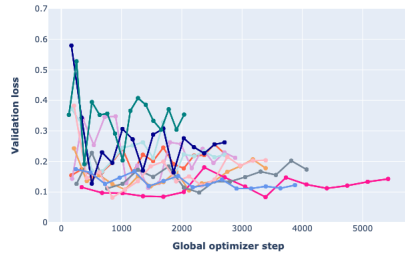
Though it was not shown in this chapter, it was noticed that the methods for extracting model predictions vary across implementations[4].

[4]QA pipeline approach to extracting model predictions `https://github.com/huggingface/transformers/blob/main/src/transformers/pipelines/question_answering.py#L377`

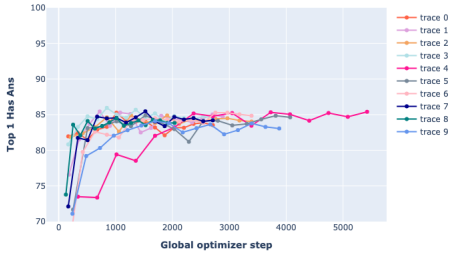
In Section 4.1.2 the approach from CUAD originating from BERT for extracting predictions was presented[14, 23]. This is $P(S \text{ and } E) = P(S)P(E)$ where S and E are the start and end logits using the notion from Equation 4.1. Recall that the implementation uses $S(T_i) + E(T_i)$ where $i \leq j$, which is the sum of the output logits from the start and end positions. The output probabilities is a relative ranking of the $nbest = 20$ predictions, as the final step for outputting the predictions is performing a softmax across the prediction scores. Other implementations perform the softmax on the output probabilities for each `feature` ignoring the P-mask and [CLS] token and then calculating the model confidence directly. Though these two approaches, in theory, should produce approximately the same results, their implementation was so different that an experiment was done to compare the two strategies. It did not reveal any major difference in performance. The new approach also evaluates all predictions before extracting the $n-best$ scores, a pain-point about the original method discussed in Section 5.3. This did not influence performance either.



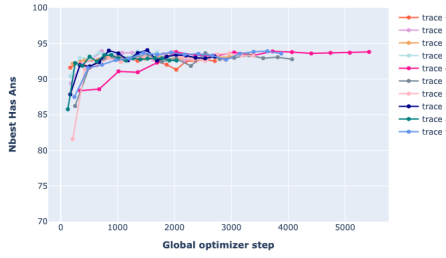
((a)) Training loss



((b)) Validation Loss



((c)) Top 1 Has Ans



((d)) Top 3 Has Ans

Figure 6.7. Training loss, validation loss, top 1 has ans, and top 3 has ans across all 18 runs presented as part of this chapter.

CHAPTER 7

Conclusion

This thesis presented a deep dive into span-selection question answering models based on the RoBERTa transformer architecture for legal contract review and extraction of contract elements, along with a thorough analysis of the Contract Understanding Atticus Dataset (CUAD), to enhance the value of the dataset and models trained on it in the perspective of using the models in a human-in-the-loop production setting.

Based on the work of Hendrycks et al. [23], a new training and validation framework for the models was built using PyTorch Lightning for a single node multi GPU environment. This allowed for efficient and repeatable experiments and extensibility to adapt to new transformer architectures or change the data manipulations by altering a single command in the training CLI. Using this framework, it was shown that performance from the CUAD could be reproduced and improved measured by AUPR, though the paper argued that the top 1 and top 3 `Has Ans F1` and `Has Ans EM` measures provided better insights into the model performance for the task at hand.

The thesis analysis of the data quality of the CUAD, shed light on elements that, to the knowledge of the author, have not been discussed in academic research yet, due to the recent publication of the dataset [23] and the small community covering the topic, as seen in Figure 1.1. This analysis covered potentially unmarked answers, the quality of the unanswerable questions, and the distribution of model annotation across both test and training datasets. Furthermore, it was shown that the dataset creation process could introduce ambiguity in the training data due to text spans appearing with both marked and unmarked answers. Additionally, potential unintended consequences of the logic implemented for the class balancing were shown, and a solution for handling it was discussed, implemented, and tested.

Based on these findings, it was attempted to enhance the model performance. Some question categories' performance benefitted from the alterations, but the overall model metrics did not significantly improve. It was shown that this was the case across almost all experiments presented in the thesis. The model would reach a decent performance within 1-2 epochs, and then the movements in the evaluation metrics from additional training would act what appears to be random. It was hypothesized that this was due to a correlation between the performance measures and objective function that could be improved, supported by an analysis of Spearman and Pearson correlation. Together, these findings indicate that the optimization space for the performance measures does not align well with the objective function's space in the finer training stages. This was also the case for training trails started from models

previously fine-tuned on other span-selection QA datasets like SQuAD.

Additionally, it was shown that model performance on most question categories was much better than indicated by the AUPR measure if evaluated by the proposed top 1 and top 3 `F1` and `EM` measures. Furthermore, model size was reduced by upwards of 20% in terms of parameter count and 50% in inference time with little to no performance loss.

The thesis also presented proposals for experiments to improve the production value of the models further, with the primary one being an alteration of the loss from cross-entropy with a single correct answer to a more smooth quantification of a correct prediction inspired by the `F1` measure and *label smoothing*. Another proposal was related to the reduction of the model size using *quantization*, *weights pruning*, or by distilling the model as done by Sanh et al. [59] with DistilBERT. Additionally, as discussed by Chalkidis et al. [11], the use of transformer models designed for handling longer inputs sequences than the standard RoBERTa model, such as the Longformer, was also suggested.

Appendices

APPENDIX A

Definitions & Additional resources

A.1 Interactive plots and source code

The project source code can be found on the authors Github: `https://github.com/gustavhartz/transformers-legal-tasks`

The Weights and Bias project for the training of the different model described in Chapter 6 can be found here: `https://wandb.ai/gustavhartz/cuad_training_balanced_features?workspace=user-gustavhartz`

The Weights and Bias project for the model size experiments described in Chapter 6 can be found here: `https://wandb.ai/gustavhartz/cuad_test_model_size?workspace=user-gustavhartz`

The best performing model(s) presented throughout the thesis will be made available through the authors Huggingface profile `https://huggingface.co/gustavhartz`

A.2 Transformers & Tokenization

Tokenizer notation

Notation	Description
[PAD]	The BERT token used to increase the size of sentences shorter than the expected input size. PAD is short for padding
[CLS]	The BERT token used for classification which is also where it gets it's name from
[SEP]	The BERT token used for separating sentence A and Sentence B in the next sentence prediction
[UNK]	The BERT token used to represent an unknown word which does not exists in the model vocabulary
[EOS]	Token used for marking the end of a sequence
[BOS]	Token used to mark the beginning of a sequence

Table A.1. Common tokenizer special tokens notation

Huggingface Transformers Example

```
from transformers import AutoTokenizer, AutoModelForQuestionAnswering
import torch
model_path = "huggingface-course/bert-finetuned-squad"

tokenizer = AutoTokenizer.from_pretrained(model_path)
model = AutoModelForQuestionAnswering.from_pretrained(model_path)

text = """Some context"""
question = """Some questions"""
inputs = tokenizer.encode_plus(question, text,\textbackslash
add_special_tokens=True, return_tensors="pt")
input_ids = inputs["input_ids"].tolist()[0]

text_tokens = tokenizer.convert_ids_to_tokens(input_ids)
answer_start_scores, answer_end_scores = model(**inputs)[0], model(**inputs)[1
# Highest probability answer start
answer_start = torch.argmax(answer_start_scores)

# Highest probability answer end
answer_end = torch.argmax(answer_end_scores) + 1

answer_tokens = tokenizer.convert_ids_to_tokens(input_ids[answer_start:answer_
answer = tokenizer.convert_tokens_to_string(answer_tokens)
print(f"Question: {question}")
print(f"Answer: {answer}\n")
```

Figure A.1. Huggingface `Transformers` span-selection QA example

APPENDIX B

Data

B.1 SQuAD text example

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.[52]

B.2 CUAD Questions

Table B.1. Categories and questions from CUAD. The capital letter in parentheses indicates the segmentation category the question belongs to for the answer occurrence analysis in Section 3.2.2

Category	Question
Document Name (R)	Highlight the parts (if any) of this contract related to "Document Name" that should be reviewed by a lawyer. Details: The name of the contract
Parties (R)	Highlight the parts (if any) of this contract related to "Parties" that should be reviewed by a lawyer. Details: The two or more parties who signed the contract
Agreement Date (R)	Highlight the parts (if any) of this contract related to "Agreement Date" that should be reviewed by a lawyer. Details: The date of the contract

Continued on next page

Table B.1 – Continued from previous page

Category	Question
Effective Date (R)	Highlight the parts (if any) of this contract related to "Effective Date" that should be reviewed by a lawyer. Details: The date when the contract is effective
Expiration Date (R)	Highlight the parts (if any) of this contract related to "Expiration Date" that should be reviewed by a lawyer. Details: On what date will the contract's initial term expire?
Renewal Term (R)	Highlight the parts (if any) of this contract related to "Renewal Term" that should be reviewed by a lawyer. Details: What is the renewal term after the initial term expires? This includes automatic extensions and unilateral extensions with prior notice.
Governing Law (R)	Highlight the parts (if any) of this contract related to "Governing Law" that should be reviewed by a lawyer. Details: Which state/country's law governs the interpretation of the contract?
Exclusivity (R)	Highlight the parts (if any) of this contract related to "Exclusivity" that should be reviewed by a lawyer. Details: Is there an exclusive dealingň commitment with the counterparty? This includes a commitment to procure all ÄúrequirementsÄù from one party of certain technology, goods, or services or a prohibition on licensing or selling technology, goods or services to third parties, or a prohibition onň collaborating or working with other parties), whether during the contract orň after the contract ends (or both).
No-Solicit Of Customers (R)	Highlight the parts (if any) of this contract related to "No-Solicit Of Customers" that should be reviewed by a lawyer. Details: Is a party restricted from contracting or soliciting customers or partners of the counterparty, whether during the contract or after the contract ends (or both)?
No-Solicit Of Employees (R)	Highlight the parts (if any) of this contract related to "No-Solicit Of Employees" that should be reviewed by a lawyer. Details: Is there a restriction on a partyÄôs soliciting or hiring employees and/or contractors from theň counterparty, whether during the contract or after the contract ends (or both)?

Continued on next page

Table B.1 – Continued from previous page

Category	Question
Rofr/Rofo/Rofn (R)	Highlight the parts (if any) of this contract related to "Rofr/Rofo/Rofn" that should be reviewed by a lawyer. Details: Is there a clause granting one party a right of first refusal, right of first offer or right of first negotiation to purchase, license, market, or distribute equity interest, technology, assets, products or services?
Anti-Assignment (R)	Highlight the parts (if any) of this contract related to "Anti-Assignment" that should be reviewed by a lawyer. Details: Is consent or notice required of a party if the contract is assigned to a third party?
Price Restrictions (R)	Highlight the parts (if any) of this contract related to "Price Restrictions" that should be reviewed by a lawyer. Details: Is there a restriction on theň ability of a party to raise or reduce prices of technology, goods, orň services provided?
Minimum Commitment (R)	Highlight the parts (if any) of this contract related to "Minimum Commitment" that should be reviewed by a lawyer. Details: Is there a minimum order size or minimum amount or units per-time period that one party must buy from the counterparty under the contract?
License Grant (R)	Highlight the parts (if any) of this contract related to "License Grant" that should be reviewed by a lawyer. Details: Does the contract contain a license granted by one party to its counterparty?
Post-Termination Services (R)	Highlight the parts (if any) of this contract related to "Post-Termination Services" that should be reviewed by a lawyer. Details: Is a party subject to obligations after the termination or expiration of a contract, including any post-termination transition, payment, transfer of IP, wind-down, last-buy, or similar commitments?
Warranty Duration (R)	Highlight the parts (if any) of this contract related to "Warranty Duration" that should be reviewed by a lawyer. Details: What is the duration of anyň warranty against defects or errors in technology, products, or servicesň provided under the contract?

Continued on next page

Table B.1 – Continued from previous page

Category	Question
Insurance (R)	Highlight the parts (if any) of this contract related to "Insurance" that should be reviewed by a lawyer. Details: Is there a requirement for insurance that must be maintained by one party for the benefit of the counterparty?
Covenant Not To Sue (R)	Highlight the parts (if any) of this contract related to "Covenant Not To Sue" that should be reviewed by a lawyer. Details: Is a party restricted from contesting the validity of the counterpartyÄôs ownership of intellectual property or otherwise bringing a claim against the counterparty for matters unrelated to the contract?
Change Of Control (R)	Highlight the parts (if any) of this contract related to "Change Of Control" that should be reviewed by a lawyer. Details: Does one party have the right to terminate or is consent or notice required of the counterparty if such party undergoes a change of control, such as a merger, stock sale, transfer of all or substantially all of its assets or business, or assignment by operation of law?
Audit Rights (R)	Highlight the parts (if any) of this contract related to "Audit Rights" that should be reviewed by a lawyer. Details: Does a party have the right toň audit the books, records, or physical locations of the counterparty to ensure compliance with the contract?
Uncapped Liability (R)	Highlight the parts (if any) of this contract related to "Uncapped Liability" that should be reviewed by a lawyer. Details: Is a partyÄôs liability uncapped upon the breach of its obligation in the contract? This also includes uncap liability for a particular type of breach such as IP infringement or breach of confidentiality obligation.
Cap On Liability (R)	Highlight the parts (if any) of this contract related to "Cap On Liability" that should be reviewed by a lawyer. Details: Does the contract include a cap on liability upon the breach of a partyÄôs obligation? This includes time limitation for the counterparty to bring claims or maximum amount for recovery.

Continued on next page

Table B.1 – Continued from previous page

Category	Question
Non-Compete (R)	Highlight the parts (if any) of this contract related to "Non-Compete" that should be reviewed by a lawyer. Details: Is there a restriction on the ability of a party to compete with the counterparty or operate in a certain geography or business or technology sector?
Competitive Restriction Exception (R)	Highlight the parts (if any) of this contract related to "Competitive Restriction Exception" that should be reviewed by a lawyer. Details: This category includes the exceptions or carveouts to Non-Compete, Exclusivity and No-Solicit of Customers above.
Volume Restriction (R)	Highlight the parts (if any) of this contract related to "Volume Restriction" that should be reviewed by a lawyer. Details: Is there a fee increase or consent requirement, etc. if one partyÄôs use of the product/services exceeds certain threshold?
Termination For Convenience (R)	Highlight the parts (if any) of this contract related to "Termination For Convenience" that should be reviewed by a lawyer. Details: Can a party terminate thisň contract without cause (solely by giving a notice and allowing a waitingň period to expire)?
Ip Ownership Assignment (R)	Highlight the parts (if any) of this contract related to "Ip Ownership Assignment" that should be reviewed by a lawyer. Details: Does intellectual property createdň by one party become the property of the counterparty, either per the terms of the contract or upon the occurrence of certain events?
Irrevocable Or Perpetual License (R)	Highlight the parts (if any) of this contract related to "Irrevocable Or Perpetual License" that should be reviewed by a lawyer. Details: Does the contract contain aň license grant that is irrevocable or perpetual?
Notice Period To Terminate Renewal (R)	Highlight the parts (if any) of this contract related to "Notice Period To Terminate Renewal" that should be reviewed by a lawyer. Details: What is the notice period required to terminate renewal?

Continued on next page

Table B.1 – Continued from previous page

Category	Question
Liquidated Damages (R)	Highlight the parts (if any) of this contract related to "Liquidated Damages" that should be reviewed by a lawyer. Details: Does the contract contain a clause that would award either party liquidated damages for breach or a fee upon the termination of a contract (termination fee)?
Revenue/Profit Sharing (R)	Highlight the parts (if any) of this contract related to "Revenue/Profit Sharing" that should be reviewed by a lawyer. Details: Is one party required to share revenue or profit with the counterparty for any technology, goods, orňservices?
Third Party Beneficiary (R)	Highlight the parts (if any) of this contract related to "Third Party Beneficiary" that should be reviewed by a lawyer. Details: Is there a non-contracting party who is a beneficiary to some or all of the clauses in the contract and therefore can enforce its rights against a contracting party?
Non-Transferable License (R)	Highlight the parts (if any) of this contract related to "Non-Transferable License" that should be reviewed by a lawyer. Details: Does the contract limit the ability of a party to transfer the license being granted to a third party?
Affiliate License-Licensee (R)	Highlight the parts (if any) of this contract related to "Affiliate License-Licensee" that should be reviewed by a lawyer. Details: Does the contract contain a license grant to a licensee (incl. sublicensor) and the affiliates of such licensee/sublicensor?
Most Favored Nation (R)	Highlight the parts (if any) of this contract related to "Most Favored Nation" that should be reviewed by a lawyer. Details: Is there a clause that if a third party gets better terms on the licensing or sale of technology/goods/services described in the contract, the buyer of such technology/goods/services under the contract shall be entitled to those better terms?
Joint Ip Ownership (R)	Highlight the parts (if any) of this contract related to "Joint Ip Ownership" that should be reviewed by a lawyer. Details: Is there any clause providing for joint or shared ownership of intellectual property between the parties to the contract?

Continued on next page

Table B.1 – Continued from previous page

Category	Question
Unlimited/All-You-Can-Eat-License (R)	Highlight the parts (if any) of this contract related to "Unlimited/All-You-Can-Eat-License" that should be reviewed by a lawyer. Details: Is there a clause granting one party an Äúenterprise,Äù Äúall you can eatÄù or unlimited usage license?
Source Code Escrow (R)	Highlight the parts (if any) of this contract related to "Source Code Escrow" that should be reviewed by a lawyer. Details: Is one party required to deposit its source code into escrow with a third party, which can be released to the counterparty upon the occurrence of certain events (bankruptcy,ň insolvency, etc.)?
Affiliate License-Licensor (R)	Highlight the parts (if any) of this contract related to "Affiliate License-Licensor" that should be reviewed by a lawyer. Details: Does the contract contain a license grant by affiliates of the licensor or that includes intellectual property of affiliates of the licensor?
Non-Disparagement (R)	Highlight the parts (if any) of this contract related to "Non-Disparagement" that should be reviewed by a lawyer. Details: Is there a requirement on a party not to disparage the counterparty?

Table B.2. Altered small (Alt-S) question formulations

Category	Question
Document Name	Show the sections of this contract (if there are any) related to the "Document Name" . This is the name of the contract.
Parties	Show the sections of this contract (if there are any) related to the "Parties" of the agreement. This is the entities/parties who signed the contract
Agreement Date	Show the sections of this contract (if there are any) related to the "Agreement Date". This is the date of the contract
Effective Date	Show the sections of this contract (if there are any) related to the "Effective Date". This is the date when the contract is effective

Continued on next page

Table B.2 – Continued from previous page

Category	Question
Expiration Date	Show the sections of this contract (if there are any) related to the "Expiration Date". This is the date when the contract's initial term expires
Renewal Term	Show the sections of this contract (if there are any) related to the "Renewal Term". This is the renewal term after the initial term expires and includes automatic extensions and unilateral extensions with prior notice.
Notice Period To Terminate Renewal	Show the sections of this contract (if there are any) related to the "Notice Period To Terminate Renewal". This is the notice period required to terminate renewal
Governing Law	Show the sections of this contract (if there are any) related to the "Governing Law". This which state or country's law governs the interpretation of the contract
Most Favored Nation	Show the sections of this contract (if there are any) related to the "Most Favored Nation". This is a clause that if a third party gets better terms on the licensing or sale of technology/goods/services described in the contract, the buyer of such technology/goods/services under the contract shall be entitled to those better terms
Non-Compete	Show the sections of this contract (if there are any) related to the "Non-Compete". This is a restriction on the ability of a party to compete with the counterparty or operate in a certain business or technology or geography sector
Exclusivity	Show the sections of this contract (if there are any) related to the "Exclusivity". This is an exclusive dealing commitment with the counterparty and includes a commitment to procure all requirements from one party of certain services, technology, or goods or a prohibition on licensing or selling technology, goods or services to third parties, or a prohibition on working or collaborating with other parties), whether during the contract, after the contract ends, or both

Continued on next page

Table B.2 – Continued from previous page

Category	Question
No-Solicit of Customers	Details: Is a party restricted from contracting or soliciting customers or partners of the counterparty, whether during the contract or after the contract ends (or both)?. Highlight the parts (if any) of this contract related to "No-Solicit of Customers" that should be reviewed by a lawyer.
Competitive Restriction Exception	Show the sections of this contract (if there are any) related to the "Competitive Restriction Exception". This is an exception or carveouts to the non-compete, exclusivity, and no-solicit of customers clauses
No-Solicit of Employees	Show the sections of this contract (if there are any) related to the "No-Solicit of Employees". This is a restriction on a party's soliciting or hiring employees and/or contractors from the counterparty
Non-Disparagement	Show the sections of this contract (if there are any) related to the "Non-Disparagement". This is a requirement on a party not to disparage the counterparty
Termination for Convenience	Show the sections of this contract (if there are any) related to the "Termination for Convenience". Can party terminate this contract without cause
Right of First Refusal, Right of First Offer, Right of First Negotiation	Show the sections of this contract (if there are any) related to the "Right of First Refusal, Right of First Offer, Right of First Negotiation". This is a clause granting one party a right of first refusal, right of first offer or right of first negotiation to purchase, license, market, or distribute equity interest, technology, assets, products or services
Change of Control	Show the sections of this contract (if there are any) related to the "Change of Control". Does one party have the right to terminate or is consent or notice required of the counterparty if such party undergoes a change of control, such as a stock sale, merger, transfer of all or substantially all of its assets or business, or assignment by operation of law?
Anti-Assignment	Show the sections of this contract (if there are any) related to the "Anti-Assignment". Is consent or notice required of a party if the contract is assigned to a third party?

Continued on next page

Table B.2 – Continued from previous page

Category	Question
Revenue/Profit Sharing	Show the sections of this contract (if there are any) related to the "Revenue/Profit Sharing". This is a clause related to if one party is required to share revenue or profit with the counterparty for any services, technology, or goods?
Price Restrictions	Show the sections of this contract (if there are any) related to the "Price Restrictions". This is a restriction on the ability of a party to raise or reduce prices of technology, goods, or services provided
Minimum Commitment	Show the sections of this contract (if there are any) related to the "Minimum Commitment". This is a minimum minimum amount, order size or units per-time period that one party has to buy from the counterparty under the contract
Volume Restriction	Show the sections of this contract (if there are any) related to the "Volume Restriction". This is a fee consent or increase requirement, etc. if one party's use of the product/services exceeds certain threshold
IP Ownership Assignment	Show the sections of this contract (if there are any) related to the "IP Ownership Assignment". Does intellectual property created by one party become the property of the counterparty, either upon the occurrence of certain events or per the terms of the contract?
Joint IP Ownership	Show the sections of this contract (if there are any) related to the "Joint IP Ownership". Is there any clause providing for joint or shared ownership of intellectual property between the parties to the contract?
License Grant	Show the sections of this contract (if there are any) related to the "License Grant". Is there a clause granting a license by one party to its counterparty?
Non-Transferable License	Show the sections of this contract (if there are any) related to the "Non-Transferable License". Is the contract limited to the ability of a party to transfer the license being granted to a third party?

Continued on next page

Table B.2 – Continued from previous page

Category	Question
Affiliate License-Licensor	Show the sections of this contract (if there are any) related to the "Affiliate License-Licensor". Is the contract containing a license grant by affiliates of the licensor or that includes intellectual property of affiliates of the licensor?
Affiliate License-Licensee	Show the sections of this contract (if there are any) related to the "Affiliate License-Licensee". Is the contract containing a license grant to a licensee (incl. sublicensor) and the affiliates of such licensee/sublicensor?
Unlimited/All-You-Can-Eat License	Show the sections of this contract (if there are any) related to the "Unlimited/All-You-Can-Eat License". Is there a clause granting one party an enterprise, all you can eat or unlimited usage license?
Irrevocable or Perpetual License	Show the sections of this contract (if there are any) related to the "Irrevocable or Perpetual License". Is the contract containing a license grant that is perpetual or irrevocable?
Source Code Escrow	Show the sections of this contract (if there are any) related to the "Source Code Escrow". Is one party required to deposit its source code into escrow with a third party, which can be released to the counterparty upon the occurrence of certain events (bankruptcy, insolvency, etc.)?
Post-Termination Services	Show the sections of this contract (if there are any) related to the "Post-Termination Services". Is a party subject to obligations after the expiration or termination of a contract, including any post-termination transfer of ip, transition, payment, wind-down, last-buy, or similar commitments?
Audit Rights	Show the sections of this contract (if there are any) related to the "Audit Rights". Is there a clause related to a party having the right to audit the records, books, or physical locations of the counterparty to ensure compliance with the contract?

Continued on next page

Table B.2 – Continued from previous page

Category	Question
Uncapped Liability	Show the sections of this contract (if there are any) related to the "Uncapped Liability". Is there a clause in the contract related to a party's uncapped liability upon the breach of its obligation in the contract? Including uncap liability for a particular type of breach such as breach of confidentiality obligation or IP infringement.
Cap on Liability	Show the sections of this contract (if there are any) related to the "Cap on Liability". Is there a clause in the contract related to a cap on liability upon the breach of a party's obligation? Including time limitation for the counterparty to bring claims or maximum amount for recovery.
Liquidated Damages	Show the sections of this contract (if there are any) related to the "Liquidated Damages". Is there a clause in the contract that would award either party liquidated damages for breach or a fee upon the termination of a contract?
Warranty Duration	Show the sections of this contract (if there are any) related to the "Warranty Duration". This is the duration of any warranty against defects or errors in products, technology, or services provided under the contract?
Insurance	Show the sections of this contract (if there are any) related to the "Insurance". Are there any requirement for insurance that must be maintained by one party for the benefit of the counterparty?
Covenant Not to Sue	Show the sections of this contract (if there are any) related to the "Covenant Not to Sue". Are there any restrictions on a party from contesting the validity of the counterparty's ownership of intellectual property or otherwise bringing a claim against the counterparty for matters unrelated to the contract?
Third Party Beneficiary	Show the sections of this contract (if there are any) related to the "Third Party Beneficiary". Are there any non-contracting parties who is a beneficiary to some or all of the clauses in the contract and therefore can enforce its rights against a contracting party?

Table B.3. Altered Large (Alt-L) question formulations

Category	Question
Document Name	What is the name of this legal document?
Parties	Who are the parties to this legal document? The parties to the contract are the two or more parties who signed the contract.
Agreement Date	What is the agreement date of this contract? Agreement Date means the date as of which this Agreement is dated.
Effective Date	In contract law, the effective date is the date that an agreement or transaction between or among signatories becomes binding. What is the effective date of this contract?
Expiration Date	What is the expiration date of the initial terms of this contract? The expiration of a contract puts an end to all the engagements of the parties, except to those which arise from the non- fulfillment of obligations created during its existence
Renewal Term	What sections are related to the "Renewal Term" in this contract? Renewal is keeping an existing arrangement in force for an additional period of time, such as a lease, a promissory note, insurance policy or any other contract. Renewal usually requires a writing or some action which evidences the new term.
Notice Period To Terminate Renewal	Show everything related to the notice required to terminate renewal
Governing Law	What is the "Governing Law" of this contract?. Governing law is the law stipulated in a contract to determine a dispute
Most Favored Nation	Is there any sections of this contract related to most favored nation?. Most Favored Nations (MFN) clauses (also known as antidiscrimination clauses or most-favored customer clauses) are common in business today. These provisions require that the supplier will treat a particular customer no worse than all other customers.

Continued on next page

Table B.3 – Continued from previous page

Category	Question
Non-Compete	Is there a any sections of this contract related to "Non-Compete". A non-compete agreement is a contract between two parties, usually two individuals or one company and one individual, in which one of the individuals promises not to compete with the other individual or company once their relationship with the company has ended. That is, he or she will not start, join, or buy a business that is similar to, and in competition with, the other.
Exclusivity	Highlight any parts of this contract related to an "Exclusivity" clause. An exclusivity clause is part of a bigger legal document that restricts the signer from buying, selling, or promoting any goods or services from any person or company other than the issuing company associated with the contract. In other words, the company or individual works exclusively with the issuer of the contract.
No-Solicit Of Customers	Show everything related to "No-Solicit of Employees". This is a clause that prohibits one party from soliciting the customers of another party during a specified period of time
No-Solicit Of Employees	Show everything related to "No-Solicit of Employees". This is a clause that prohibits one party from soliciting the employees of another party during a specified period of time
Non-Disparagement	Show everything related to "Non-Disparagement" in this contract?. Non-disparagement clauses prevent parties from making derogatory comments about the other party.
Termination For Convenience	Is there a clause related to "Termination for Convenience"?. A contractual right to terminate an agreement for any reason. It may also be referred to as termination without cause. A right to terminate for convenience usually requires the terminating party to provide a certain period of notice before the termination is effective and usually in writing

Continued on next page

Table B.3 – Continued from previous page

Category	Question
Change Of Control	Highlight any parts of this contract related to the "Change of Control". A provision in an agreement giving a party certain rights (such as consent, payment or termination) in connection with a change in ownership or management of the other party to the agreement.
Anti-Assignment	Show any section of this contract related to "Anti-Assignment". An anti-assignment clause prevents either of the parties to a contract from assigning tasks to a third party without the consent of the non-assigning party.
Revenue/Profit Sharing	Highlight all sections related to "Revenue/Profit Sharing". Revenue sharing is a somewhat flexible concept that involves sharing operating profits or losses among associated financial actors. Revenue sharing can exist as a profit-sharing system that ensures each entity is compensated for its efforts.
Minimum Commitment	Highlight any parts related to "Minimum Commitment". Clause sets out the minimum quantities of the goods or products that the buyer must purchase under the agreement and could include remedies for the buyer's failure to meet the minimum purchase requirement
Volume Restriction	Highlight any section related to "Volume Restriction" in this contract. This is a fee consent or increase requirement, etc. if one party's use of the product/services exceeds certain threshold
Ip Ownership Assignment	Highlight the sections of this contract related to the "IP Ownership Assignment". The assignment of intellectual property (IP) refers to the process by which ownership of work product created for one party is assigned to another party?
Joint Ip Ownership	Show the sections of this contract (if there are any) related to the "Joint IP Ownership". Joint ownership of intellectual property refers to the sharing of intellectual property (IP) rights to a particular invention between two or more parties. It usually occurs as a result of two or more people co-inventing a patentable product, creative work, design, or concept

Continued on next page

Table B.3 – Continued from previous page

Category	Question
License Grant	Show the sections of this contract related to the "License Grant". The object of the a license grant clause is to grant permission to the one party to use certain intellectual property rights of the other party
Non-Transferable License	Show the sections of this contract related to the "Non-Transferable License". Clauses ensuring that the entire agreement, or some of its rights, obligations, and terms; may not be transferred to a party that has not signed the original contract?
Affiliate License-Licensor	Show the sections of this contract (if there are any) related to the "Affiliate License-Licensor". A licensing agreement is a contract between a licensor and licensee in which the licensee gains access to the licensor's intellectual property. Does this legal document include a license grant that includes IP of partners of the licensor?
Affiliate License-Licensee	Show the sections of this contract (if there are any) related to the "Affiliate License-Licensee". A licensing agreement is a contract between a licensor and licensee in which the licensee gains access to the licensor's intellectual property. Is the legal document containing a license grant to a licensee (incl. sublicensor) and the affiliates of such licensee/sublicensor?
Unlimited/All-You-Can-Eat-License	Show the sections of this contract related to the "Unlimited/All-You-Can-Eat License". Are there any clauses giving one of the contracting parties and unlimited usage license also refeered to as an all you can eat license?
Irrevocable Or Perpetual License	Show the sections of this contract related to the "Irrevocable or Perpetual License". The terms irrevocable and perpetual mean different things. A perpetual license is one that has no given end. An irrevocable license cannot be cut short. Either a perpetual license or a license with a stated term can be either revocable or irrevocable.

Continued on next page

Table B.3 – Continued from previous page

Category	Question
Source Code Escrow	Show the sections of this contract related to the "Source Code Escrow". Source code escrow is the deposit of the source code of software with a third-party escrow agent. Escrow is typically requested by a party licensing software, to ensure maintenance of the software instead of abandonment or reelease in occurrence of special events.?
Audit Rights	Show the sections of this contract related to the "Audit Rights". This is clause that entitles one party to review another party including but not limited to work, records, physical locations, and books which may include self-assessments, third-party audits and other, official documents detailing the sufficiency of internal systems and controls?
Uncapped Liability	Show the sections of this contract related to the "Uncapped Liability". A clause stating that there is no limit to the damage a party incurs if things go wrong, and thus no limit to the money to be paid out in respect of that damage upon the breach of the contract. Includes but not limited to ip infringement and breach of confidentiality obligations.
Liquidated Damages	Show the sections of this contract (if there are any) related to the "Liquidated Damages". A fixed or determined sum agreed by the parties to a contract to be payable on breach of the contract by one of the parties
Covenant Not To Sue	Show the sections of this contract related to the "Covenant Not to Sue". A covenant not to sue is a legal agreement that obliges a party that could seek damages to refrain from suing the party that it has cause against? This could be for reasons of intellectual property, or for other reasons.
Third Party Beneficiary	Show the sections of this contract related to the "Third Party Beneficiary". A third-party beneficiary is a person or business that benefits from the terms of a contract made between two other parties. Are there any such parties in the contract that can enforce its rights against a contracting party?

APPENDIX C

Results and Analysis

C.1 Correlation and distribution

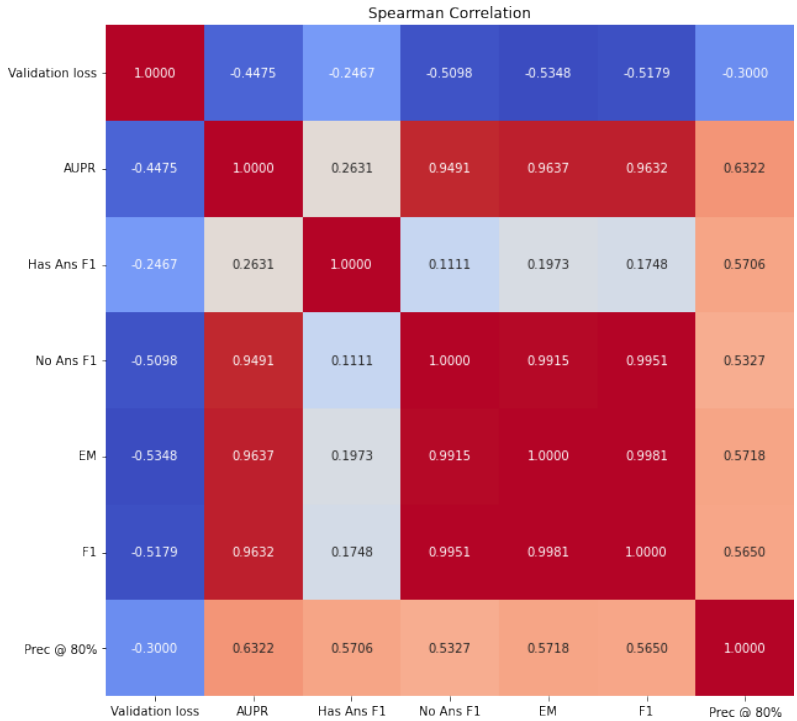


Figure C.1. Spearman correlation calculated from the data presented in Figure 5.1(b)

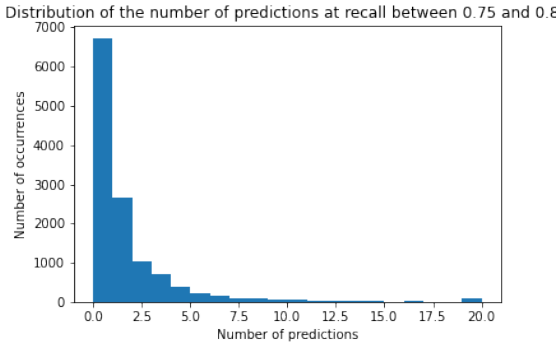


Figure C.2. Distribution of the number of predictions at a recall level between 0.75 and 0.85 non-included, based on the **RB-S** model checkpoint on the test set using the standard confidence intervals defined in Chapter 4 and *n-best=20*

C.2 Attention

CUAD text

Question
Highlight the parts (if any) of this contract related to "Document Name" that should be reviewed by a lawyer. Details: The name of the contract
Context
EXHIBIT 10.6 DISTRIBUTOR AGREEMENT THIS DISTRIBUTOR AGREEMENT is made by and between Electric City Corp., a Delaware corporation ('Company') and Electric City of Illinois LLC ('Distributor') this 7th day of September, 1999A. The Company's Business. The Company is presently engaged in the business of selling an energy efficiency device

SQuAD text

Question
When was the Latin version of the word Norman first recorded?
Context
The English name "Normans" comes from the French words Normans/Normanz, plural of Normant, modern French normand, which is itself borrowed from Old Low Franconian Nortmann "Northman" or directly from Old Norse Norðmaðr, Latinized variously as Nortmannus, Normannus, or Nordmannus (recorded in Medieval Latin, 9th century) to mean "Norseman, Viking".

BertViz

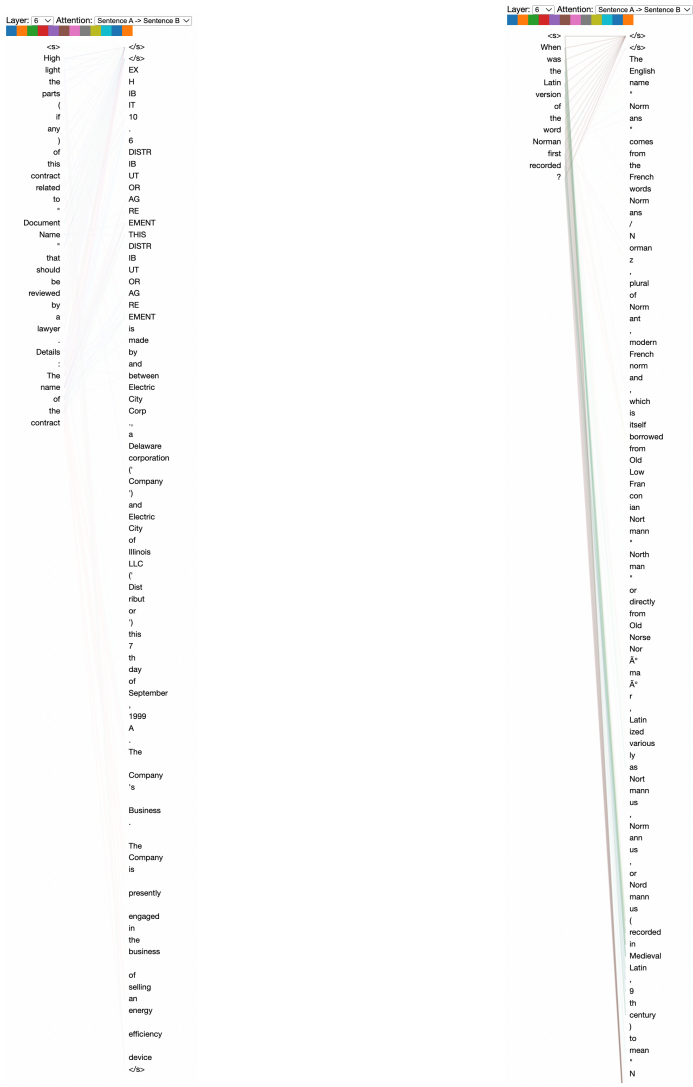


Figure C.3. BertViz Head View[69] of: The text in Section C.2 (left) and C.2 (right) using the **RB-S** model for CUAD and the **deepset/roberta-base-squad2** model from `Transformers` for SQuAD

((a)) Layer 1

((b)) Layer 2

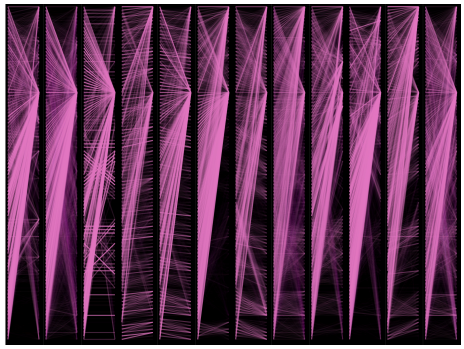
((c)) Layer 3

((d)) Layer 4

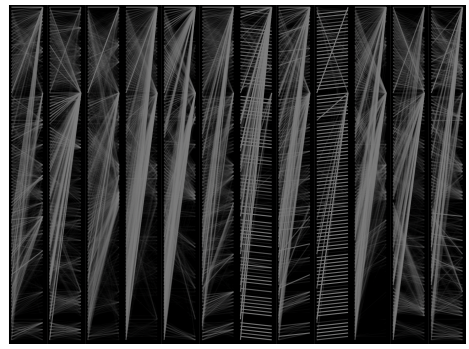
((e)) Layer 5

((f)) Layer 6

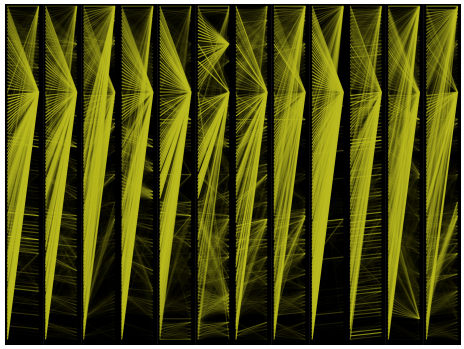
Figure C.4. Attention Model View[69] on the CUAD text from Section C.2 from layer 1-6 in the model where each column in the subfigures represents an attention head



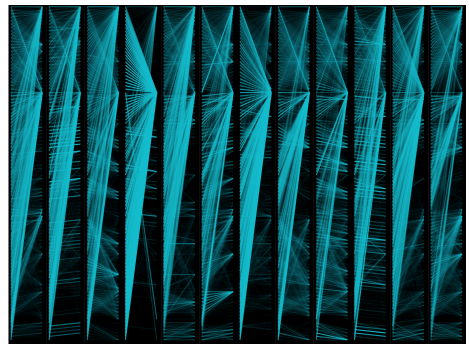
((a)) Layer 7



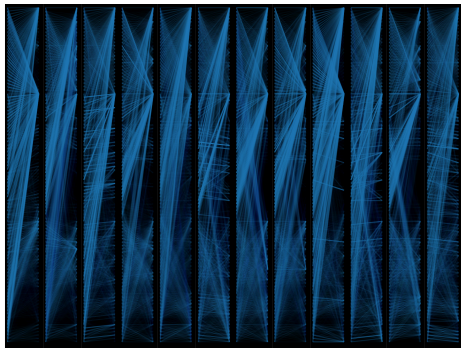
((b)) Layer 8



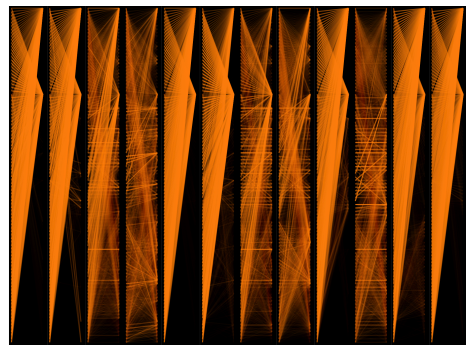
((c)) Layer 9



((d)) Layer 10

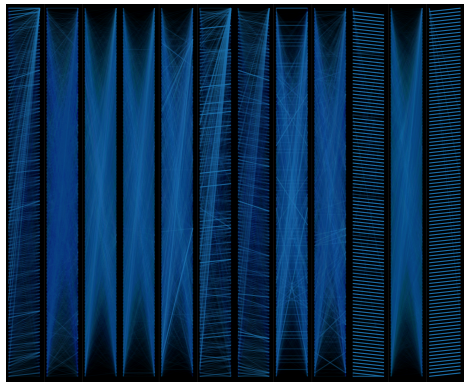


((e)) Layer 11

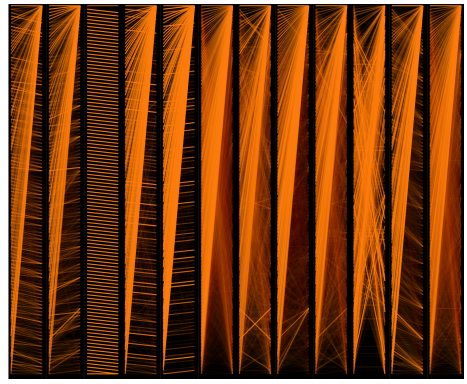


((f)) Layer 12

Figure C.5. Attention Model View[69] on the CUAD text from Section C.2 from layer 7-12 in the model where each column in the subfigures represents an attention head



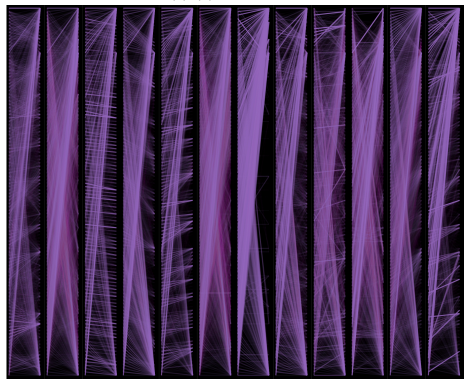
((a)) Layer 1



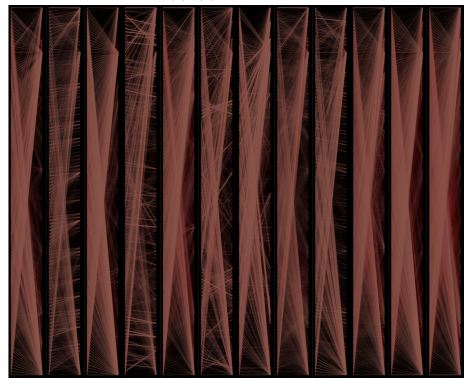
((b)) Layer 2

((c)) Layer 3

((d)) Layer 4

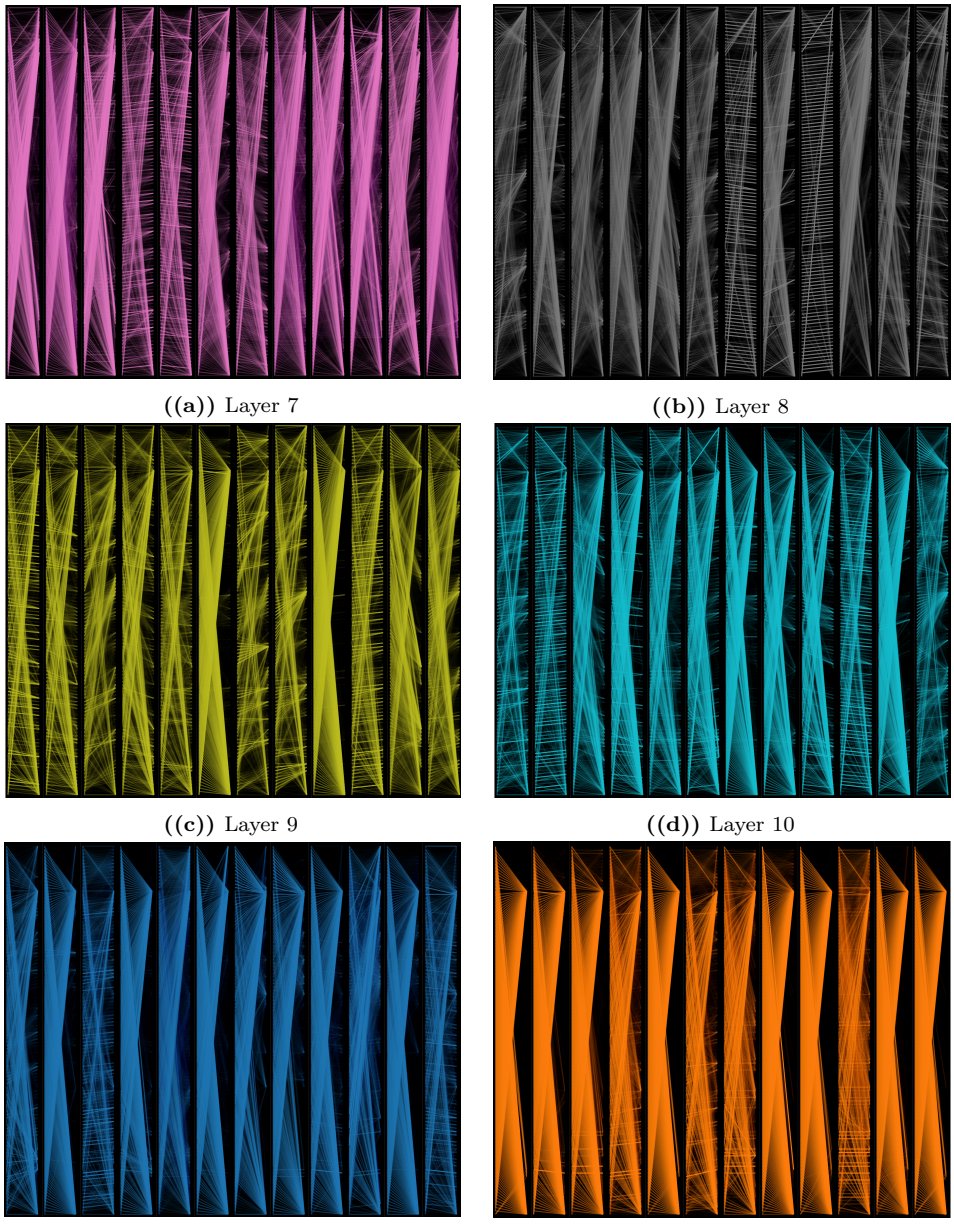


((e)) Layer 5



((f)) Layer 6

Figure C.6. Attention Model View[69] on the SQuAD text from Section C.2 from layer 1-6 in the model where each column in the subfigures represents an attention head



((a)) Layer 7

((b)) Layer 8

((c)) Layer 9

((d)) Layer 10

((e)) Layer 11

((f)) Layer 12

Figure C.7. Attention Model View[69] on the SQuAD text from Section C.2 from layer 7-12 in the model where each column in the subfigures represents an attention head

C.3 Model performance

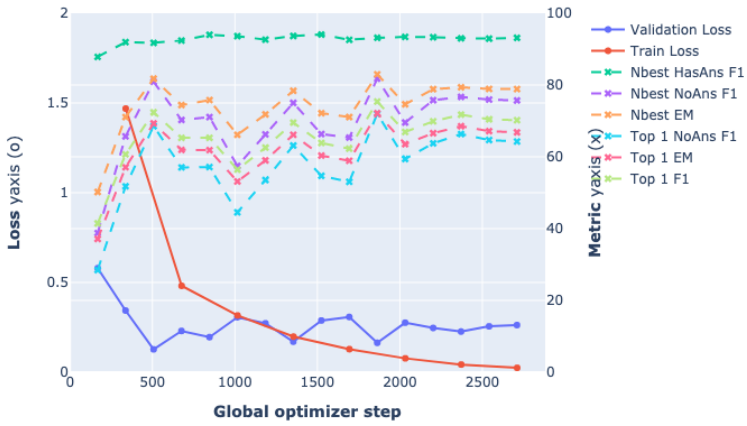


Figure C.8. Training and validation loss along with selected metrics for the **RB-S-Con-** $\delta = 1$ model

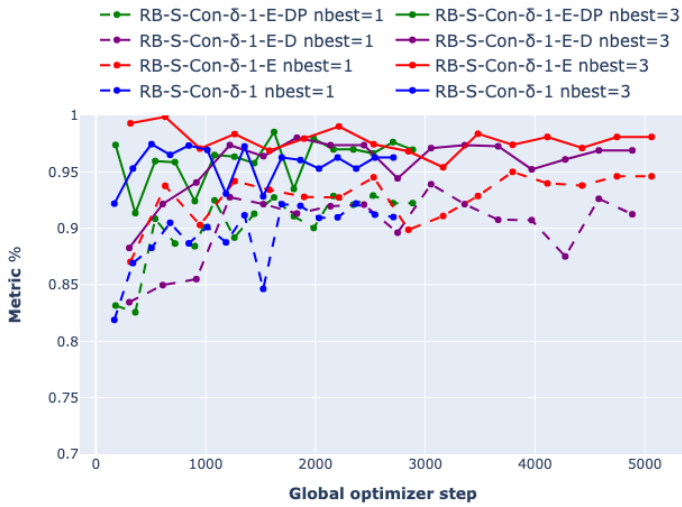


Figure C.9. Model performance with added answers as described in Section 6.1.3 on the Anti-Assignment question category

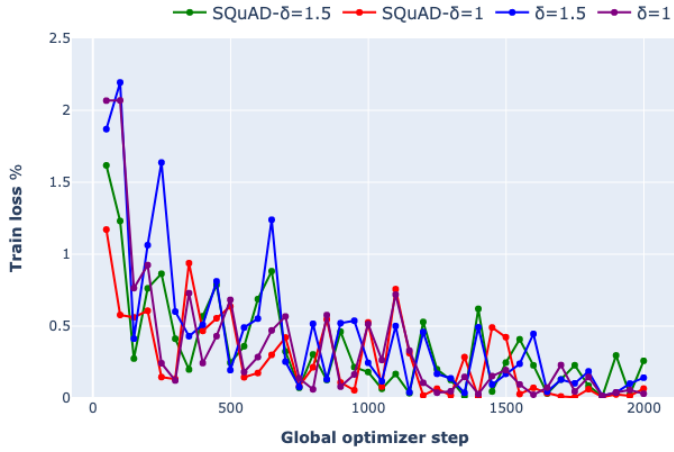


Figure C.10. Training loss from SQuAD pre-trained model analysis in Section 6.2

HA F1	NA F1	EM	F1	HA F1	NA F1	EM	F1	$\notin L$
Top 3				Top 1				-
88.75	81.55	80.20	83.69	80.52	70.80	69.06	73.69	3, 9, 10
69.19	8.20	13.39	26.34	54.49	3.95	7.60	18.98	4, 9, 10, 11
80.66	94.55	86.49	90.42	60.30	86.49	74.84	78.70	1, 3
91.27	84.96	84.29	86.83	82.35	72.97	72.12	75.76	1, 10
75.04	97.79	86.90	91.02	52.86	93.09	78.43	81.12	1, 3, 10
78.63	97.17	86.99	91.66	54.16	89.86	75.56	79.24	1, 3, 9
89.88	74.95	76.18	79.39	80.66	65.11	64.75	69.74	1, 9, 10
92.94	83.56	84.00	86.35	84.04	73.49	73.24	76.62	
90.78	73.96	76.06	78.97	83.44	65.45	66.55	70.80	9, 10
91.04	89.79	87.28	90.16	79.67	80.26	75.97	80.08	8, 9
90.12	86.93	85.13	87.88	80.23	74.06	72.24	75.90	5, 9
65.98	89.79	78.07	82.71	44.83	78.69	65.21	68.62	1, 2
38.94	20.18	19.99	25.76	27.18	16.13	14.97	19.42	2, 4, 5
39.64	36.28	31.90	37.28	27.17	24.17	21.04	25.06	2, 4, 6
87.75	69.91	72.55	75.22	80.47	66.27	66.86	70.49	3, 11
90.78	73.96	76.06	78.97	83.44	65.45	66.55	70.80	9, 10
72.89	65.59	64.40	67.76	58.94	44.01	45.22	48.45	3, 6
91.68	75.73	78.22	80.48	82.51	60.04	63.32	66.72	4
91.95	87.99	86.82	89.16	82.48	78.45	76.23	79.65	3
92.28	82.64	83.33	85.51	82.20	72.36	71.71	75.29	1
91.87	82.20	82.83	85.07	80.84	71.44	70.64	74.24	2
87.20	74.51	75.82	78.28	76.55	54.66	58.06	61.17	6
91.02	82.37	82.54	84.94	81.22	69.09	69.49	72.70	5
90.87	78.25	79.41	82.00	84.45	73.28	73.29	76.60	11
92.16	84.82	84.62	87.00	84.62	73.89	73.70	77.08	10
92.97	85.60	85.27	87.79	83.83	76.96	75.47	79.00	9

Table C.1. Full results table from Section 6.3 showing model performance for different configuration of deleted layers from the $\delta = 1$ model presented in Section 6.2. The results are obtained using the model checkpoint from the `global_step` = 1864

C.4 CUAD Paper Results

Model	AUPR	Prec @ 80% R	Prec@ 90% R
BERT-base	32.4	8.2	0
BERT-large	32.3	7.6	0
ALBERT-base	35.3	11.1	0
ALBERT-large	34.9	20.9	0
ALBERT-xlarge	37.8	20.5	0
ALBERT-xxlarge	38.4	31	0
RoBERTa-base	42.6	31.1	0
RoBERTa-base + Pre-training	45.2	34.1	0
RoBERTa-large	48.2	38.1	0
DeBERTa-xlarge	47.8	44	17.8

Table C.2. Results table from CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review[23] on the test set using a range of different models

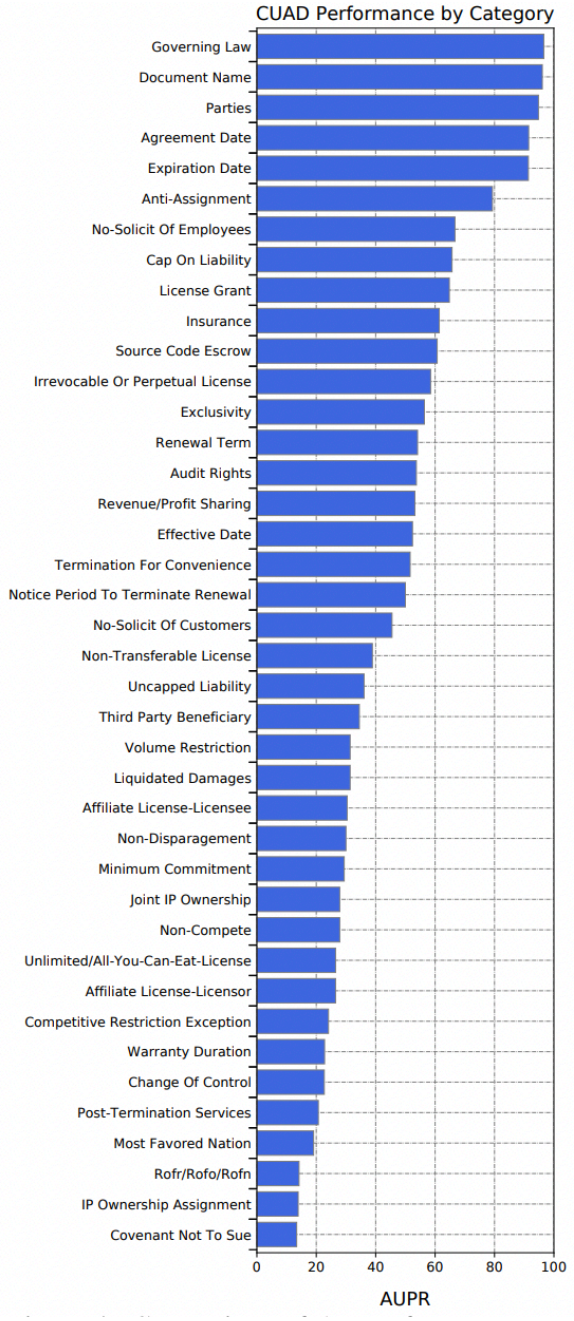


Figure C.11. Performance measured by AUPR by question category figure from Hendrycks et al. [23] based on **DeBERTa-xlarge** model checkpoint

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL `https://arxiv.org/abs/1409.0473`.

[3] Petr Baudiš and Jan Šedivý. Modeling of the question answering task in the yodaqa system. In *Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction - Volume 9283*, CLEF'15, page 222228, Berlin, Heidelberg, 2015. Springer-Verlag. ISBN 9783319240268. doi: 10.1007/978-3-319-24027-5_20. URL `https://doi.org/10.1007/978-3-319-24027-5_20`.

[4] Payal Biswas, Aditi Sharan, and Nidhi Malik. A framework for restricted domain question answering system. *Proceedings of the 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques, Icict 2014*, pages 613–620, 2014. doi: 10.1109/ICICICT.2014.6781351.

[5] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3302. URL `https://aclanthology.org/W14-3302`.

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

[7] Cristian Buciluundefined, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, page 535541, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150464. URL `https://doi.org/10.1145/1150402.1150464`.

[8] Ilias Chalkidis and Ion Androutsopoulos. A deep learning approach to contract element extraction. In *JURIX*, 2017.

[9] Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. Extracting contract elements. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, page 1928, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348911. doi: 10.1145/3086512.3086515. URL `https://doi.org/10.1145/3086512.3086515`.

[10] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.261. URL `https://aclanthology.org/2020.findings-emnlp.261`.

[11] Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. Lexglue: A benchmark dataset for legal language understanding in english. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dubln, Ireland, 2022.

[12] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051, 2017. URL `http://arxiv.org/abs/1704.00051`.

[13] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does bert look at? an analysis of bert's attention, 2019. URL `https://arxiv.org/abs/1906.04341`.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

[15] Hai Doan-nguyen and Leila Kosseim. Improving the precision of a closed-domain question-answering system with semantic information. 2008.

[16] Yadolah Dodge. *Spearman Rank Correlation Coefficient*, pages 502–505. Springer New York, New York, NY, 2008. ISBN 978-0-387-32833-1. doi: 10.1007/978-0-387-32833-1_379. URL https://doi.org/10.1007/978-0-387-32833-1_379.

[17] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. URL https://github.com/PyTorchLightning/pytorch-lightning.

[18] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning, 2017. URL https://arxiv.org/abs/1705.03122.

[19] R. Stuart Geiger, Kevin Yu, Yanlai Yang, Mindy Dai, Jie Qiu, Rebekah Tang, and Jenny Huang. Garbage in, garbage out? In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. ACM, jan 2020. doi: 10.1145/3351095.3372862. URL https://doi.org/10.1145%2F3351095.3372862.

[20] R. Stuart Geiger, Dominique Cope, Jamie Ip, Marsha Lotosh, Aayush Shah, Jenny Weng, and Rebekah Tang. garbage in, garbage out revisited: What do machine learning application papers report about human-labeled training data? *Quantitative Science Studies*, 2:795–827, 2021.

[21] Google. Recent advances in google translate. https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html. Accessed: 2022-05-05.

[22] Allison Hegel, Marina Shah, Genevieve Peaslee, Brendan Roof, and Emad Elwany. The law of large documents: Understanding the structure of legal contracts using visual cues. *CoRR*, abs/2107.08128, 2021. URL https://arxiv.org/abs/2107.08128.

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

[24] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015. URL http://arxiv.org/abs/1506.03340.

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

[26] Sarthak Jain and Byron C. Wallace. Attention is not explanation, 2019. URL https://arxiv.org/abs/1902.10186.

[27] Ł ukasz Kaiser and Samy Bengio. Can active memory replace attention? In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/fb8feff253bb6c834deb61ec76baa893-Paper.pdf.

[28] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time, 2016. URL https://arxiv.org/abs/1610.10099.

[29] Yoon Kim. Convolutional neural networks for sentence classification, 2014. URL https://arxiv.org/abs/1408.5882.

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

[31] Yuta Koreeda and Christopher Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.findings-emnlp.164.

[32] Sotiris Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30:25–36, 11 2005.

[33] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, 2018. URL https://arxiv.org/abs/1808.06226.

[34] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[35] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations, 2019. URL https://arxiv.org/abs/1909.11942.

[36] Xiaoxu Liu, Haoye Lu, and Amiya Nayak. A spam transformer model for sms spam detection. *IEEE Access*, PP:1–1, 05 2021. doi: 10.1109/ACCESS.2021.3081479.

[37] 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, 2019. URL https://arxiv.org/abs/1907.11692.

[38] Google LLC. Google cloud platform. URL https://console.google.com.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL https://arxiv.org/abs/1711.05101.

[40] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation, 2015. URL https://arxiv.org/abs/1508.04025.

[41] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one?, 2019. URL https://arxiv.org/abs/1905.10650.

[42] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013. URL https://arxiv.org/abs/1301.3781.

[43] Lukas Muttenthaler. Subjective question answering: Deciphering the inner workings of transformers in the realm of subjectivity, 2020. URL https://arxiv.org/abs/2006.08342.

[44] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help?, 2019. URL https://arxiv.org/abs/1906.02629.

[45] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.03.091. URL https://www.sciencedirect.com/science/article/pii/S092523122100477X.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[47] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[48] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[49] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[50] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019. URL `https://arxiv.org/abs/1910.10683`.

[51] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016. URL `http://arxiv.org/abs/1606.05250`.

[52] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad, 2018. URL `https://arxiv.org/abs/1806.03822`.

[53] Pranav Rajpurkar, Robin Jia, and Percy Liang. Squad leaderboard. `https://rajpurkar.github.io/SQuAD-explorer/`, 2022. Accessed: 2022-06-01.

[54] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model?, 2020. URL `https://arxiv.org/abs/2002.08910`.

[55] Sebastian Ruder. *Neural Transfer Learning for Natural Language Processing*. PhD thesis, National University of Ireland, Galway, 2019.

[56] Alexander Rush. The annotated transformer. pages 52–60, 01 2018. doi: 10.18653/v1/W18-2509.

[57] Beth Sagar-Fenton and Lizzy McNeill. How many words do you need to speak a language? `https://www.bbc.com/news/world-44569277`. Accessed: 2022-06-23.

[58] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. On the effect of dropping layers of pre-trained transformer models, 2020. URL `https://arxiv.org/abs/2004.03844`.

[59] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019. URL `https://arxiv.org/abs/1910.01108`.

[60] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012. doi: 10.1109/ICASSP.2012.6289079.

[61] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units, 2015. URL `https://arxiv.org/abs/1508.07909`.

[62] Samuel Suík. Compressing bert for faster prediction, 2019. URL https://rasa.com/blog/compressing-bert-for-faster-prediction-2/.

[63] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. URL https://arxiv.org/abs/1512.00567.

[64] Igor V. Tetko, Pavel Karpov, Ruud Van Deursen, and Guillaume Godin. State-of-the-art augmented nlp transformer models for direct and single-step retrosynthesis. *Nature Communications*, 11(1):5575, 2020. ISSN 20411723. doi: 10.1038/s41467-020-19266-y.

[65] Rob Toews. Ai will transform the field of law. *Forbes*, 2019. URL https://www.forbes.com/sites/robtoews/2019/12/19/ai-will-transform-the-field-of-law/?sh=198aad317f01.

[66] Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. Ledgar: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *LREC*, 2020.

[67] Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A. Gers. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 18231832, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3358028. URL https://doi.org/10.1145/3357384.3358028.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[69] Jesse Vig. A multiscale visualization of attention in the transformer model, 2019. URL https://arxiv.org/abs/1906.05714.

[70] Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model, 2019. URL https://arxiv.org/abs/1906.04284.

[71] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2018. URL https://arxiv.org/abs/1804.07461.

[72] Sandra Wankmuller. Neural transfer learning with transformers for social science text analysis [arxiv]. *Arxiv*, page 67 pp., 2021.

[73] Lilian Weng. How to build an open-domain question answering system? *lilianweng.github.io*, 2020. URL https://lilianweng.github.io/posts/2020-10-29-odqa/.

[74] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation, 2019. URL https://arxiv.org/abs/1908.04626.

[75] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

[76] Zack Written. Zack witten: Extracting structured data from legal documents, 2018. URL https://www.youtube.com/watch?v=KrXJmaSHBJU&ab_channel=PyData.

[77] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019. URL https://arxiv.org/abs/1906.08237.