# Enhancing the CUAD Dataset to Improve Automated Legal Contract Review

SAM ZENG
SID: 480363907

Supervisor: Dr Ying Zhou

This thesis is submitted in partial fulfilment of
the requirements for the degree of
Bachelor of Advanced Computing

School of Computer Science
The University of Sydney
Australia

08 November 2022

# Student Plagiarism: Compliance Statement

I certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Academic Board Policy: Academic Dishonesty and Plagiarism can lead to the University commencing proceedings against me for potential student misconduct under the 2012 Academic Dishonesty and Plagiarism in Coursework Policy.

This Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

**Name: Sam Zeng**

**Signature:**  **Date:** 08/11/2022

# Abstract

The Contract Understanding Atticus Dataset (CUAD) is a highly specialised contribution for deep learning within the legal domain (Hendrycks et al., 2021). While it is well annotated, there is room for improvement to its accuracy in contract review and related ML tasks. We argue that the relatively low performance of the QA models is caused by the data imbalance issue in the original data set. We propose two enhancement strategies to address this issue. We find that while steps have been taken to address imbalance within the dataset, there is high imbalance within the data of different categories. Additionally, as CUAD models the Stanford Question and Answer Dataset (SQuAD), we find that their adaption of fitting entire contracts as context paragraphs results in issues of excess false positives. Thus, we propose separating each question answer contract into smaller paragraphs to be more in line with the SQuAD data format and benefit more from unanswerable questions. We also propose splitting the label categories into different datasets grouped by label count to prevent particular categories from dominating the samples until further annotations for lower label-count categories can be introduced.



# Acknowledgements

I would like to thank my supervisor Dr Ying Zhou for taking the time to meet with me weekly and guide me through a lot of the issues we encountered through the process.

# Contents

# List of Figures

# List of Tables

# 1. Introduction

## 1.1 Problem statement

Contract Review is an extremely labour and time intensive field in the law industry despite being a relatively low-level process. The market research company Provoke Insights has found in a report by Onit (2022) that 40% of legal professionals spend over 50% of their work hours in contract review or management. Thus, automation or partial automation of this process would serve to significantly improve the efficiency of legal professionals and the legal services industry.

Contract Review has been a relatively unexplored domain within Natural Language Processing (NLP), likely due to its high specialisation and lack of data. Biagioli et. al. (2005) found success in extracting semantic information from contracts, identifying provisions such as “repeal” and “penalty”. Chalkidis et. al. (2017) attempted to utilise simple ML classifiers to extract basic contract elements such as contract titles, parties and dates, finding the most success when paired with manually written post-processing rules.

The Atticus Project (Hendrycks et al., 2021) has attempted to utilize question answering (QA) NLP techniques and machine learning to fully automate the low-level contract review process and identify existence and location of certain legal clauses or terminology within a contract, producing the CUAD dataset. However, this dataset has yielded lower than ideal results when paired with contemporary fine-tuned models such as the RoBERTa architecture (Hendrycks et. al., 2021). We believe that this dataset is a promising avenue in automating contract review, and thus the focus of this project will be to improve this dataset and enhance its performance in automated contract review.

## 1.2 Research Aim

The aim of this project is to analyse the CUAD dataset to determine methods of improving performance of their dataset on benchmark NLP-focused models.

While CUAD is well annotated and highly domain-specific, there is room for significant improvement to its accuracy in contract review and related ML tasks. We argue that the relatively low performance of the question answer models is caused by imbalanced datasets.

Notably, of the dataset's 41 categories, there is a significant spread in performance between the top performing categories and the lowest. This implies many avenues of potential improvements which could be addressed in order to improve the overall performance of this dataset. Additionally, there are significant imbalances in the samples within the dataset, in terms of both the label counts per category as well as the positive and negative samples generated when a sliding window is passed through the contract to generate features. We argue that these imbalances are a significant contributor to the low performance of this dataset. Furthermore, as CUAD models the Stanford Question and Answer Dataset (SQuAD), we find that their adaption of fitting entire contracts as context paragraphs results in issues resulting in the model generating too many false positives resulting in poor performance from overprediction.

Therefore, the focus of experimentation is in providing solutions to mitigate the data imbalance and overpredicting issues present within CUAD.

## 1.3 Thesis Contributions

This work provides the following contributions:

- Present and analyse issues with the structure of CUAD
- Provide two areas of approach in revising CUAD which improve performance in automated contract review
    - Modifying the context paragraphs from being the entire contract to split up paragraphs
    - Splitting the model to train specifically on more similarly balanced subsets of categories

## 1.4 Thesis Structure

The remainder of this thesis is structured as follows:

Section 2 reviews literature and past work in the legal NLP domain with focus on contract analysis and review. This involves drawing insights from their work to justify the methodology of our project, as well as examining the key components to our methodology. Section 3 describes CUAD and details the methodology used to experiment and improve the dataset, with a review and empirical justification of techniques. Section 4 presents the results, analysis and discussion. Section 5 concludes the paper and provides insight into the limitations of our work and suggest options for future work.

# 2. Literature Review / Related Works

The purpose of this review is to analyse past and current trends in the legal NLP domain with respect to the contract review process. The research into legal NLP is lower compared to other domains within NLP due to the high barrier of specialisation required to analyse legal terms (Hendrycks et. al., 2021). Research with a focus on contract review is thus further limited. We find that CUAD, produced by Hendrycks et. al. (2021) provides a strong benchmark for automated contract review whilst leaving room to explore improvements in their dataset to improve performance when utilising conventional NLP models.

## 2.1 Legal NLP Domain

Biagioli et. al. (2005) found success in extracting semantic information from legal clauses, identifying provisions such as “repeal” and “penalty”. They broke down specific paragraph clauses from Italian legislative text and applied both Support Vector Machines (SVM) binary classification and NLP processing techniques on the data, with both techniques yielding strong results. While related to information extraction in the legal domain, the research in Biagioli et. al. (2015) is only tangentially relevant as the focus of our project is more specialised.

Curtotti & McCreath (2011) utilise NLP techniques to conduct phrase analysis of Australian contract language, extracting information on vocabulary usage, sentence length and phrase usage. Their results show clear collocations of terms such as ‘intellectual property’ and ‘governing law’ within the contracts. While not related to automating contract review directly, this research provides meaningful insight into training using smaller contract sections as they would be more likely to capture information related to the section topic, thus supporting the method of paragraph training.

Chalkidis et. al. (2019) applied NLP text classification to extensively label 57,000 EU legislative documents with over 4000 labels drawn from the European Vocabulary (EUROVOC). They conduct their learning with various ML models and found the

highest degree of success with BERT based models, arguing that strong pretrained models with task-specific fine-tuning yields high performance even in highly specialised tasks. This shows that utilising a BERT based model for the benchmark of comparison is significant as application of automated contract review in practice will likely utilise such models.

## 2.2 Contract Review

In the legal context, contract review is defined as the process of analysing contracts to determine key terms, sections and relevant parties, with an assessment of fairness and risk each party bears, as well as their rights and obligations (Hendrycks, 2021; Ward, 2021). This process is highly time consuming as legal professionals will often be required to comb through vast amounts of text to identify key terms and clauses prior to review. As the evaluation of risk, fairness and obligations are a significantly higher-level process, such a task is more difficult to automate, and we have found no present research attempting such task. Thus, for the purposes of this review and subsequent project, we define contract review as the lower-level process of searching documents and locating particular clauses, terms and conditions within a contract. Within the legal NLP domain there has been effort to speed up this process through partial or full automation of contract review.

**Automated Review of Single Contract Types**

Indukuri & Krishna (2010) were able to classify payment related clauses within business e-contracts. While they were able to achieve success, their task focused on only one contract element in a corpus of primarily payment-centric contracts of loan and security agreements. Gao et. al. (2012) conducts a similar investigation extracting information on service exceptions in business payment contracts, which is a similarly narrow scope in comparison to the large coverage of contract type and elements in CUAD. Thus, these findings are too specific to be utilised for the broader scope of our project.

Leivaditi et. al. (2020) also conduct a similarly specialised experiment, identifying contract elements they refer to as “red flags” to identify risk bearing in lease contracts. They utilise BERT based methods with a similarly heavily imbalanced dataset as in Hendrycks et. al. (2021), with a low training epoch of 3 in order to prevent overfitting. Their training parameters are similar to the process in Hendrycks et. al. (2021), and thus provides justification to the use of training parameters in CUAD. Additionally, their use of area under precision recall and precision at 80% recall as evaluation metrics are also common to Hendrycks et. al. (2021) and so provides additional basis to the metrics to be used in our evaluation.

**Simple Element Extraction**

Roegiest et. al. (2018) present similar research topic to Hendrycks et. al. (2021) in locating clauses and contract elements such as “change of control”, within a more varied set of legal contracts. These contract elements share overlap with those annotated in CUAD, however their method focuses on non-deep learning models including Conditional Random Fields (CRF) and SVMs. They notably exclude experimentation on deep learning networks including Recurrent Neural Networks (RNNs) and more complex models, citing computational requirements and time. Therefore, as these considerations are less of a priority for our project, this justifies the decision to utilise state-of-the-art BERT models to yield greater performance. We find further evidence supporting the use of BERT models for our project in Elwany et. al. (2019), who fine-tune BERT on a large-scale legal contract corpus to improve detection of two specific agreement terms.

Chalkidis et. al. (2017) attempted to utilise simple ML classifiers to extract basic contract elements such as contract titles, parties and dates. This research is similar to Hendrycks et. al. (2021), however they experimented with only 11 contract elements that were more broad or easier to identify, as opposed to the 41 in CUAD. Additionally, their method relied on handcrafted features and found the highest success using manually written post-processing rules for issues such as errors in sliding-windows classifiers missing parts of dates, for example the year after a string of date and month. While possible to review and process issues such as this given a smaller set of categories

and samples, given the higher amount of data and categories in CUAD, such a task would be significantly more time consuming and less practical to apply to our project.

**Comprehensive Automated Review**

The work in Hegel et. al. (2021) is the most similar to our project. Their work also involves attempts at improving the results obtained in Hendrycks et. al. (2021) by segmenting CUAD, however their attempts differ to ours methodology. Hegel et. al. (2021) utilise optical character recognition (OCR) to introduce physical aspects of contract to their model, factoring in aspects such as page layout, text placement, italicisation and visual grouping. Our work focuses more on addressing data imbalances and reduction in irrelevant features which inhibit training. Thus, separation of contracts into smaller sections is only a portion of our work and less focus is made on physical aspect and technique of separating the contract into components. Rather, this work focuses on reducing sections to small sizes to reduce irrelevant features being created with sliding window feature generation. Their results in Hegel et. al. (2021) further justify this as a viable approach, as they find that model performance drops with increase in document length.

## Review Summary

| Author/Year | Research Focus | Dataset Used in Project | Findings and Relevance to project |
|---|---|---|---|
| Biagioli et. al. (2005) | Extracting semantic information from legal clauses. | 582 paragraphs from legal domain | Successfully used classifiers to provide solutions to problems within the legal domain. |
| Curtotti & McCreath (2011) | Phrase analysis of Australian legal corpus. | ~1,000,000 word, 256 Australian legal contracts | Conducted sentence analysis and identification of elements relevant to our project within contracts. |
| Chalkidis et. al. (2019) | Large-scale multi-label text classification in EU legislation. | EURLEX57K: 57,000 English EU legislative documents | Found success in applying BERT models in highly specialised tasks, justifying the use of BERT in our project. |
| Indukuri & Krishna (2010) | Extract clause information in e-contracts | <50 paragraphs of Loan and Security Agreement Contracts | Identified service exceptions in Loan and Security Agreements. Similar to our project but with narrowed focus. |
| Gao et. al. (2012) | Extracting service exceptions in e-contracts | 2647 business e-contracts sourced from Onecle | Identified payment related clauses in business payment contracts. Similar to our project but with narrowed focus. |
| Leivaditi et. al. (2020) | Locate "red flag" elements to identify risk bearing in lease contracts. | 179 annotated lease agreement documents | Utilised BERT model for highest performance with similar training parameters to Hendrycks et. al. (2021). Provides justification to training parameters used in our project. |
| Roegiest et. al. (2018) | Locating clauses and contract elements within a more varied set of legal contracts | 4412 manually annotated legal documents spanning 50 topics | Non-deep learning approach in automating contract review. Authors note deep learning is computationally expensive but implied to yield higher performance, justifying the use of BERT in our project. |
| Elwany et. al. (2019) | Fine-tune BERT on a large-scale legal contract corpus to improve detection of two specific agreement terms. | A few thousand legal agreements | Achieves strong results with their developed specialised BERT model. Highlights the strength of BERT models in specialised NLP tasks. |
| Chalkidis et. al. (2017) | Utilise simple ML classifiers to extract basic contract elements such as contract titles, parties and dates. | 3500 English legal contracts | Similar to the focus of Hendrycks et. al. (2021), however utilises a more manually-assisted approach to contract review. Offers insights into alternative approaches to ML contract review. |
| Hegel et. al. (2021) | Utilise OCR to improve the performance of CUAD. | CUAD: Over 500 contracts with 13,000 annotations | Similar to our project focus, set out using a different approach. Provides justification to the evaluation metrics utilised in our project. |
| Hendrycks et. al. (2021) | Provide a significant corpus of over 13,000 expert-annotated legal data of 41 contract elements in 510 documents. | CUAD: Over 500 contracts with 13,000 annotations | Introduced CUAD, which serves as the basis of our project. Our project aims to restructure CUAD to yield better performance in automating contract review. |

*Table 1: Summary of findings in related works and their relevance to project*

The subsequent component of the review details the literature of the key components utilised in the methodology section.

## 2.3 BERT Models

### Transformer Models

Attention mechanisms are a "method to encode sequence data based on the importance score each element is assigned" (Hu, 2019), and have been introduced into recurrent neural networks (RNNs) to mitigate the vanishing and exploding gradient issues which RNN models face. They are a significant component that have been used extensively in model architecture within the NLP domain.

The transformer model, first introduced in Vaswani et. al. (2017), is a network which consists entirely of attention mechanisms, dropping recurrence and convolutions which traditional sequence models were based on. This model yielded exceptional performance in both accuracy improvements and training time reduction compared to other state-of-the-art architecture and has become a staple model design choice for a broad range of NLP tasks including sentiment analysis and question answering (Vig, 2019).

### Bidirectional Encoder Representations from Transformers (BERT)

The BERT language representation model is a transformers-based model first introduced in Devlin et. al. (2018). The model is based heavily on the implementation described in Vaswani et. al. (2017), utilising bidirectional self-attention and pretrained on masked language modelling and next sentence prediction. This model has been designed to be easily fine-tuned for more specialised application, further evident in Devlin et. al. (2018) designing the model to handle both single sentence and paired sentence tasks for tasks such as question answering. The BERT model achieved strong performances across a set of 11 NLP tasks, notably becoming the highest performing model architecture (at the time of publishing) on the question answering benchmark SQuAD (Devlin et. al., 2018).

The RoBERTa model, produced in Liu et. al. (2019), expands on the work in Devlin et. al. (2018) by improving the BERT pretraining process. RoBERTa was trained for

longer with higher batch size and additional data, and importantly without the next sentence prediction as in the original BERT pretraining. Liu et. al. (2019) find performance improvements by dynamically changing the masking pattern in the masked language modelling task as well as training longer sequences. Additionally, they find a decrease in performance in adding next sentence prediction in contrast to the findings of Devlin et. al. (2018). The RoBERTa model significantly outperforms BERT and achieved comparable results to other state-of-the-art models including XLNet large when tested against SQuAD 1.0 and 2.0 (Liu et. al., 2019). This model will serve as the model to which we evaluate our improvements to CUAD within our project.

## 2.4 Question Answering

Question answering is a topic within NLP which focuses on developing a system which can receive a natural-language-style question input with some context, analyse and provide a response to the given question (Hirschman & Gaizauskas, 2001). The first known question answering program is BASEBALL (Green et. al., 1961), which answered questions specifically about baseball games within the context of one season of American League baseball. This program utilised linguistic knowledge to convert natural language questions into database queries which retrieved answers from a structured database. Modern question answering systems have evolved to broader tasks over a wider domain including specialised information retrieval as well as more general open-domain question answering, which focuses on large-scale unstructured documents (Zhu et. al., 2021). State-of-the-art NLP neural networks such as BERT based models perform well in modern question answering tasks (Devlin et. al., 2018; Liu et. al., 2019).

### Stanford Question Answering Dataset (SQuAD)

SQuAD was first produced in Rajpurkar et. al. (2016) as a reading comprehension dataset comprising over 100,000 question and answers. The data is collected from a set of Wikipedia articles and structured in a series of questions associated with a given

paragraph ‘context’. The corresponding answers to the questions are present as segments of the given context, and thus the task in locating answers given question and context is similar to ‘answer extraction’, which is the final step in the open-domain question answer pipeline (Rajpurkar et. al., 2016). This dataset has a human performance of 86.8% and has been deemed as a challenging dataset (Rajpurkar et. al., 2016) to evaluate the performance of question answer models, serving as a benchmark of comparison in many papers such as in Devlin et. al. (2018) and Liu et. al. (2019).

However, there has also been criticism to the SQuAD format. Weissenborn et. al. (2017) argue models may perform well against SQuAD without complex heuristics as a large portion of questions can be answered simply based on type and keyword-matching. Jia & Liang (2017) also challenge the use of SQuAD as an evaluation metric as introducing adversarial sentences into context passages results in sharp drop-offs in performance for many models, indicating a potential reliance on superficial cues without deep language understanding.

Rajpurkar et. al. (2018) attempt to remedy this with a revised SQuAD 2.0 dataset, which includes unanswerable questions which are relevant to the context. This is designed to improve evaluation on language understanding, with models performing significantly worse on SQuAD 2.0 when compared with SQuAD 1.0. SQuAD 2.0 has a very similar task focus to our project, where given a contract as context, there is no guarantee for all contract elements to be “answerable” in that they may not exist.

# 3. Methodology

## 3.1 CUAD

The Contract Understanding Atticus Dataset (CUAD) is the expert-annotated dataset produced in Hendrycks et. al. (2021). It consists of over 13,000 expert-annotated legal data of 41 contract elements in 510 documents. The contracts are structured in question answer format similarly to SQuAD, with each contract element (such as 'governing law', 'covenant to not sue') being phrased as a question and the context being the given contract. All 41 contract elements are converted into questions, and so if a contract element is not present within a document, the question would be considered unanswerable as similar to SQuAD 2.0 (Rajpurkar et. al., 2018). Hendrycks et. al. (2021) utilised fine-tuned BERT models to develop their automated contract review system, with training parameters of 4 epochs, learning rate of $1x10^{-4}$ and Adam optimiser as similar to standard BERT fine-tuning in other literature (Leivaditi et. al., 2020). However, their results were relatively poor, yielding an area under precision recall of 42.6% for their RoBERTa-base test, and 31.1% precision at 80% recall. Additionally, they were unable to yield results in their third evaluation metric, precision at 90% recall, for all models except DeBERTa-xlarge.

From the literature review conducted it is clear that the task of automating contract review over a broad scope strongly resembles that of answer extraction. With current BERT-based models performing rather well in general reading comprehension, applying question answering format to automating contract review is a novel concept with significant potential, despite poor initial performance from CUAD. Thus, as Hendrycks et. al. (2021) establishes a new benchmark for automated contract review, our project aims to provide insights into potential areas of improvement to the dataset and consequently to automating contract review.

Hendrycks et. al. (2021) found that quantity of training data had a significant impact on performance, and so it is possible that CUAD is hindered by insufficient data. This is further evident in the magnitude higher annotation count of similar datasets such as SQuAD 2.0 in Rajpurkar et. al. (2018). However, as annotating legal contracts

presented a high barrier of expertise, improvements to the dataset by means of increasing annotated contract count were unfeasible. Thus, our experimentation focused on improvements to performance by modifying the data structure instead.

Our project utilised CUAD as the dataset to modify. We utilised the same training and testing split as the work in Hendrycks et. al. (2021) to maintain a fair comparison. Due to low annotations of the contract element 'price restrictions', this category was excluded in evaluation in Hendrycks et. al. (2021). As our project utilised the same train test split, we similarly excluded the category for our evaluation.

## 3.2 Training Model

We recognised further extensions of the BERT model exist, including the DeBERTa model produced in He et. al. (2020), which produced the strongest performance results in Hendrycks et. al. (2021). However, due to memory and computational power limitations, experimentation using DeBERTa was unfeasible and thus our project utilised RoBERTa base instead. We found a similar trend in performance to all models with respect to AUPR and precision drop-offs in Hendrycks et. al. (2021), and so we assumed our changes to CUAD would similarly influence more complex models as it did for RoBERTa.

Therefore, RoBERTa-base was selected as the model to train with our modified datasets. The RoBERTa-base trained model produced by Hendrycks et. al. (2021) was utilised as our benchmark for comparison.

## 3.3 Training Environment

All experimentation was conducted in the following environment: QA features were generated on the University of Sydney Artemis High-Performance Computing Cluster. Model training and evaluation was conducted on Google Colab GPU runtimes with Colab Pro+ allocated RAM and GPUs. Model evaluation was separately validated on a Nvidia GeForce RTX 2070 GPU.

## 3.4 Experimentation Overview

As outlined in the introduction, the purpose of our experimentation was to provide two potential techniques in modifying CUAD to improve model performance. We measured the impacts of splitting CUAD by category into sub-groups of more balanced categories, as well as splitting the full contract into sub-contexts. Our focus was to assess these effects in isolation due to the potential confounding complexity of analysis when paired together.

Both techniques involved first conducting a processing technique to modify CUAD, then generating features from the modified dataset, before using the features as the input to train the RoBERTa-base model. A flowchart of our experimental process is presented below.

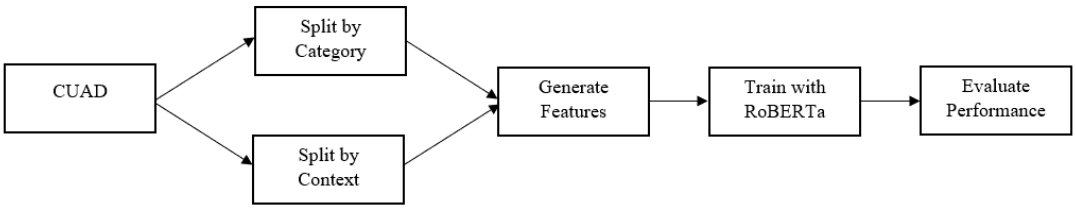



*Figure 1: Experimentation Overview*

The subsequent sections go into more detail about the methodology of each dataset modification technique.

## 3.5 Context Splitting

For our first approach, we opted to modify the method in Hendrycks et. al. (2021) using the entire contract as the context in their question answer data format. We found that using the entire contract as context likely reduced the importance of unanswerable questions. In the base CUAD format, if a contract element exists within the contract, there is only an answerable question of that category for the entire contract. This means the model would only learn to identify what a contract element looks like, but not where they don't exist. Using the same example with a split context, we introduced

unanswerable questions for every context which the contract element did not exist in, thus allowing the model to learn where the contract element does not appear.

For our context splits, we decided to split based on somewhat fixed character-size contexts. We chose to use 1000, 2000, 4000, 6000, 8000 and 10000 character-size contexts for our split size.

The decision to use character-size rather than tokens was made due to the base CUAD using character indices to mark the answer start. For convenience, we opted to preserve their answer start notation and so the only transformation for answer starts needed was to take the original answer start modulo the context size. For example, if a contract element originally had an answer start of 4952, this would be converted to 952 in our 1000-, 2000- and 4000-character context sizes within their relevant respective contexts.

Due to some answers potentially being cut off between contexts, we implemented a more flexible character-size splitting option. At the end of every context, if an answer was to be cut off, we simply extended the end of the context until the end of the answer, recursively checking if this new extension would again cut off an answer. This resulted in a slight variation in context sizes, but in general the context sizes remained similar.

While the context size splits were used to observe a general trend and thus somewhat arbitrary, we selected the size splits based on an analysis of the sizes of CUAD contracts. A general overview of the statistics is presented below for reference:

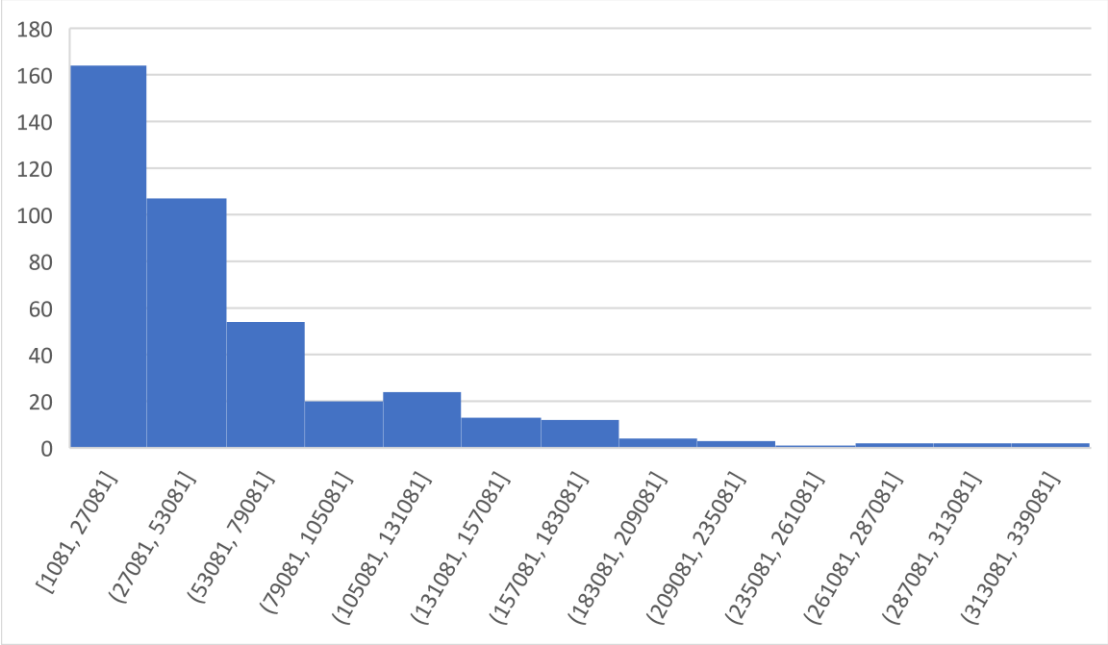



*Figure 2: Contract Size Distribution*

Of the 408 total contracts in the training set, over 66% of them are between 1000 to 50,000 characters large. The median contract length is 35,961 characters, with about a third of all contracts being less than 25,000 characters. Therefore, our splits ranging from 1000- to 10,000-characters generated a reasonable number of contexts.

Additionally, the table below is used as a reference to provide more meaning to the character sizes by indicating the relative size of each context split.

| **Context Size** | **Words (Approximate Range)** | **Number of Contexts produced in median contract** |
|---|---|---|
| 1000 | 160-200 | 36 |
| 2000 | 350-400 | 18 |
| 4000 | 700-750 | 9 |
| 6000 | 1000-1100 | 6 |
| 8000 | 1300-1400 | 5 |
| 10000 | 1600-1800 | 4 |

*Table 2: Reference Table Providing Meaning to Character-Size*

## 3.6 Category Splitting

We analysed the sample count of all categories within CUAD and identified a large imbalance in samples between the largest categories and the smallest. The table below highlights the discrepancy, with categories such as Parties dominating, having almost 100 times more samples than the smallest sized category.

| Grouping | Contract Element | Samples |
|---|---|---|
| Model 1 | Parties | 2554 |
| | License Grant | 716 |
| | Cap On Liability | 632 |
| | Audit Rights | 613 |
| | Anti-Assignment | 612 |
| | Insurance | 531 |
| | Document Name | 521 |
| Model 2 | Agreement Date | 475 |
| | Governing Law | 457 |
| | Expiration Date | 456 |
| | Effective Date | 412 |
| | Post-Termination Services | 409 |
| | Revenue/Profit Sharing | 395 |
| | Minimum Commitment | 392 |
| | Exclusivity | 388 |
| | Rofr/Rofo/Rofn | 350 |
| Model 3 | Ip Ownership Assignment | 296 |
| | Non-Transferable License | 282 |
| | Non-Compete | 226 |
| | Termination For Convenience | 223 |
| | Change Of Control | 213 |
| | Renewal Term | 199 |
| Model 4 | Volume Restriction | 165 |
| | Warranty Duration | 164 |
| | Covenant Not To Sue | 159 |
| | Uncapped Liability | 151 |
| | Irrevocable Or Perpetual License | 150 |
| | Notice Period To Terminate Renewal | 117 |
| | Competitive Restriction Exception | 114 |
| | Liquidated Damages | 113 |
| | Affiliate License-Licensee | 104 |
| | Joint Ip Ownership | 100 |
| | No-Solicit Of Employees | 75 |
| | Source Code Escrow | 63 |
| | Non-Disparagement | 54 |

| | | |
|---|---|---|
| **Model 5** | Affiliate License-Licensor | 47 |
| | No-Solicit Of Customers | 45 |
| | Third Party Beneficiary | 39 |
| | Most Favored Nation | 36 |
| | Unlimited/All-You-Can-Eat-License | 27 |
| **N/A (Excluded)** | Price Restrictions | 26 |

*Table 3: Total Samples per Contract Element*

Imbalanced multiclass classification is a challenging problem more difficult than imbalanced binary classification (Tanha et. al., 2020). Proposed techniques to address these issues include problem transformation, involving converting the main problem into many binary classification problems (Sahare, Gupta, 2012), and data approaches such as oversampling and undersampling. We found that binary transformation would entail a significant process both in pre-processing and training due to the significant number of categories in CUAD, and thus would be unfeasible for the scope of our project. Additionally, as seen in the above table, we found low sample size of the smallest categories and somewhat low sample size of all categories in general. Due to this, oversampling would likely result in model overfitting and undersampling would result in a significant loss of training data which as Hendrycks et. al. (2021) argues, results in magnitudes lower performance.

Therefore, our solution to mitigating class imbalance was to split the categories as separate models. While this may have cause issues with a reduced dataset size, in addition to balancing the classes, this option had additional benefits in potentially reducing the vast spread of contract elements and answer types of categories present within the dataset. Chalkidis et. al. (2021) similarly argued that the high difference in category types was a potential limitation of CUAD.

The table above represents the category groupings which we used to split the dataset and train separately. With the exception of Parties which is significantly larger than the rest, we chose these splits as each category grouping would be comprised of a more balanced set of elements, roughly within 100 samples of each other.

## 3.7 Feature Generation and Model Training

Each processed dataset was converted into features the same way as in Hendrycks et. al. (2021) using SQuAD's feature generation (HuggingFace, 2022). This involved using a sliding window technique of size 512 across the contexts at every 256 token intervals. We found that with the baseline CUAD, after feature generation Hendrycks et. al. (2021) removed many negative samples to mitigate the negative to positive sample imbalance. We conducted this same balancing to remove redundant windows with no question answer associated to it.

Datasets were trained separately using the RoBERTa base model with the same training parameters as Hendrycks et. al. (2021). This included 4 training epochs with the Adam optimiser and using a learning rate of $1x10^{-4}$.

# 4. Results and Discussion

## 4.1 Evaluation Metrics

For evaluation, we utilised the same metrics as Hendrycks et. al. (2021) in order to develop a performance comparison with the benchmark. This included the Area Under Precision Recall (AUPR) and Precision at % Recall, which are fairer to data imbalance. In addition, we included F1-score due to it being a prevalent metric of choice while also responsive to imbalanced classes, as well as positive and negative sample F1-score as in the Hugging Face SQuAD 2.0 evaluation metrics (HuggingFace, 2022).

**Jaccard Similarity Index**

We evaluated correct matches in the same way as Hendrycks et. al. (2021) to maintain a fair comparison. Thus, we utilised the Jaccard similarity as a threshold to determine matches. The Jaccard similarity is measured by $\mathrm{J}(\mathrm{A},\mathrm{B}) = \frac{|\mathrm{A}\cap\mathrm{B}|}{|\mathrm{A}\cup\mathrm{B}|}$, where $\mathrm{J}(\mathrm{A},\mathrm{B})$ represents the Jaccard index and A, B represent the ground truth and predicted text respectively. The threshold used is set to 0.5, which can be interpreted that at least half of the predicted text overlaps with the ground truth.

**AUPR**

AUPR is evaluated by calculating the area under the curve bound by precision and recall at various confidence thresholds. We took precision recalls at every percentage change in confidence threshold from 99% to 0 with an additional one at 0.1% as per Hendrycks et. al. (2021). This yielded 100 data points to which we calculated the area under the curve using the trapezoidal rule, providing a reasonable estimate of the true AUPR.

**Precision at % Recall**

Precision is defined as the fraction of true positives over the total predicted positives and thus represents the percentage of correctly predicted positives out of all the positives predicted. Recall is defined as the fraction of true positives over the total positives present, and thus represents the portion of correct positive answers predicted out of all correct positive answers. Thus, to take a precision at a % recall, we took a confidence threshold to which the model is predicted a high rate of correct positives, then determined how many of its positive predictions were accurate (to determine whether or not false positives were being generated).

This measurement has been utilised in similar NLP tasks (Leivaditi et. al., 2020; Hegel et. al., 2021). Hendrycks et. al. (2021) presented the practical application of CUAD in contract review as finding "needles in a haystack", and thus high recall would be required. 80% and 90% were cited as the metrics, however we could not find justification on the choice of percentage.

A recall of 80% implies that 20% of the positive samples were missed and so lawyers would likely still be required to manually comb through contracts if they wished to find the remaining elements. Therefore, in practice, we found that such a metric is rather arbitrary. Taking a precision at 70% recall would result in only 10% additional positive samples being missed which would be similarly time consuming to comb over. As all models including the RoBERTa-base model in Hendrycks et. al. (2021) were not able to reach 90% recall, we excluded the metric precision at 90% recall in favour of precision at 70% recall.

**F1-Score**

Additionally, we included F1-score as it is a well-recognised metric of model evaluation. All previous metrics focused on the performance in detecting positive samples and so we included both the sample-specific (positive and negative) F1-Score and combined weighted F1-Score as calculated in the SQuAD evaluation metrics in Rajpurkar et. al. (2019).

## 4.2 Split Context Tests

| Subsection Size (characters) | F1-Score | F1-Score (Positive Sample) | F1-Score (Negative Sample) | AUPR | Precision at 80% Recall | Precision at 70% Recall |
|---|---|---|---|---|---|---|
| Benchmark (CUAD Base) | 0.751 | 0.836 | 0.716 | 0.426 | 0.311 | 0.393 |
| 1000-character | 0.947 | 0.865 | 0.971 | 0.578 | 0.000 | 0.000 |
| 2000-character | 0.938 | 0.848 | 0.965 | 0.576 | 0.000 | 0.384 |
| 4000-character | 0.916 | 0.805 | 0.950 | 0.529 | 0.000 | 0.454 |
| 6000-character | 0.912 | 0.839 | 0.934 | 0.549 | 0.000 | 0.564 |
| 8000-character | 0.903 | 0.849 | 0.919 | 0.522 | 0.000 | 0.543 |
| 10000-character | 0.891 | 0.849 | 0.904 | 0.513 | 0.000 | 0.539 |

*Table 4: Model Performance for Split Context Tests*

The above table provides a succinct table summary for all model performances produced from context splitting.

Overall, we found significant benefit in splitting the context into smaller sections when compared with the benchmark. There is a clear performance gain across all metrics except Precision at 80% Recall, which are 0 due to the models never reaching a recall of 80%. However, lowering the recall threshold by 10% (to Precision at 70% Recall) yields strong performances for almost all tests in comparison to the benchmark. Reasons as to why split-context models do not achieve 80% recall are explained in the Precision Recall Trade-Off section.

The following sections provide our interpretation to the trends across the experiments. We argue that the 6000-character context sized dataset produced the best performing model when evaluated against its purpose of automating contract review.

**F1 Score**

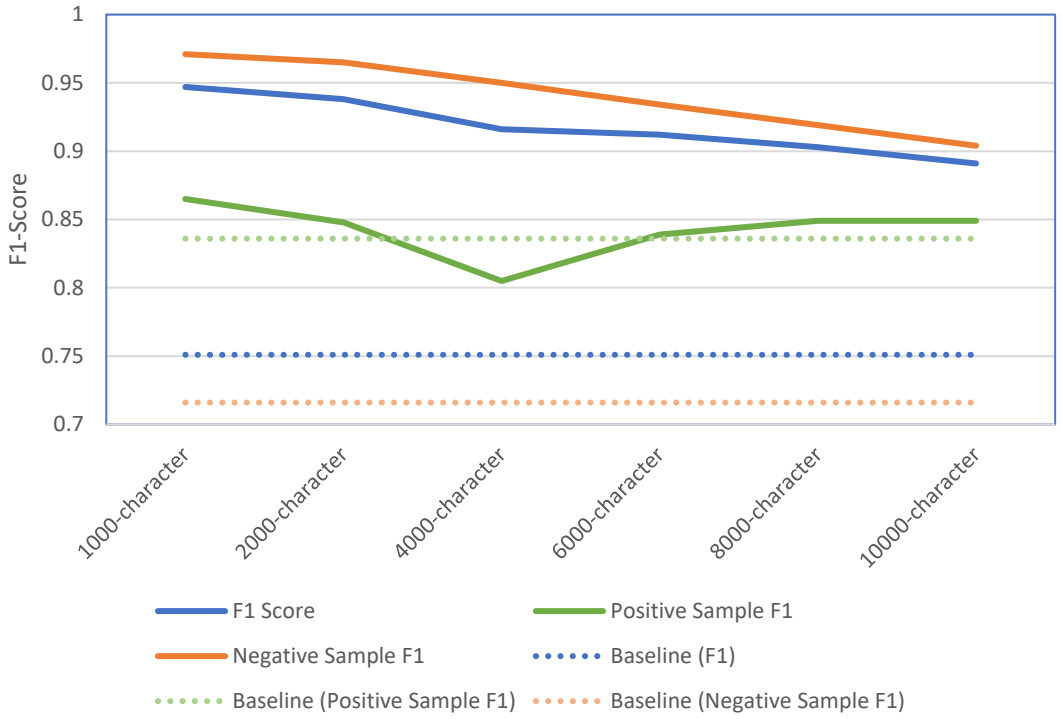



*Figure 3: Split Context Test F1-Scores with Benchmark Comparison*

From the above visualisation of F1 Scores, we see that the overall F1 score is negatively correlated to context size. The best F1 score was obtained in the model with the smallest context size. This appeared to be influenced primarily by the negative sample F1, which the overall F1 follows closely.

The benchmark model performed the worst in terms of negative sample F1, which lowered its overall performance. All split-context models have higher negative F1 than positive F1, however the reverse is true for the benchmark. This observation was interesting and not immediately intuitive. We expected the baseline to have a higher success predicting negative samples due to the data imbalance and therefore have a higher negative F1-score.

From this we inferred either or both of two conditions. Firstly, as context size increased, the models missed more positive samples, thus incorrectly labelling positive samples as negative. Secondly, as context size increased, the models generated more false positives, thus labelling more negative samples as positive. Both conditions would result in a lower than expected negative F1-score for the benchmark when compared

with the split-context tests. However, the level of influence of either factor cannot be determined from this metric alone. Thus, we additionally assessed the precision and recall to determine this.

**Precision Recall Trade-off**

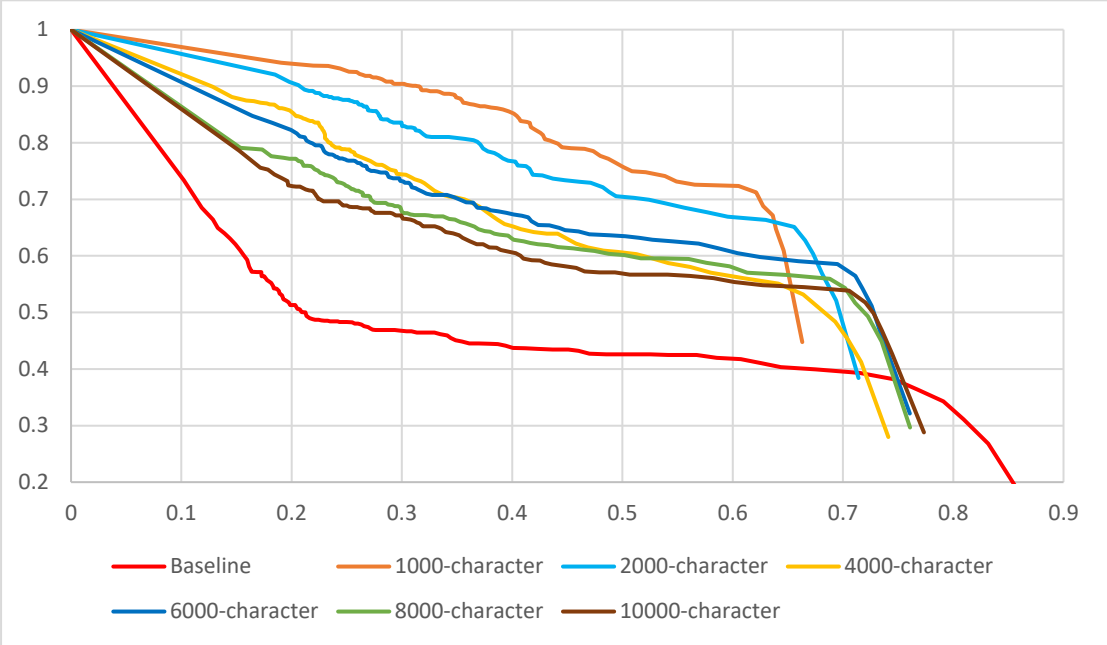



*Figure 4: Precision Recall Curves of Split Context Tests with Comparison to Benchmark*

As presented the precision recall curves above, we found a general trend that as context size decreased, the model typically traded higher precision values for an earlier recall drop-off. Maximum recall improves and precision decreases with an increase in context size. From this, in addition to the decrease in negative sample F1, we infer that the models must be generating more positive predictions, resulting in additional false positives.

This is identified by analysing the definition of precision and recall:

$$Precision = \frac{TP}{TP + FP} \text{ and } Recall = \frac{TP}{TP + FN}$$

As precision is decreasing, the ratio of true positives to false positives is diminishing, indicating more false positives being generated. Recall increases as context size

increases, and thus a lesser portion of false negatives are being generated. This means that as context size increases, models create more positive predictions overall, resulting in a higher recall due to more correct positive predictions. However, this also results in lower precision as the model becomes less accurate and overpredicts more. This presented a clear example of the trade-off between precision and recall.

From the graph we can interpolate that as the model trends towards the full-size context, the models trade further recall improvements at the cost of precision until the benchmark precision-recall curve is reached. This provides an explanation for split-context models not reaching 80% precision, and also an explanation for the reasons as to why negative F1-score is comparatively lower than expected. First, lower context-size models peak higher in terms of recall at the cost of a lower maximum recall, with most of our test models reaching a maximum recall between 70 to 80%. Secondly, the reduced precision as context-size increases indicates additional false positives being generated, which lowers negative F1-score. Lower context-size models have higher precision and thus less false positives, which contributes to a higher negative F1-score.

Therefore, from the overall model performance we find that splitting the context into smaller sections results in a more effective precision gain to recall loss as a result of the model predicting more tightly. This results in a slight reduction in recall as not as many true predictions are found, but significantly reducing the number of false positives generated in comparison to the benchmark.

**Individual Category Performance**

One explanation of this trend in precision-recall trade-off is that shorter contexts result in better learning of syntactic and language, which could present strong indication of what contract elements exist in a span of text, thus improving precision for predicting certain contract elements. We found this to be the case when analysing performance of individual categories below. Additionally, this is supported by the literature in Weissenborn et. al. (2017), who suggest that many question answering tasks may be solved by language heuristics, which could also extend to specialised domain question answering as in CUAD. The shorter contexts also result in loss of broader information

including location-specific information, such as certain contract elements being likely to occur in a particular location of the full contract.

This is also suggested based on the findings of individual category performance, presented on the following page. Notably, the “Parties” and “Document Name” are typically always included within the first page of every contract. Splitting the context results in the loss of this information, which could explain performance decreases in these categories in comparison to the base CUAD benchmark. This is further supported by the findings in Hegel et. al. (2021), who find that some layout features can hurt precision when contracts are split up as they mislead the model when identifying header information.

We opted to present the individual category performances of the 1000-character context and the 6000-character context as we believe they are the best performing models; however, the full category performances of all models are detailed in the appendix.

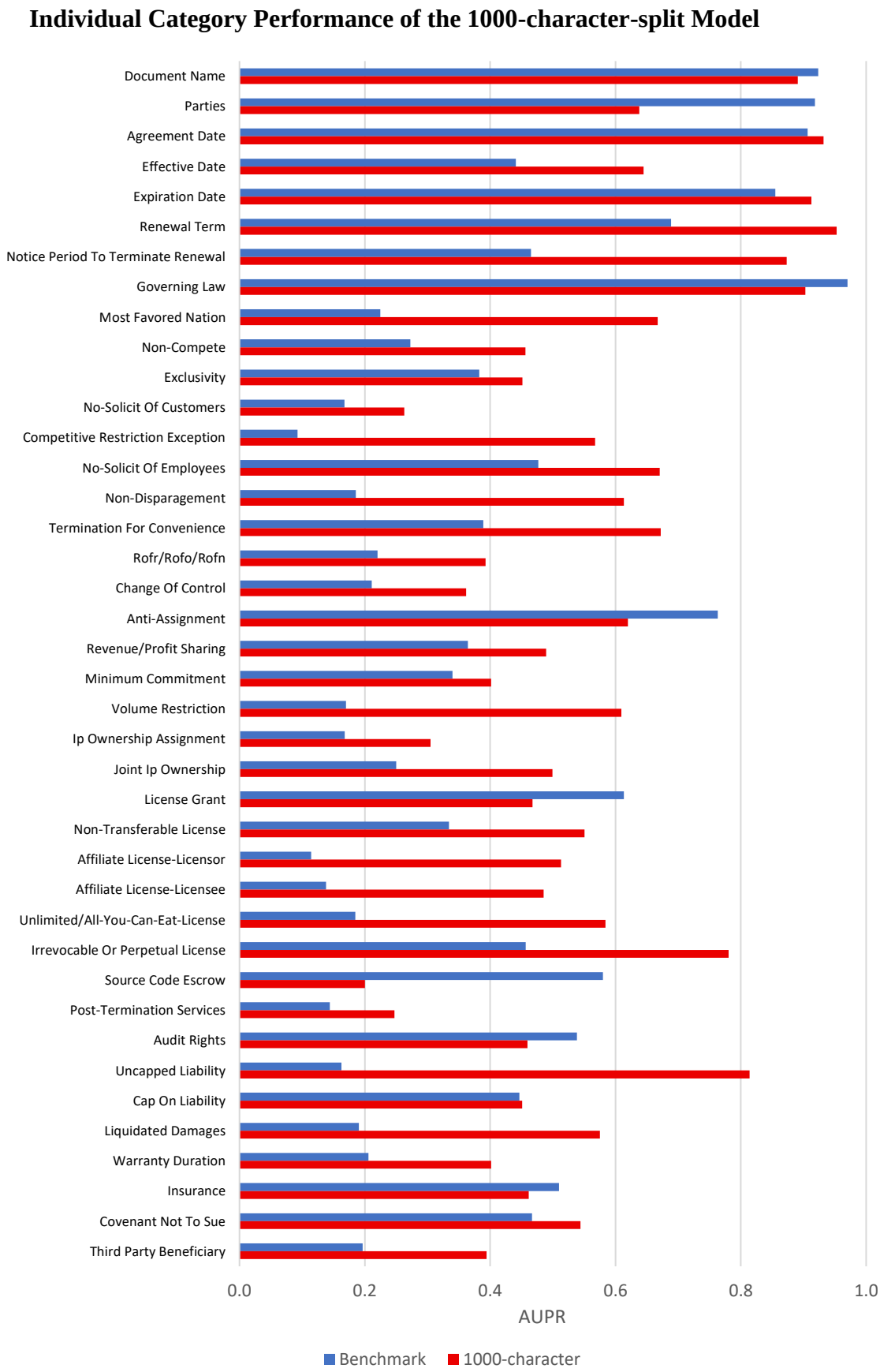



*Figure 5: 1000-character Split Context Model Comparison with Benchmark*

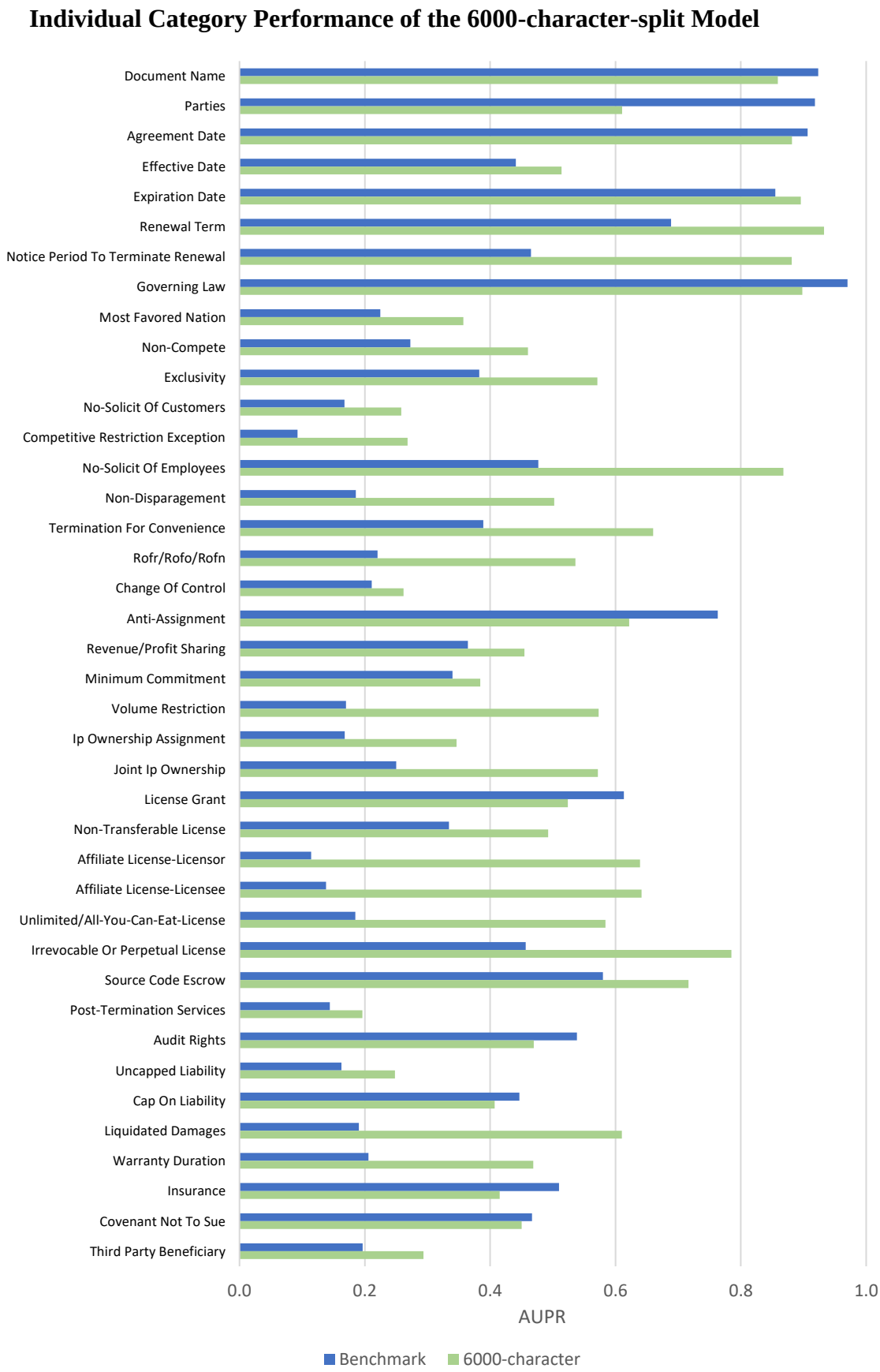



*Figure 6: 6000-character Split Context Model Comparison with Benchmark*

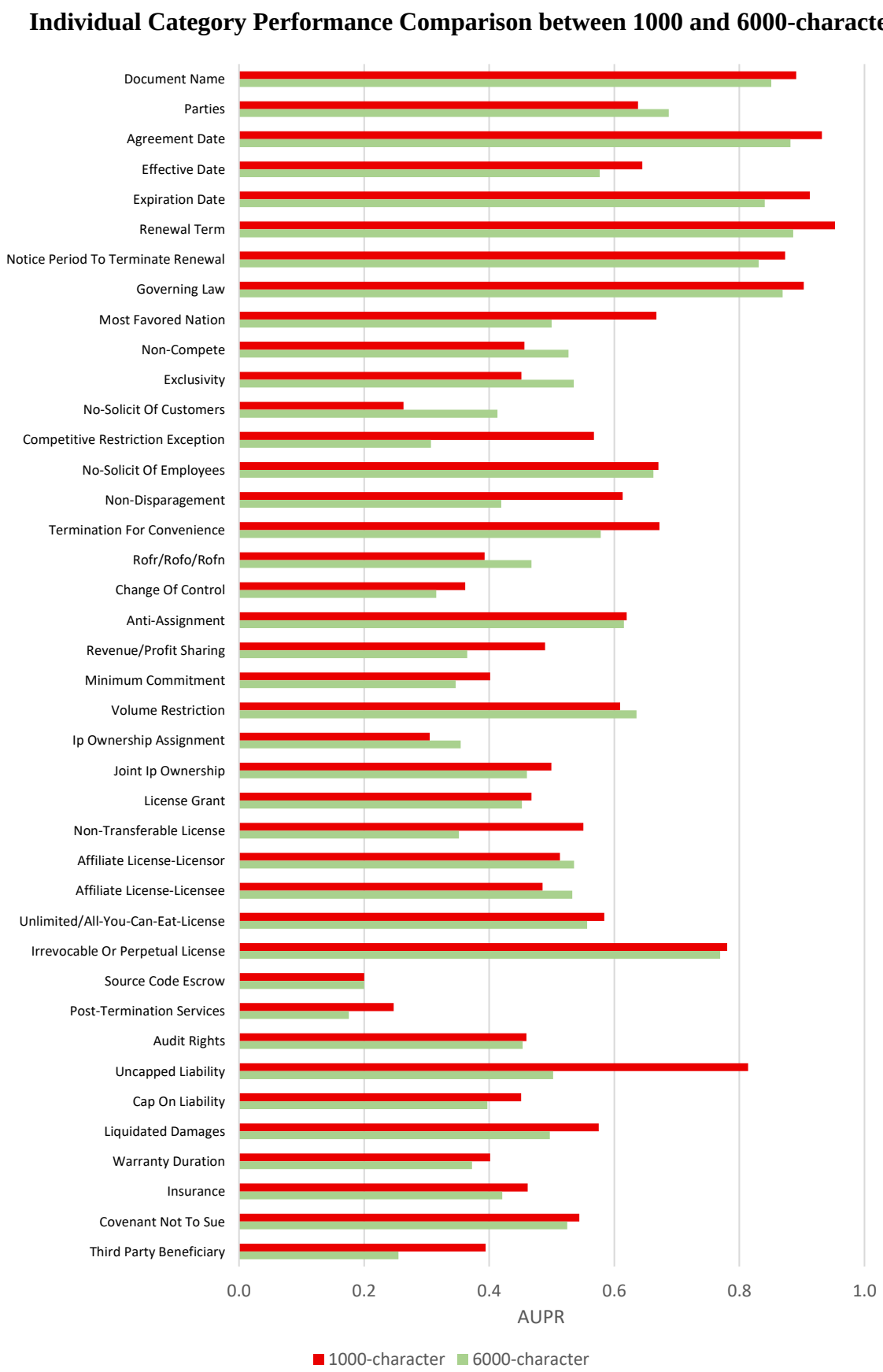



*Figure 7: Performance Comparison between 1000 and 6000-character Split Context Models*

**Improvements Across Categories in Comparison to Benchmark - AUPR**

From the individual category analysis, we see that when compared to the benchmark, a majority of individual categories saw performance gains with split contexts, with the exception of elements such as Parties and Document Name as explained above. However, as the Parties and Document Name elements dominate the positive samples, we saw lower improvements than would be expected from the overall AUPR results shown below. The AUPR of the benchmark was 0.426, which is derived from the total positive predictions across all categories. However, if equal weighting is assigned to all categories, then taking the arithmetic average of individual category AUPR yields an average performance of 0.403. In contrast, the 6000-character model saw an increase, from a total AUPR of 0.549 to an average category AUPR of 0.553. This implies that there was an even higher performance improvement across lower sample categories.

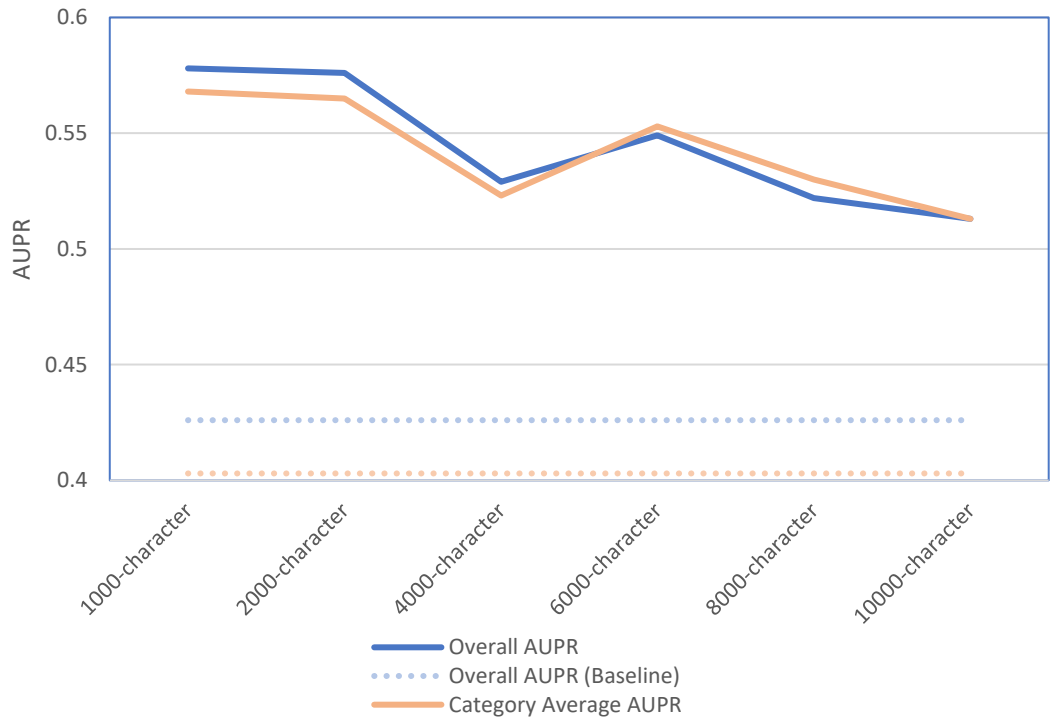



*Figure 8: Split Context Test AUPR with Benchmark Comparison*

When comparing between the split context tests, we found that the low context size models outperform higher context sizes with respect to AUPR, caused by higher precision across lower recall values, despite peaking at a lower maximum recall.

**Precision at 70% Recall**

However, when evaluating by precision at 70% recall, we find that the 6000-character context size produces the highest precision for the given recall.

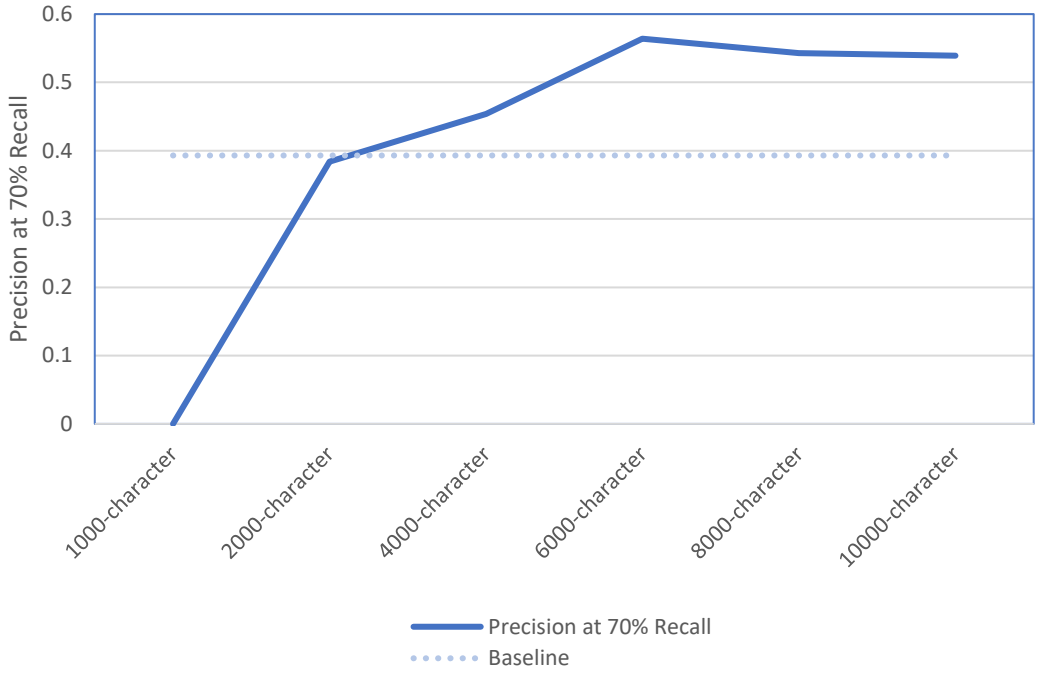



*Figure 9: Split Context Test Precision at 70% Recall with Benchmark Comparison*

We found that as context size increases, number of predictions increases. While this results in higher recall as some of these additional predictions are correct, this also lowers precision as there are more false positives. As we prioritise recall over precision, we find that the 6000-character context size maximises the precision-recall trade-off whilst maintaining relatively high precision. Beyond this point, we found that recall gain is marginal whilst precision loss still occurs, thus a decrease in AUPR past 6000 occurs. As the context size tends toward the size of full contracts, the AUPR is likely to continue downwards until the benchmark precision and recall is reached. Therefore, the 6000-character context size model produces the highest precision with an acceptable 70% recall.

Additionally, Hendrycks et. al. (2021) deemed the following 6 categories to be of highest importance:

| Category | Benchmark Precision @ 70% Recall | 6000-character context size Precision @ 70% Recall | Change |
|---|---|---|---|
| Effective Date | 0.424 | 0.509 | +8.5% |
| Renewal Term | 0.565 | 0.929 | +36.4% |
| Anti-Assignment | 0.851 | 0.667 | -18.4% |
| Governing Law | 0.975 | 0.973 | -0.2% |
| Irrevocable or Perpetual License | 0.400 | 1.000 | +60.0% |
| Non-Disparagement | 0.109 | 0.391 | +28.2% |

*Table 5: Comparison of Split Context Test to Benchmark in the 6 Most Important Categories*

We found significant improvements in Precision at 70% Recall from Renewal Term, Perpetual License and Non-Disparagement, with Governing Law remaining almost the same, a decrease in performance in Anti-Assignment and a slight improvement to Effective Date. Importantly, the largest improvements were observed in the lowest performing categories, thus bringing the weakest performing categories to a decent level.

The benchmark model significantly overpredicts with a low confidence threshold, resulting in high recall but low precision as there are many false positives. We believe this is attributed to the following reason. Full-context contracts likely contain many of the contract elements, however due to the large volume of empty sliding windows generated from full contracts being trimmed, only the samples with questions are kept. This results in the benchmark model only learning what positive samples look like, and not what they don't look like, thus resulting in excess amount of false positive predictions as it finds clauses that are similar to a contract element but are not. Thus, by splitting the contract and generating impossible questions for elements that do not exist in a sub-context, the models become better at identifying if an element does not exist, hence reducing the false positive count and improving precision.

However, the reverse appears to be the case with Anti-Assignment. The split context model generates more false positives as opposed to the Benchmark. One possible explanation of this is that anti-assignments are similar syntactically to other clauses, however, are distinguishable from a whole-contract perspective, such as always

presenting in a particular section of a contract. We find evidence to support this in analysing some predictions and false positives being generated by the 6000-character model in the Anti-Assignment category:

| **Ground Truth** | **Predicted Sample** |
|---|---|
| This Agreement may not be assigned without the prior written consent of the other Party hereto. | This Agreement may not be assigned without the prior written consent of the other Party hereto. |
| This Agreement or any rights or obligations granted hereunder may not be assigned by ABW without the prior written consent of PCQ | This Agreement or any rights or obligations granted hereunder may not be assigned by ABW without the prior written consent of PCQ |
| Neither party shall assign, transfer, or subcontract this Agreement or any of its obligations hereunder without the other party's express, prior written consent, which will not be unreasonably withheld | Neither party shall assign, transfer, or subcontract this Agreement or any of its obligations hereunder without the other party's express, prior written consent, which will not be unreasonably withheld |
| N/A (False Positive) | In sum, Minn. Stat. §80C.14 (subd. 5) currently requires that consent to the transfer of the franchise may not be unreasonably withheld |
| N/A (False Positive) | Neither party shall, during the term of this Agreement and for one (1) year after its termination, solicit for hire as an employee, consultant or otherwise any of the other party's personnel who have had direct involvement with the Services, without such other party's express written consent, which shall not be unreasonably withheld |

*Table 6: Example Predictions of Anti-Assignment Generated by Split Context Models*

The first 3 predictions are examples of correct matches, and the following 2 are examples of false positives. We find similarities in clause structure and keywords, however the false positives generated could potentially be caused by lack of contract-wide context which result in location specific information being lost.

## Practical Application

The RoBERTa-base model in Hendrycks et. al. (2021) produced approximately 30% correct predictions with a recall of 80%. Thus, a practical application of the model would mean lawyers would have to review 2 incorrect clauses for every one correct, in a set of predictions that cover 80% of all correctly labelled clauses. If lowering the threshold to cover 70% of all correctly labelled clauses, their model yields 39.3%

precision, which means that they would be reviewing 3 incorrect clauses for every 2 that are correct. Our models yield an accuracy of over 56% precision at this threshold, which is significantly higher accuracy than the CUAD base. Thus, at the cost of reviewing contracts for an additional 10% of elements, our model would improve the accuracy of predictions by over 15%. For the most important elements, this is even higher of an improvement. Thus, we see a clear, practical improvement to splitting the contract into smaller contexts.

## 4.3 Split Category Models

As discussed in the methodology, we divided the question answers into 5 separate categories by sample count. The results summary is presented below.

| Model | Categories | F1-Score | Positive F1 | Negative F1 | AUPR | Precision @ 80% Recall | Precision @ 70% Recall |
|---|---|---|---|---|---|---|---|
| Model 1 | Parties<br>License Grant<br>Cap on Liability<br>Audit Rights<br>Anti-Assignment<br>Insurance<br>Document Name | 0.823 | 0.847 | 0.785 | 0.661 | 0.679 | 0.706 |
| Model 2 | Agreement Date<br>Governing Law<br>Expiration Date<br>Effective Date<br>Post-Termination Services<br>Revenue/Profit Sharing<br>Minimum Commitment<br>Exclusivity<br>Rofr/Rofo/Rofn | 0.778 | 0.852 | 0.701 | 0.565 | 0.255 | 0.519 |
| Model 3 | Ip Ownership Assignment<br>Non-Transferable License<br>Non-Compete<br>Termination For Convenience<br>Change of Control<br>Renewal Term | 0.675 | 0.806 | 0.633 | 0.339 | 0.101 | 0.216 |
| Model 4 | Notice Period to Terminate Renewal<br>Competitive Restriction Exception<br>Volume Restriction<br>Joint Ip Ownership<br>Affiliate License-Licensee<br>Irrevocable Or Perpetual License<br>Uncapped Liability<br>Liquidated Damages<br>Warranty Duration<br>Covenant Not To Sue | 0.656 | 0.776 | 0.637 | 0.244 | 0.093 | 0.131 |
| Model 5 | No-Solicit of Employees<br>Source Code Escrow<br>Non-Disparagement<br>Affiliate License-Licensor<br>No-Solicit of Customers<br>Third Party Beneficiary<br>Most Favored Nation<br>Unlimited/All-You-Can-Eat-License | 0.620 | 0.372 | 0.620 | 0.082 | 0.023 | 0.059 |

*Table 7: Results Summary of Split Category Model Tests*

The results of the model performances suggest that low sample size has a negative impact on performance. Notably, the model containing the lowest sample elements performed very poorly in comparison to the other models, even when adjusting for performance of its respective categories in the benchmark. We found that the lower sample models produced more false positives as indicated by the lower negative sample F1. This can be explained by the lack of samples resulting in low certainty of predictions, thus a high number of incorrect predictions, or false positives, being generated.

**Individual Category Performances**

While the overall model performance provides some insights, each model is only a subsection of the entire problem. Thus, to evaluate the performance of splitting categories, we collectively compare the individual category performances of these models by their respective performances in the CUAD benchmark. The results are presented on the following page.

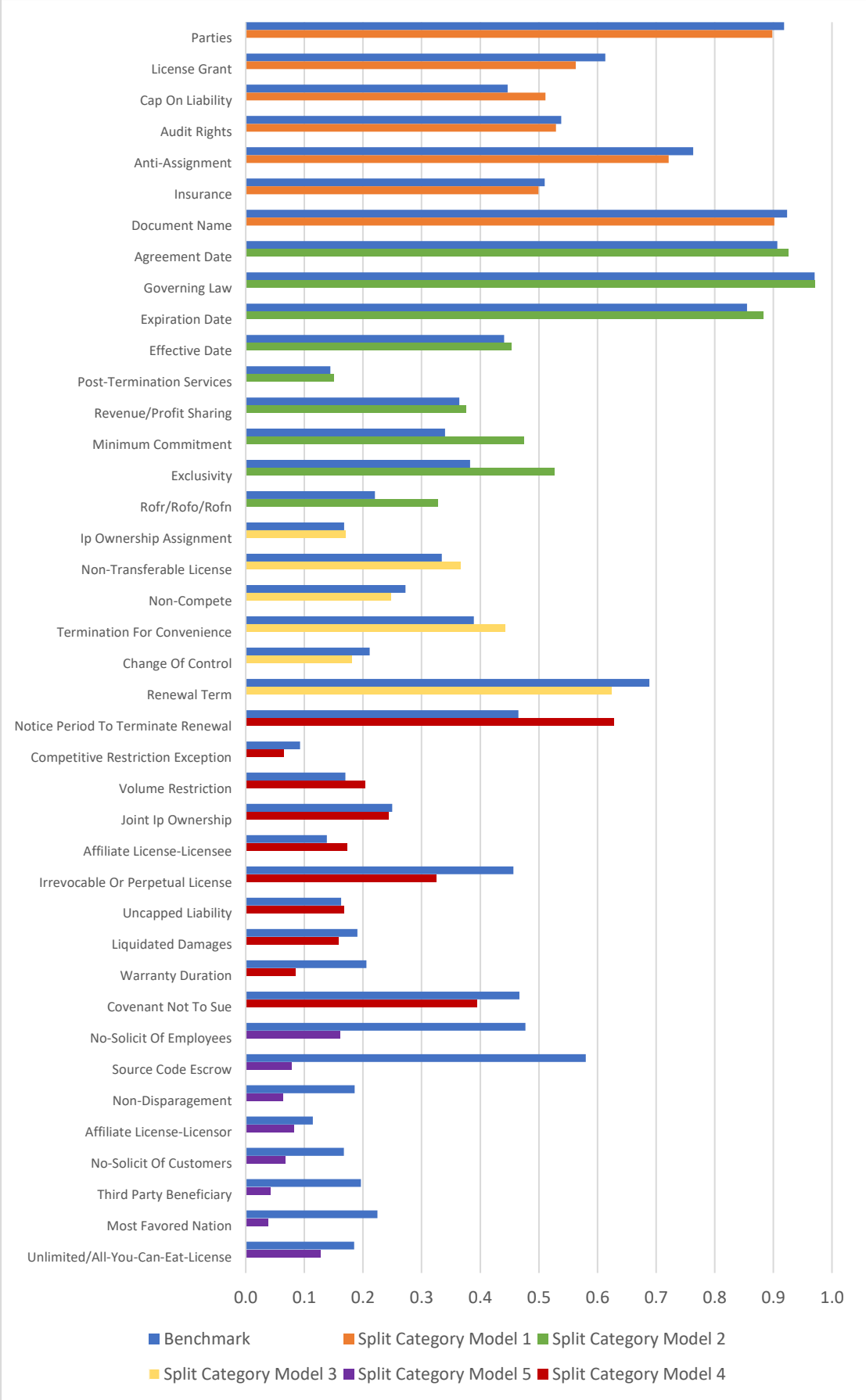



*Figure 10: Individual Category AUPR of Split Category Models with Comparison to Benchmark*

From the individual category performances, we find a more mixed performance across categories in comparison to the benchmark. Of 40 categories, we found improvements to 16 categories. However, if excluding the results of the lowest performing model consisting of 8 categories, we find improvements to exactly 50% of categories. On average, we find a marginal improvement to average category performance when excluding the results of model 5.

| | **Average Category AUPR** | **Average Category AUPR Excl. Model 5** |
|---|---|---|
| **Split Category Models** | 0.371 | 0.443 |
| **Benchmark** | 0.403 | 0.438 |

*Table 8: Average Category AUPR of Split Category Models with Benchmark Comparison*

From these findings we draw the following observations.

Firstly, the results are contrary to our expectations. The intention of splitting categories by sample size was to reduce the bias between predicting incorrect classes. However, these results do not support this hypothesis. One explanation is that the performance loss as a result of reduction in total samples far outweighs the benefits of balancing samples. These results do support the findings in Hendrycks et. al. (2021), who find that dataset size has a significant impact on model accuracy. Thus, our results suggest that any additional samples may help improve model performance, regardless of whether it causes sample bias. Conversely, reducing total samples decreases model performance more than any benefits gained from balancing samples. This explains the poor performance of the lower sample models. For reference, the lowest sample models have a combined sample total of less than the samples in the most and second most sample models.

Secondly, the performance of model 2 (second largest sample size group) shows by far the most improvements across categories. All categories yield improvements in comparison to the benchmark except for Governing Law, which performs equally. As this result is somewhat of an outlier in comparison to the other trained models, it is difficult to ascertain the exact cause of this performance. One explanation is that the grouped categories in model 2 result in more optimised learning for the model, possibly due to them being somewhat related. Finding such groupings to optimise CUAD could

be a further avenue of follow-up investigation, however this would involve a deeper domain knowledge and thus was not considered for this project.

Lastly, some low sample categories perform well in benchmark and split context tests, but low when trained with other low sample categories. This could potentially be related to our second observation, in which we suggested models performed better with certain category groupings due to some overlap in information contained across category elements. Thus, higher sample categories could help the model learn characteristics of lower sample categories and result in synergistic performance improvements. For example, Source Code Escrow has one of the lowest sample sizes, but yields an AUPR of 0.580 in the benchmark model and 0.717 in the 6000-character split context model. When trained with other low sample categories, it yields an extremely poor AUPR of 0.079. Warranty Duration is another low-sample category displaying this observation. The benchmark yields an AUPR of 0.206, with the 6000-character split context model yielding 0.469. The split category model yields 0.085. The following tables present examples of correctly matched predictions from the two categories, as well as incorrect predictions generated by the split category model.

| **Warranty Duration Predictions** | | |
|---|---|---|
| **Ground Truth** | **Prediction** | **Actual Category of Prediction** |
| Products repaired or replaced by Philips within the Warranty Term are warranted for the remainder of the original Warranty Term of said Products. | Products repaired or replaced by Philips within the Warranty Term are warranted for the remainder of the original Warranty Term of said Products. | Warranty Duration |
| The Customer shall inspect all Products promptly upon receipt thereof and may reject any defective Product, provided that the Customer shall within seven (7) days after receipt of such alleged defective Product, notify the Manufacturer of its rejection and either: (i) request to destroy in field for credit of the value of the defective product and the associated shipping costs (with approval), or (ii) request a Return Material Authorization ("RMA") number and within seven (7) days of receipt of the RMA number from the | The Customer shall inspect all Products promptly upon receipt thereof and may reject any defective Product, provided that the Customer shall within seven (7) days after receipt of such alleged defective Product, notify the Manufacturer of its rejection and either: (i) request to destroy in field for credit of the value of the defective product and the associated shipping costs (with approval), or (ii) request a Return Material Authorization ("RMA") number and within seven (7) days of receipt of the RMA number from the | Warranty Duration |

| | | |
|---|---|---|
| Manufacturer return such rejected Product to the Manufacturer. | Manufacturer return such rejected Product to the Manufacturer. | |
| N/A (False Positive) | Notwithstanding the foregoing, if the Agreement is terminated for any reason other than Berkshire's material breach, then for thirty (30) days following such termination, Berkshire may continue to use any printed material already produced under this Agreement. | Post-Termination Services |
| N/A (False Positive) | The Professional hereby grants to the Company the worldwide right during the Term and for a period of six (6) months after the Term as provided in Section 2.8 to use, reproduce, print, publish, distribute, broadcast, modify, edit, condense, or expand any materials containing her name, image, likeness or personal identification that are created hereunder. | License Grant |
| N/A (False Positive) | M&I may, at any time, withdraw any of the Services (other than the Core Services) upon providing ninety (90) days' prior written notice to Customer. | Termination For Convenience |
| N/A (False Positive) | By any of the PHL Parties, on the one hand, or ICC, on the other, providing one hundred and twenty (120) days prior written notice to the other Parties. | Termination For Convenience |

*Table 9: Examples of Warranty Duration Predictions for Split Category Models*

| **Source Code Escrow Predictions** | | |
|---|---|---|
| **Ground Truth** | **Prediction** | **Actual Category of Prediction** |
| In the event Customer obtains a copy of the source code pursuant to Section 23.4 above, Customer (or its designee) shall use the source code during the term of the license granted herein<br>solely for Customer's own internal processing and computing needs and to process the Customer Data, but shall not (1) distribute, sell, transfer, assign or sublicense the source code or any parts thereof to any third party, (2) use the source code in any manner to provide service bureau, time sharing or other | In the event Customer obtains a copy of the source code pursuant to Section 23.4 above, Customer (or its designee) shall use the source code during the term of the license granted herein<br>solely for Customer's own internal processing and computing needs and to process the Customer Data, but shall not (1) distribute, sell, transfer, assign or sublicense the source code or any parts thereof to any third party, (2) use the source code in any manner to provide service bureau, time sharing or other | Source Code Escrow |

| | | |
|---|---|---|
| computer services to third parties, or (3) use any portion of the source code to process data under any application or functionality other than those applications or functionalities which were being provided by M&I to Customer at the time Customer became entitled to receive a copy of the source code. | computer services to third parties, or (3) use any portion of the source code to process data under any application or functionality other than those applications or functionalities which were being provided by M&I to Customer at the time Customer became entitled to receive a copy of the source code. | |
| N/A (False Positive) | The parties acknowledge and agree that there are no third party beneficiaries to this Agreement. | Third Party Beneficiary |
| N/A (False Positive) | Airspan warrants that during the term of this Agreement, the prices at which Airspan sells to Distributor products supplied under this Agreement shall be no less favorable to the Distributor than those prices at which Airspan sells, at substantially the same time in the United States, similar products and pursuant to similar terms and conditions as those by which Airspan sells Products to the Distributor under this Agreement. | Most Favoured Nation |
| N/A (False Positive) | knowingly and intentionally induce, solicit, or encourage PHL GIE Persons to terminate their respective contracts, or otherwise change their relationship, with any of the PHL Parties or their Affiliates; or 8.07.1.3 without the prior written consent of the PHL Parties, employ or otherwise contract with any PHL GIE Persons. | No-Solicit of Employees |

*Table 10: Examples of Source Code Escrow Predictions for Split Category Models*

We found that these categories performed worse due to the model predicting false positives of clauses from other categories. For example, Warranty Duration tended to generate false negatives focusing on Termination, as well as larger sample categories such as License Grant. These categories were not included within the model which Warranty Duration was in, and thus we see a potential benefit in training larger sample categories with lower sample categories. In the benchmark and split context models, the addition of those categories could allow the model to better differentiate between

Warranty Duration and the Termination or License Grant categories, thus reducing false positives being generated for the category. Source Code Escrow false positives tended to be other categories present within the lowest-sample group, such as No-Solicit of Employees. This could be due these categories being too small to learn meaningful information to distinguish between categories.

These observations further support the conclusion that increasing samples improves accuracy by more than the imbalance it causes. Additionally, this opens an additional avenue of investigation to attempt to identify and split categories to maximise synergistic gains in training some low-sample categories with higher-sample ones. However, this task would entail a significant scope and thus was not explored within our project.

# 5. Conclusions

In this project, we addressed potential issues of structure and data imbalance within the original CUAD dataset by Hendrycks et. al. (2021) and tested two potential approaches to improve performance by mitigating these issues. CUAD serves as a strong baseline for question answer style approaches to automated contract review, and our project aims to further enhance the approach in Hendrycks et. al. (2021). We investigate the effects of splitting the full contract context into smaller sub-contexts and splitting the dataset categories by sample size into different smaller datasets to balance the samples.

We found significant improvements to average model performance in split context tests when considering overall accuracy. We found that these models yielded significantly more precise predictions than the benchmark, at the cost of slight reduction in maximum recall. We found that the 6000-character context model yielded the best performance based on maximising precision whilst favouring higher recall. The smaller context sizes likely yielded better recognition of clauses on a micro-level, which resulted in performance losses in some categories. We attribute this being due to some categories being more easily identifiable from location-specific information within the contract. These results align with past works and highlight the trade-off between precision and recall. Furthermore, the split context models significantly improved on the benchmark in four of six key categories of highest importance. Thus, this technique shows significant potential to be utilised to improve automated contract review in practice.

While the split category results are mixed, we still were able to draw meaningful conclusions from the results. Our results suggested that the benefit of larger total sample counts outweighed any potential drawbacks from sample imbalances. Additionally, we found potential avenues for follow-up investigation into different ways of splitting categories. Notably, we observed some categories with low samples which performed well in the benchmark and split context models, but poorly when grouped with other low-sample categories. This opens up the idea that there may exist synergistic categories which improve performance when grouped together, which could be a further topic to investigate.

## 5.1 Limitations

While CUAD is one of the largest specialised annotated contract datasets, it is still smaller in comparison to other domain specific training corpuses such as in Chalkidis et. al. (2019). We recognise this to be the primary limitation in our project. As identified in the methodology, despite their being over 13,000 annotations in the dataset, due to the large range of categories, many small-sample categories suffer from lack of samples. Due to the domain expertise and manual annotating process required, this lack of data is a difficult issue and while our project addresses techniques somewhat independent of dataset size, it is likely that this was a major limitation in achieving higher model performance.

## 5.2 Future Works

As mentioned in the discussion of split category tests, we identified some key avenues of follow-up investigation which could potentially be explored in future works. This involves revising the category groupings in different ways, including finding synergistic categories using domain specific analysis, or attempting to group high sample categories with lower sample categories to offset poor results in some categories.

Additionally, as we focused primarily on analysing the isolated impact of splitting context and splitting categories independently, further investigation could be conducted on an experiment to determine the performance of combining both techniques to CUAD. Notably, in the split context tests, we found that some categories performed worse with split contexts compared to the full contract. Further investigation could involve splitting off these categories and processing them separately to the other categories which benefited from context splitting.

Furthermore, due to the experimental process prioritising controlling variables to isolate performance improvements caused by the context splitting and category splitting, another avenue of investigation could be to identify potential benefits of model architecture or pipeline changes, such as including additional pretraining.

# Reference List


Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., Soria, C. (2005). Automatic semantics extraction in law documents. *Proceedings of the 10th International Conference on Artificial Intelligence and Law* p133-140. https://doi.org/10.1145/1165485.1165506

Chalkidis, I., Androutsopoulos, I., Michos, A. (2017). Extracting Contract Elements. *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law* p19-28. https://doi.org/10.1145/3086512.3086515

Chalkidis, I., Emmanouil, F., Malakasiotis, P., Androutsopoulos, I. (2019). Large-Scale Multi-Label Text Classification on EU Legislation. *Association for Computational Linguistics (57)* p6314-6322. http://dx.doi.org/10.18653/v1/P19-1636

Chalkidis, I., Jana, A., Hartung, D., Bommarito, M., Androutsopoulos, I., Katz, D. M., Aletras, N. (2021). Lexglue: A benchmark dataset for legal language understanding in English. arXiv preprint arXiv:2110.00976.

Curtotti, M., McCreath, E. (2011). A corpus of Australian contract language: description, profiling and analysis. *Proceedings of the 13th International Conference on Artificial Intelligence and Law* p199-208. https://doi.org/10.1145/2018358.2018387

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Elwany, E., Moore, D., Oberoi, G. (2019). Bert goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding. https://doi.org/10.48550/arXiv.1911.00473

Gao, X., Singh, M., Mehra, P. (2012) Mining Business Contracts for Service Exceptions. *IEEE Transactions on Service Computing* 5 (3), p333-344. https://doi.org/10.1109/TSC.2011.1

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced BERT with disentangled attention. arXiv preprint arXiv:2006.03654.

Hegel, A., Shah, M., Peaslee, G., Roof, B., Elwany, E. (2021). The Law of Large Documents: Understanding the Structure of Legal Contracts Using Visual Cues. https://doi.org/10.48550/arXiv.2107.08128

Hendrycks, D., Burns, C., Chen, A., Ball, S. (2021) CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. *35th Conference on Neural Information Processing Systems.* https://doi.org/10.48550/arXiv.2103.06268

HuggingFace. (2022). *Source Code for transformers.data.processors.squad*. https://huggingface.co/transformers/v2.4.0/_modules/transformers/data/processors/squad.html

HuggingFace. (2022). *SQuAD v2 – a Hugging Face Space by evaluate-metric.* https://huggingface.co/spaces/evaluate-metric/squad_v2

Green Jr, B. F., Wolf, A. K., Chomsky, C., Laughery, K. (1961). Baseball: an automatic question-answerer. *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference,* p219-224. https://doi.org/10.1145/1460690.1460714

Hirschman, L., Gaizauskas, R. (2001). Natural language question answering: The view from here. *Natural Language Engineering,* 7(4), p275-300. https://doi.org/10.1017/S1351324901002807

Hu, D. (2020). An Introductory Survey on Attention Mechanisms in NLP Problems. In: Bi, Y., Bhatia, R., Kapoor, S. (eds) *Intelligent Systems and Applications*. IntelliSys 2019. *Advances in Intelligent Systems and Computing*, 1038. https://doi.org/10.1007/978-3-030-29513-4_3

Indukuri, K., Krishna, P. (2010). Mining e-contract documents to classify clauses. *COMPUTE' 10: Proceedings of the Third Annual ACM Bangalore Conference* 7, p1-5. https://doi.org/10.1145/1754288.1754295

Jia, R., Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. arXiv preprint arXiv:1707.07328.

Leivaditi, S., Rossi, J., Kanoulas, E. (2020). A Benchmark for Lease Contract Review. https://doi.org/10.48550/arxiv.2010.10386

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Onit. (2022) Enterprise Legal Reputation Report. Retrieved from https://www.onit.com/elr/?utm_source=securedocs-elr-chapter2&utm_medium=website&utm_campaign=ELR-2022

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Conference on Empirical Methods in Natural Language Processing* p2383-2392. https://doi.org/10.48550/arXiv.2004.03705

Rajpurkar, P., Jia, R., Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. arXiv preprint arXiv:1806.03822.

Roegiest, A., Hudek, A., McNulty, A. (2018). A Dataset and an Examination of Identifying Passages for Due Diligence. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*, p465–474. https://doi.org/10.1145/3209978.3210015

Sahare, M., Gupta, H. (2012). A review of multi-class classification for imbalanced data. *International Journal of Advanced Computer Research*, 2(3), p160-164.

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, 7(1), 1-47. https://doi.org/10.1186/s40537-020-00349-y

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vig, J. (2019). A Multiscale Visualization of Attention in the Transformer Model. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, p37–42 https://doi.org/10.18653/v1/P19-3007

Ward, C. (2021). What is a Contract Review?. https://www.linkedin.com/pulse/what-contract-review-colin-a-ward

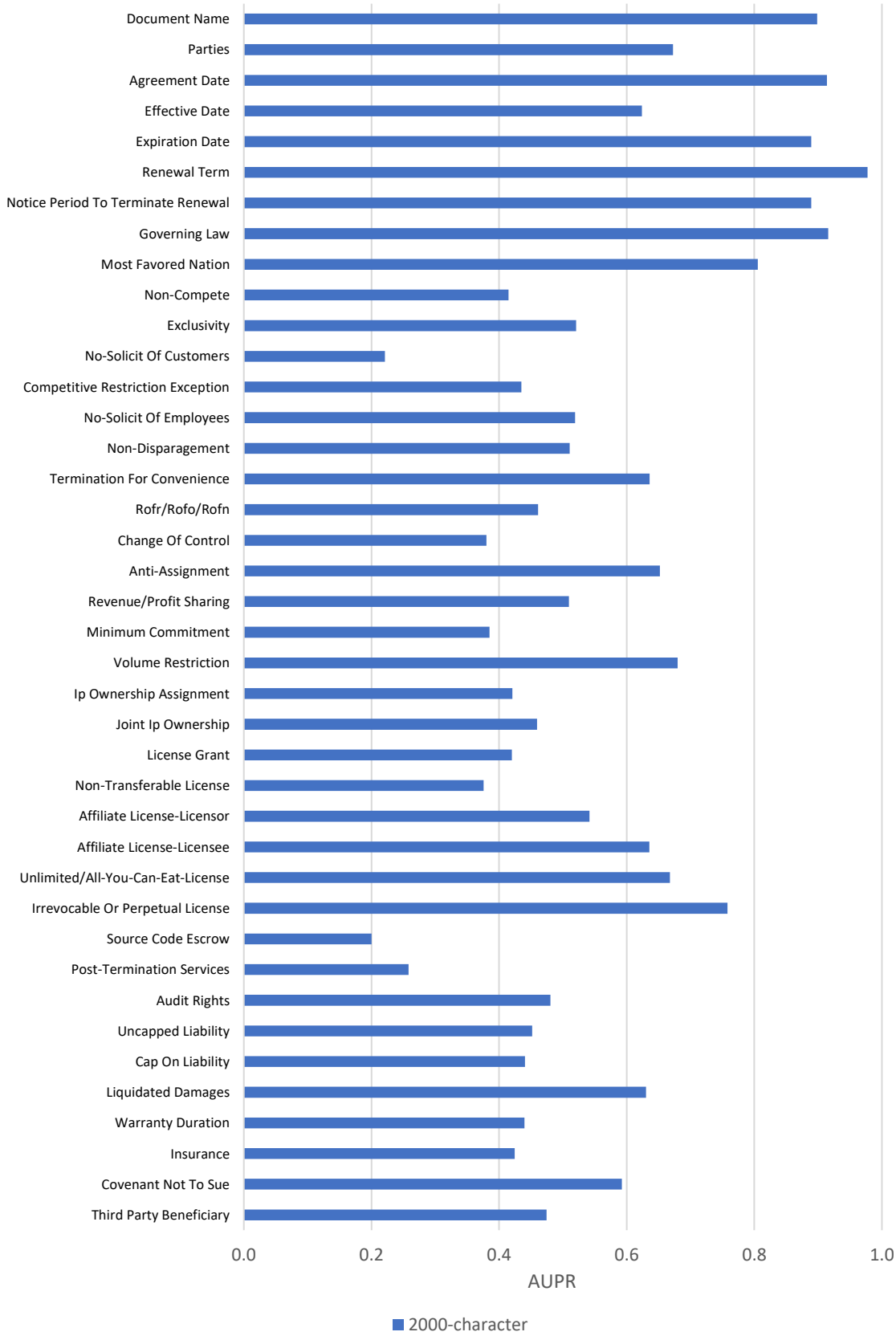
Weissenborn, D., Wiese, G., Seiffe, L. (2017). Making neural QA as simple as possible but not simpler. arXiv preprint arXiv:1703.04816.

Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., Chua, T. S. (2021). Retrieving and reading: A comprehensive survey on open-domain question answering. arXiv preprint arXiv:2101.00774.

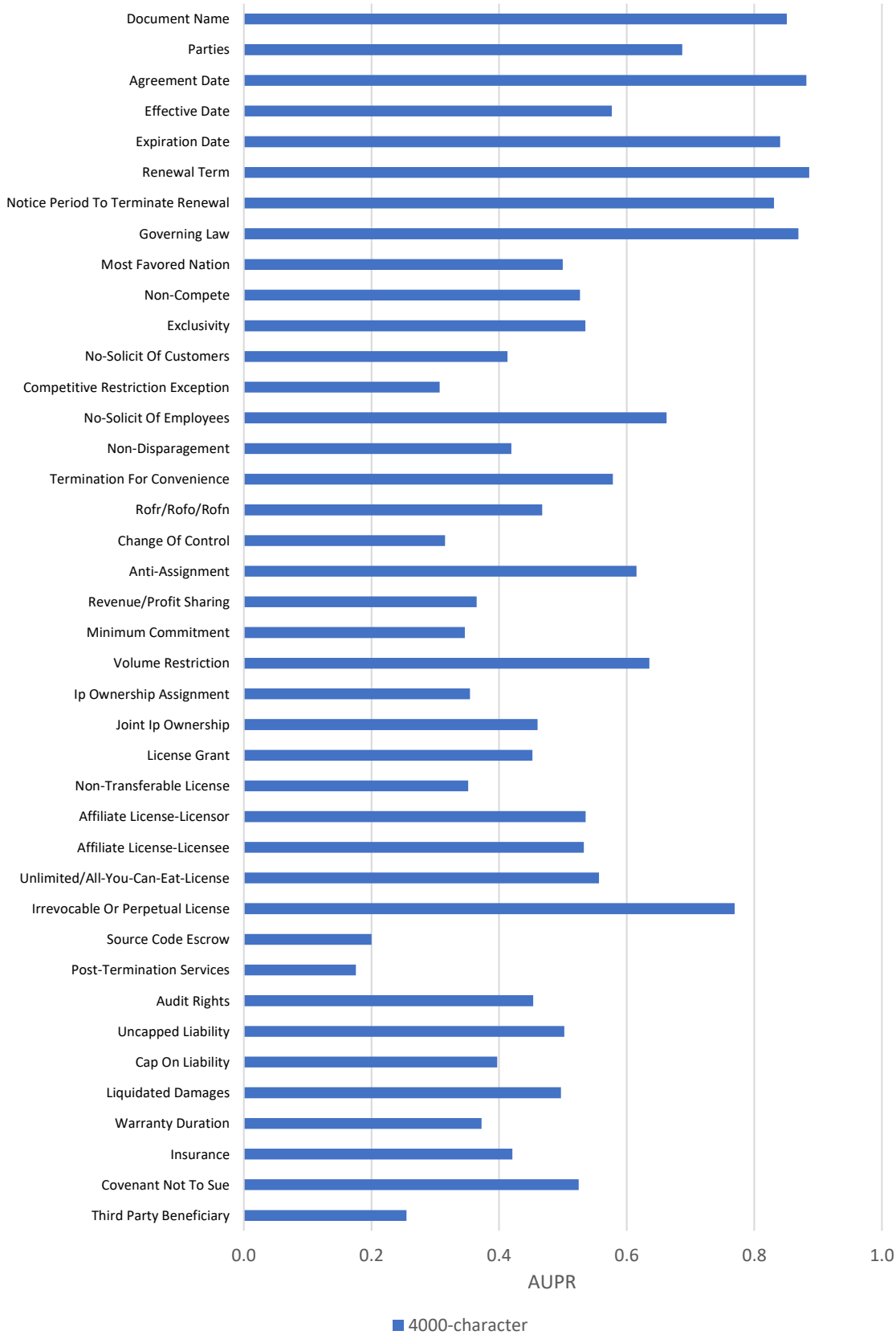
# Appendix A: Split Context Model Performance

| | AUPR | | | | | | |
|---|---|---|---|---|---|---|---|
| **category** | Benchmark | 1000 | 2000 | 4000 | 6000 | 8000 | 10000 |
| Document Name | 0.923 | 0.891 | 0.898 | 0.851 | 0.859 | 0.762 | 0.785 |
| Parties | 0.918 | 0.638 | 0.672 | 0.687 | 0.611 | 0.634 | 0.644 |
| Agreement Date | 0.907 | 0.932 | 0.914 | 0.882 | 0.882 | 0.843 | 0.894 |
| Effective Date | 0.441 | 0.645 | 0.624 | 0.577 | 0.514 | 0.525 | 0.490 |
| Expiration Date | 0.855 | 0.912 | 0.889 | 0.840 | 0.896 | 0.839 | 0.869 |
| Renewal Term | 0.689 | 0.953 | 0.977 | 0.886 | 0.933 | 0.896 | 0.878 |
| Notice Period To Terminate Renewal | 0.465 | 0.873 | 0.889 | 0.831 | 0.881 | 0.829 | 0.807 |
| Governing Law | 0.970 | 0.903 | 0.916 | 0.869 | 0.898 | 0.896 | 0.902 |
| Most Favored Nation | 0.225 | 0.667 | 0.806 | 0.500 | 0.357 | 0.524 | 0.338 |
| Non-Compete | 0.273 | 0.456 | 0.415 | 0.527 | 0.460 | 0.465 | 0.380 |
| Exclusivity | 0.383 | 0.452 | 0.521 | 0.535 | 0.571 | 0.497 | 0.522 |
| No-Solicit Of Customers | 0.168 | 0.263 | 0.221 | 0.413 | 0.258 | 0.321 | 0.558 |
| Competitive Restriction Exception | 0.093 | 0.568 | 0.435 | 0.307 | 0.268 | 0.173 | 0.260 |
| No-Solicit Of Employees | 0.477 | 0.670 | 0.519 | 0.662 | 0.868 | 0.635 | 0.703 |
| Non-Disparagement | 0.186 | 0.614 | 0.511 | 0.419 | 0.502 | 0.407 | 0.428 |
| Termination For Convenience | 0.389 | 0.672 | 0.636 | 0.578 | 0.660 | 0.620 | 0.517 |
| Rofr/Rofo/Rofn | 0.220 | 0.393 | 0.461 | 0.468 | 0.536 | 0.416 | 0.430 |
| Change Of Control | 0.211 | 0.362 | 0.380 | 0.315 | 0.262 | 0.224 | 0.252 |
| Anti-Assignment | 0.763 | 0.620 | 0.652 | 0.615 | 0.622 | 0.620 | 0.594 |
| Revenue/Profit Sharing | 0.364 | 0.489 | 0.509 | 0.365 | 0.455 | 0.470 | 0.567 |
| Minimum Commitment | 0.340 | 0.401 | 0.385 | 0.347 | 0.384 | 0.341 | 0.315 |
| Volume Restriction | 0.170 | 0.609 | 0.680 | 0.635 | 0.573 | 0.690 | 0.725 |
| Ip Ownership Assignment | 0.168 | 0.305 | 0.421 | 0.354 | 0.346 | 0.356 | 0.225 |
| Joint Ip Ownership | 0.250 | 0.500 | 0.460 | 0.460 | 0.572 | 0.361 | 0.486 |
| License Grant | 0.613 | 0.467 | 0.420 | 0.452 | 0.524 | 0.510 | 0.487 |
| Non-Transferable License | 0.334 | 0.550 | 0.376 | 0.352 | 0.493 | 0.423 | 0.303 |
| Affiliate License-Licensor | 0.114 | 0.513 | 0.542 | 0.535 | 0.639 | 0.511 | 0.461 |
| Affiliate License-Licensee | 0.138 | 0.485 | 0.636 | 0.533 | 0.641 | 0.554 | 0.519 |
| Unlimited/All-You-Can-Eat-License | 0.185 | 0.584 | 0.668 | 0.557 | 0.584 | 0.751 | 0.626 |
| Irrevocable Or Perpetual License | 0.457 | 0.781 | 0.758 | 0.769 | 0.785 | 0.773 | 0.740 |
| Source Code Escrow | 0.580 | 0.200 | 0.200 | 0.200 | 0.717 | 0.732 | 0.200 |
| Post-Termination Services | 0.144 | 0.247 | 0.258 | 0.175 | 0.196 | 0.218 | 0.217 |
| Audit Rights | 0.538 | 0.460 | 0.480 | 0.454 | 0.470 | 0.477 | 0.501 |
| Uncapped Liability | 0.163 | 0.814 | 0.452 | 0.502 | 0.248 | 0.293 | 0.237 |
| Cap On Liability | 0.447 | 0.451 | 0.441 | 0.397 | 0.407 | 0.368 | 0.381 |
| Liquidated Damages | 0.190 | 0.575 | 0.630 | 0.497 | 0.610 | 0.569 | 0.611 |
| Warranty Duration | 0.206 | 0.401 | 0.440 | 0.373 | 0.469 | 0.402 | 0.329 |
| Insurance | 0.510 | 0.461 | 0.424 | 0.421 | 0.415 | 0.409 | 0.394 |
| Covenant Not To Sue | 0.467 | 0.544 | 0.592 | 0.525 | 0.450 | 0.563 | 0.585 |
| Third Party Beneficiary | 0.196 | 0.394 | 0.474 | 0.255 | 0.294 | 0.301 | 0.351 |
| **AVERAGE** | **0.403** | **0.568** | **0.565** | **0.523** | **0.553** | **0.530** | **0.513** |

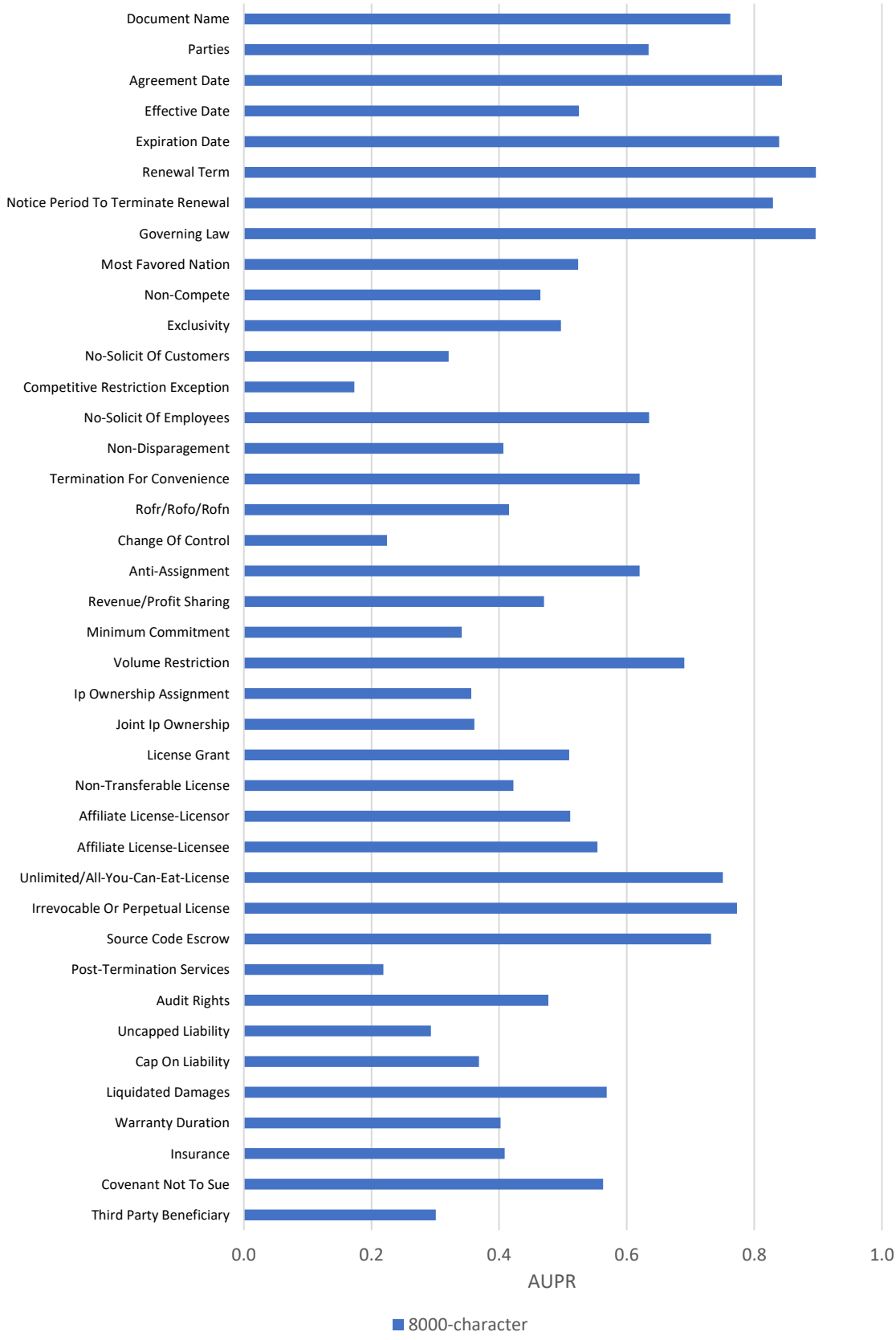
## Individual Category Performance of the 2000-character-split Model



## Individual Category Performance of the 4000-character-split Model



## Individual Category Performance of the 8000-character-split Model



## Individual Category Performance of the 10000-character-split Model