340 Project Paper

Josh Yip, Qingyuan Kong

Classification of Contract Clauses using Selective Models

Introduction

With the increase of people becoming individual business owners and content creators in recent years, the contracts people read also increased. However, not everyone is familiar with complex legal terms and may take a long time to identify important information in those contracts. The lack of legal knowledge could potentially make those independent business owners vulnerable if met with malicious contracts. A potential method to mitigate the harms may be to consult a lawyer, yet this may not be accessible to everyone. This project aimed at creating a model that helps individuals break down their contract to key clauses, and identify them based on the scope the user wants to focus on, such as terms and conditions, termination and renewal, IP rights and more. Based on our vision, this model can empower individuals in their decision making and simplifies the process of signing contracts.

Methodology

We created a model that applies transfer learning to an existing subset of LegalBERT called ContractBERT (https://huggingface.co/nlpaueb/bert-base-uncased-contracts). This transformer architecture was trained on 100,000+ legal documents, making it a suitable foundation for our project. Based on this transformer architecture, we decided to use contract datasets to finetune the model for our use. Upon comparison of several datasets available online, we chose the CUAD dataset by the Atticus Project (https://www.atticusprojectai.org/cuad) for it is labeled manually by experienced lawyers and separated into clauses, and created specifically for NLP research.

The CUAD dataset consists of 510 contracts with over 13,000 labels. The 510 contracts are separated into 25 different types, with each type ranging from 3 individual contracts to more than 30. Each contract is then separated into clauses that may or may not be present in an CSV file. The columns of clauses included information such as Parties, Revenue/Profit Sharing, Ip Ownership Assignment, Post-Termination Services and more.

Based on our aims on making a model that could cater specific needs of different users, we decided to use a selective model approach for this project. We broke down the different columns of clauses into groups that we believed potential users of our model would be interested in, and settled in these 5 groups: Terms and Conditions, Compensation and Financials, Intellectual Property (IP) Rights, Confidentiality and Non-Disclosure, and Termination and Renewal. Each group will have a model that is specifically trained with more emphasis on columns we decided are relevant for the group. See the list of label columns for each group in the table below.

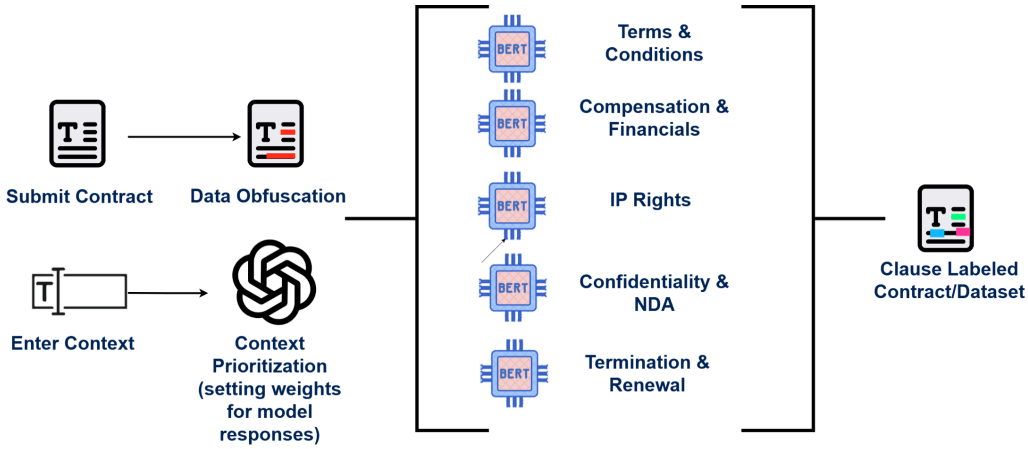
Terms and Conditions	**Compensation and Financials**	**Intellectual Property (IP) Rights**	**Confidentiality and Non-Disclosure**	**Termination and Renewal**
Governing Law	Revenue/Profit Sharing	Ip Ownership Assignment	Non-Disparagement	Termination For Convenience
Competitive Restriction Exception	Price Restrictions	Joint Ip Ownership	No-Solicit Of Customers	Post-Terminatio n Services
Non-Compete	Minimum Commitment	License Grant	No-Solicit Of Employees	Expiration Date
Exclusivity	Volume Restriction	Non-Transferable License		Renewal Term
Anti-Assignment	Liquidated Damages	Affiliate License-Licensor		Notice Period To Terminate Renewal
		Affiliate License-Licensee		Change Of Control
		Unlimited/All-You-C an-Eat-License		
		Irrevocable Or Perpetual License		
		Source Code Escrow		

Model Architecture

We opted for a selective-model approach, creating separate models for specific clause classification tasks (e.g., NDA vs. IP) instead of a general model for all clause categories. This modular design simplifies updates and addresses our imbalanced dataset, which hindered general

model performance in preliminary tests. Tailored models improved training for individual clause types and allowed targeted updates for specific classifications, despite requiring more time for initial training. This approach ultimately streamlines future updates with new datasets.

Our work in contract classification is intended for a full scale web application for helping users (individuals, entrepreneurs, small businesses) understand their contract's stipulations and clauses. To this end, our model seeks to label contract clauses given the context of a user's focus of interest (if they are interested in IP rights over Compensation for example). An additional use case of our model would be producing labeled contract datasets for future model training and improvement.



The columns we found related to each model are then concatenated row-wise to form a single text input. These positive samples are labeled 1; negative samples are labeled 0. Positive and negative samples are combined into a single DataFrame with "text" and "labels". Finally the tokenizer prepares the text for input suitable model training.

Results

The initial 5 models were trained solely on the 510 contracts and their labels. The train-test split separated them into a training set of 408 and testing set of 102. However, due to the limited data, the accuracy for all of them was extremely low. With the lowest accuracy found in the Intellectual Property (IP) Rights model, as low as 5.8% accuracy. The other model ranged between 38% to 45%, with the highest being the Compensation and Financials model.

The low accuracy made us realize that the model may be overfitting on the training group due to the small sample size. To mitigate the effects of the small dataset size, we decided to conduct some data augmentation on the original data. While there are other methods such as looking for additional datasets, we did not find other datasets that were labeled in a similar fashion, and we also lack the professional knowledge to label them ourselves, making data augmentation the best alternative.

The data augmentation process was conducted via paraphrasing the texts, using a Hugging Face Transformers library named T5-small (https://huggingface.co/google-t5/t5-small) and its text to text generation capabilities. This model is pre-trained on tasks such as translation, summarization, and paraphrasing, making it suitable for our use case. As the paraphraser generates paraphrased versions of each input sentence using beam search, the labels of the original sentence were also applied. Finally, the paraphrased dataset was concatenated into the original dataset, increasing the sample size from 510 to 3570. This process gave us 2856 samples for training and 714 samples for testing.

After training on the augmented dataset, the testing accuracy increased significantly. The largest increase is found in the Intellectual Property (IP) Rights model, increasing from 5.8% to 86.3%. The Terms and Conditions model also reported a 98% accuracy. However, the high accuracy should not be completely trusted, as the data augmentation potentially resulted in data leakage between the training set and testing sets, with similar paraphrased text of the same original text present in both training and testing data.

Therefore, we decided to conduct some blind testing on unlabelled, unseen contracts and see how individual sentences get classified (1 as relevant information to this group and 0 as irrelevant). First, we noticed why older models are performing badly. It has a tendency to classify a large percentage of sentences as relevant or a large percentage as irrelevant, which matches the low accuracy we found in older models. With one particular case in the IP model, the old model with 5.8% accuracy classified only one sentence as irrelevant but everything else as relevant in a test contact. This performance is not satisfactory as it does not match our needs of highlighting the important information for the users to simplify the process for them. On the other hand, the improved models proved to be much better at the task, especially on short sentences and headings. The same testing contract now does not have headers marked as important information. However, we did notice the tendency to mark long sentences as all relevant. This

could be due to the fact that contract sentences are long, and any information considered relevant by the model will be marked as relevant. We are considering in future tests and improvements on models, we will have to reduce the length of clauses that are marked as relevant, for example separating by commas rather than sentences.

Conclusions

Looking at the results, we found that while our model showed some progress in classifying relevant clauses, there were still several areas of improvement. First, the data augmentation should be applied after the train test split to avoid any data leakage. Second, the model should be trained on more data to gain better understanding on different clauses. It would be useful if we could consult professionals and create more labelled data. Finally, based on the model's behavior on blind testing, it may indicate that a summarization job may be more useful than labelling clauses, as the models tend to label everything as relevant and important.

In addition to improvements on the model itself, we can work on completing our initial goals on the model architecture. Data obfuscation and context prioritization can both be useful on the user side of this project. The potential user would not face fears of personal information leakage if we could successfully employ data obfuscation, while context prioritization allows the correct specialized model to be picked.