SConE: Contextual Relevance based Significant CompoNent Extraction from Contracts

Hiranmai Sri Adibhatla, Manish Shrivastava
Language Technologies Research Center, KCIS
IIIT Hyderabad, India
hiranmai.sri@research.iiit.ac.in,m.shrivastava@iiit.ac.in

Abstract

Automatic extraction of "significant" components of a legal contract, has the potential to simplify the end user's comprehension. In essence, "significant" pieces of information have 1) information pertaining to material/practical details about a specific contract and 2) information that is novel or comes as a "surprise" for a specific type of contract. It indicates that significance of a component may be defined at an individual contract level and at a contract-type level. A component, sentence or paragraph, may be considered significant at a contract level if it contains contract specific information (CSI), like names, dates or currency terms. At a contract-type level, components which deviate significantly from the norm for the type may be considered significant (type specific information (TSI)). In this paper, we present approaches to extract "significant" components from a contract at both these levels. We attempt to do this by identifying patterns in a pool of documents of the same kind. Consequently, in our approach, the solution is formulated in two parts: *identifying CSI* using a BERT based contract-specific information extractor and *identifying TSI* by scoring sentences in a contract for their likelihood. In this paper, we even describe the annotated corpus of contract documents that we created as a first step toward the development of such a language-processing system. We also release a dataset of contract samples containing sentences belonging to CSI and TSI.

1 Introduction

Contracts are agreements, between two or more parties, that govern what each party can or cannot do and are usually dense in information. Extracting contract elements and locating novel clauses and assignments from a legal contract is a desired feature by many as it will greatly simplify and accelerate user comprehension. Traditionally, it requires a domain expert as there are parts of a contract that can only be noticed by a reader experienced in reviewing contracts. For an untrained eye, it is often difficult and time consuming to identify rare and unique sentences. To reduce dependency on experts and to lessen the human effort required, in this paper we introduce approaches for automatic identification and extraction of significant components of the contract.

When compared with the corpora on which most pre-trained deep models are based, the structure and vocabulary of texts in contracts differ significantly. Contracts frequently take constrained forms, sometimes even "template-like" for the sake of ensuring legal unambiguity. On carefully examining the semantics and structure of diverse legal contracts sourced from SEC EDGAR [1] *(employment, software license, purchase, severance)*, we observe that

i) *within contracts of same category*, although the wording and sentence structure differ between individual contracts, the information conveyed remains almost the same,

ii) *within an individual contract*, within an individual contract, we have compo- ii) within an individual contract, we have components, sentences or paragraphs that are remarkably distinct with little redundancy

Components in an individual contract can be broadly classified as:

Templatised sentences are sentences that follow a template. A phrase or only a part of the sentence may vary and the rest of the content is semantically same across contracts. Examples include contract elements (Chalkidis et al., 2017) like title of the contract, parties involved in the contract, dates, governing law.

As observed in Table 1, the sentences for an individual contract can be generated from a template by filling in relevant information for the "effective date" and "governing law". The values for "effec-

[1]https://www.sec.gov/edgar/search-and-access

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 161 - 171
December 15-18, 2022 ©2022 Association for Computational Linguistics

Templatised Sentences
This Agreement shall be effective as of *November 5, 2014 (the Effective Date).*
GOVERNING LAW.This Agreement shall be construed and interpreted in accordance with the internal laws of the *State of California*.

Table 1: Sentences with a Template Structure

Boilerplate Sentences
While employed by the Company hereunder, Executive shall be eligible to participate in the Company 's employee benefit plans as in effect from time to time pursuant to the terms of those employee benefit plans.
No waiver of any breach or condition of this Award Agreement shall be deemed to be a waiver of any other or subsequent breach or condition whether of like or different nature.

Table 2: Sentences with Standardized Clauses

Rare Sentences
To achieve balance, your current tax withholdings may cease and a hypothetical rate of tax may be calculated and withheld from your wages.
No additional Stock Units granted as part of the Award may be earned following the Change in Control.

Table 3: Sentences with Rare elements

tive date" and "governing law" will be different for different contracts. In templatised sentences, the information changes rapidly for each document, as the values are unique to each contract.

Boilerplate sentences[2] sentences that are standard formulations, and uniformly found in all contracts of a type. They are huge in number and constitute a large portion of contract. In this paper, we extend the definition of Boilerplate sentences to include sentences which contain the same semantic content across contracts but differ lexically and structurally. As we can see in Table 2, these clauses are standard across contracts of a specific type. Business and technical documents often use boilerplate sentences to improve efficiency and standardize language and structure. The information divergence between contracts of a type is almost constant for boilerplate sentences.

Rare sentences in a contract include content not commonly found in contracts of that type and hence are conspicuous by their presence in the current contract. In Table 3, first example refers to hypothetical tax rate which applies to employees who work at an onsite location. Similarly in the second example, no additional stock units are granted if there is a change in control of the organization. Both these clauses are situational and do not appear in most contracts of that category. Intuitively, these sentences will be of interest to anyone examining the contract because they bring in novelty. Rare sentences in a contract are identified on the basis of contract type to which a contract belongs.

In summary, a contract has -

i) template sentences, which contain contract specific information (CSI) and are generic across contract types.

ii) rare sentences which deviate from other contracts of the same type, they convey type specific information (TSI) and can be recognised only if one has an in depth understanding of the content usually present in the contract type.

iii) common and well understood clauses that constitute boilerplate sentences. In terms of volume, they account for majority of the sentences in a contract.

This approach of extracting significant components does not really qualify as a standard summarization task because there is no merit in summarizing boilerplate sentences which are well understood. Abstractive summarization (Zhang et al., 2020a) techniques would inadvertently change the semantics of the contract. Even when compared to an extractive setting (Nallapati et al., 2017), in this study our main focus is to accentuate rare and templatised sentences as significant components in comparison to boilerplate sentences. Unfortunately, we are not aware of any large, open corpora of contracts for running comparable experiments.

The outcome of our approach is presented in two formats *a) highlighted input document* Figure 1- where sections of interest are highlighted within the overall contract. This helps in vizualizing the significant components of the contract. *b) a cover-page* - a consolidated page containing the extracted significant components. The effectiveness of the automated significant components identification model was further evaluated by conducting an experimental study that compares the performance between human and machine for the task. The contribution of our work in addition to identifying “significant" components is to understand

[2]The term boilerplate refers to standardized text, copy, documents, methods, or procedures that may be used over again without making major changes to the original.

how much fine-tuned data is required for achieving a moderately reasonable accuracy. This becomes important as contract types can vary considerably, and organizations would be burdened with huge annotation efforts for every document type.

2 Related Work

Language being the core of law and legal contracts, an increased interest in applying natural language processing techniques to a wide range of problems ranging from information extraction to sentence prediction in law (Zhong et al., 2020; Hendrycks et al., 2021; Kalamkar et al., 2021; Zheng et al., 2021) has been observed. Considerable amount of work has been done in contract analysis and information extraction from contracts (Yang et al., 2013; Silva et al., 2020; Mittal et al., 2015).

The most obvious approach to automatic contract element extraction is to model it as sequence labeling task. Statistical methods like Conditional Random Fields (Finkel et al., 2005; Xu and Sarikaya, 2013) were popular for sequence labeling prior to neural networks. (Chalkidis et al., 2017) involved hand written rules along with hand crafted features to uniquely identify and extract the contract elements. Recently, neural networks (Huang and Xu, 2015; Ma and Hovy, 2016; Chalkidis and Androutsopoulos, 2017) and BERT (Devlin et al., 2019) based approaches (Zhang et al., 2020b; Chen et al., 2019) were developed for sequence labeling and slot with joint intent classification. Our work for CSI extraction closely resembles (Zhang et al., 2020b) where the contract elements are extracted from regulatory filings and property lease agreements using the standard BIO tagging scheme for the contract elements of interest. We include more categories of contracts (employment, incentive, purchase, severance, software-license) and the contract elements are majorly kept consistent.

Scope identification is another popular area of research in legal domain as it is tedious to read legal documents. Contracts or legal documents contain many key sentences. It often becomes necessary to have domain knowledge regarding contracts to avoid missing any important or key information. Summarization (Andhale and Bewoor, 2016) is a reliable approach and summarizing legal contracts was attempted (Kubeka and Ade-Ibijola; Kore et al., 2020) by taking the document features and ordering the sentences according to their importance. Classification and hand crafted rules (Le et al., 2020) was another recent approach to precisely identify the scope and was applied to construction contracts to identify requirements automatically. These techniques do not differentiate between boilerplate sentences which forms the bulk of the contract and the other sentences of the contract. Instead of summarizing well understood and accepted clauses, our study intends to focus on contract specific information (CSI) and contract type specific information (TSI).

Regression (Ren et al., 2016; Zopf et al., 2018) is another technique where the sentences are scored on their importance and the model learns to include sentences in a summary based on the scores it predicts. Based on our observations that legal contracts of same category have repetitive information, we devised an approach to calculate sentence likelihood with respect to the contract type and use these scores to identify TSI. The likelihood scores calculated using LaBSE (Feng et al., 2022) while BERT (Devlin et al., 2019) was adapted to learn and predict these likelihood scores.

3 Approach

Significant component extraction is accomplished in two stages:

(1) Identifying CSI by processing each sentence of the document and identifying sentences with contract elements (Chalkidis et al., 2017).

(2) Identifying TSI by assigning a likelihood score to all sentences in a contract.

These stages contribute in effectively identifying the scope of significant components, by automating contract processing and extracting text relating to CSI and TSI from the contracts. We use LEGAL-BERT-BASE (Chalkidis et al., 2020) which is fine-tuned on BERT (Devlin et al., 2019) for legal domain and has shown substantial improvement in challenging downstream tasks like multi-label-classification. Within the wide categories of legal contracts available, we ran our experiments on the contract types mentioned in Table 9.

The overall architecture is shown in Figure 2. The input to the model is a document D containing a set of sentences S. The output is a set of sentences P, that effectively highlight information unique and specific to the document D, such that $P \in S$.

Notwithstanding the foregoing, Employer agrees that Employee may continue his work with Justworks, Inc. and Consonance Capital Partners during the term of this Agreement.

Employee shall at all times observe and abide by the Employers policies and procedures as in effect from time to time. BOILERPLATE SENTENCES

Section 2.Compensation. In consideration of the services to be performed by Employee hereunder, Employee shall receive the following compensation and benefits: 2.1Base Salary.

Employer shall pay Employee a base salary of one million five hundred thousand dollars ($1,500,000) per annum, less standard withholdings and authorized deductions.

Figure 1: Example snippet of a highlighted contract. Sentence in green is TSI while the yellow sentence is CSI. The other two sentences are boilerplate.

4 Identifying CSI

Identifying contract elements is similar in approach to identifying named entities but is not directly extendable without retraining them on contracts. NER systems typically identify persons, organizations, dates, locations, currency terms etc,. Contract elements would carry more features attributed to it along with being a named entity. For example, a NER system can identify dates and persons but will not be able to differentiate if the date is an effective start date or termination date. Similarly not all instances of persons, organization or location in a contract would be contract parties or governing law elements. The sentences that contain these CSI are almost in a template like schema, therefore training a sequence labeling model to understand the sentence semantics and to extract sentences which contain contract elements, yields better results. We sampled 500 legal documents (100 documents of each category mentioned in Table 9). These documents are then pre-processed into paragraphs. A paragraph as a unit might be of a higher value than an isolated sentence. The documents are split into train, test and validation bins in the ratio 7:2:1. Commonly applicable contract elements are identified and selected as contract elements of interest. Most of the contract elements are phrases rather than a single token, therefore we pose it as a sequence labeling task using a standard BIO tagging scheme (Tjong Kim Sang, 2002). We manually annotated the contracts to mark the selected contract elements. The contract elements are kept consistent across the contract types as it is common for contracts to follow a fixed structure with a certain number of prescribed elements (*contract title, contract parties, effective start date, termination|maturity date, governing law etc.*). It also reduces the training and annotation effort and increases the generality of the model. The contract elements we annotated are listed in Table 7.

4.1 Identifying CSI Model

In the CSI model we extend BERT (LEGAL-BERT-BASE) for sequence labeling in order to identify phrases of interest. All contracts are divided into paragraphs. The input sequences are tokenized using BERT tokenizer and special tokens [CLS] and [SEP] are added at the beginning and end of the input sequence respectively. All the input sequences are padded to a maximum length of 256 tokens. After passing through BERT, we apply a linear layer and CRF layer on top of the hidden states output of the last layer. The model is trained for 25 epochs with learning rate of 1e-05.

5 Identifying TSI

Contract type specific information (TSI) extraction problem has not been studied extensively and is the main focus of our study. We identify unique or novel details concerning the contract by looking at structural and semantic similarities among a pool of contracts belonging to a specific type. A clause that is rare for an employment type contract may not be rare for a stock options awards type contract.

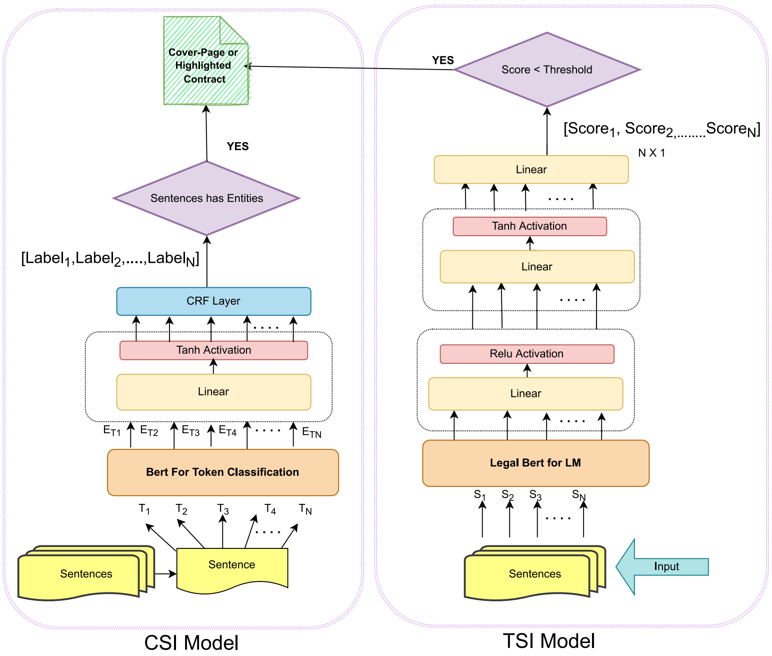

Figure 2: SCoNE Architecture for CSI and TSI

Figure 3 highlights few clauses that may seem ordinary but are different from their usual construction in contracts.

Based on our observations, legal contracts of same category have repetitive information (Boilerplate). This requires ranking the sentences based on a metric for rarity. Scoring sentences (Ren et al., 2016; Zopf et al., 2018) based on both importance and redundancy among sentences was attempted for summarization (Nallapati et al., 2017) tasks. The approach however, does not guarantee inclusion of rare and unique sentences as sentences scored based on their importance are most likely to pick boilerplate sentences since they are the core of any contract. Redundancy is almost negligible for business documents like contracts. TextRank (Mihalcea and Tarau, 2004) is a popular graph-based unsupervised ranking model for text processing. It identifies text units that best define the task at hand and links them with sub units of text by identifying relations among them. The Local Outlier Factor (LOF) algorithm (Pedregosa et al., 2011) is an unsupervised anomaly detection method which computes deviation of a given data point with respect to its neighbors. We applied both TextRank and LOF algorithm on our sampled data as baselines. Since our aim is to capture the rare sentences, we sorted TextRank scores in ascending order and considered the top sentences as rare. Though the model works well in capturing rare information, deciding on the threshold or cut-off is often difficult as it would differ from contract to contract and contract type to contract type.

We devised an unsupervised approach to calculate TSI score with respect to the contract type and use these scores to identify TSI. TSI score of a sentence here indicates the confidence with which a given sentence is a part of a specific contract type.

Identifying rare components of a contract type is often limited by the presence of named entities in templatised sentences. These templatised sentences, though common across contract types, would be counted as rare by the virtue of having named entities in them. The information contained in such sentences is often extracted using Contract Element Extraction approaches (Chalkidis et al., 2017). In order to ignore these sentences and to make sentences more comparable across contracts we mask all the named-entities in contracts using spaCy[3] to replace named-entities by their type (people's names to PERSON, organization names to ORG). Masking sentences that contain named-entities increases its TSI score. Table 4 shows examples of few sentences whose TSI score has increased after masking named-entities.

5.1 Mean-Max Pooling

Though contracts of a type contain repetitive information, the vocabulary and structure might change.

[3]http://spacy.io

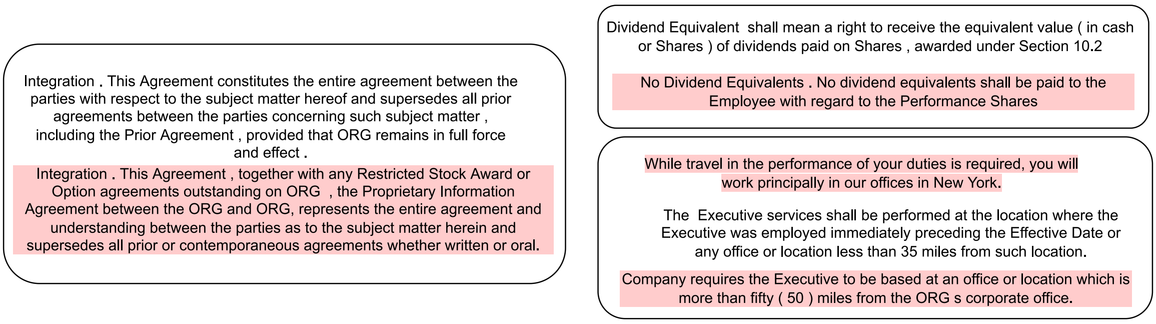

Figure 3: Novel sentences snippets, highlighted in pink

Original Sentence	lscore	Entities Masked Sentence	lscore
EX-10.8 4 a17-1046_1 EX-10.8 EXHIBIT 10.8 EMPLOYMENT AGREEMENT This EMPLOYMENT AGREEMENT (the Agreement) is entered into and effective as of this 3rd day of March	0.75	EX-10.8 4 a17-1046_1 EX-10.8 EXHIBIT10.8 EMPLOYMENT AGREEMENT This EMPLOYMENT AGREEMENT (the Agreement) is entered into and effective as of this DATE DATE DATE (the Effective Date)	0.86
Term of this Agreement. The Term of this Agreement shall mean the period commencing on the Effective Date and ending on March 31	0.64	Term of this Agreement . The Term of this Agreement shall mean the period commencing on the Effective Date and ending on DATE DATE.	0.88

Table 4: Sentence Likelihood Scores (lscore) with and without Masking Entities

Textual overlap methods, therefore, would not be able to capture similar sentences across documents.

In order to estimate how frequently a sentence appears in the documents of a type, we compute semantic similarity between all sentences across all documents using LaBSE (Feng et al., 2022). We consider the maximum semantic overlap depicted by LaBSE as the indicator of semantic presence of the concept expressed by a sentence. Thus, we are approximating the expected count of a sentence (concept) occurring for a type using LaBSE score as proxy.

Let, S_{ij} be the j^{th} sentence in document D_{i} and S_{kl} be the l^{th} sentence in document D_{k}. Assuming no redundancy of concepts in legal contracts (each concept occurs once in a contract), we want to "count" the number of times a sentence appears in a document of a specific type. Thus,

$$Count_k(S_{\mathrm{ij}}) = \max_{1 \leq l \leq p} (LaBSE(S_{\mathrm{ij}}, S_{\mathrm{kl}})) \quad (1)$$

Where, p is the length of document D_k and LaBSE(S_{ij},S_{kl}) is the semantic overlap between the sentences. $Count_k(S_{\mathrm{ij}})$ would determine the degree of semantic overlap of S_{ij} with S_{kl}.

The "count" obtained for the sentence S_{ij} is mean pooled over the number of Documents N. This Mean-Max pooled LaBSE similarity score is assigned as the likelihood of a sentence.

The Likelihood score of S_{ij} is calculated using Equation 2.

$$TSIScore(S_{\mathrm{ij}}) = \frac{(\sum_{i=1}^{N} Count_k(S_{\mathrm{ij}})}{N} \quad (2)$$

Sentences that are very common across a type would have a higher likelihood score compared to sentences whose occurrence is semantically low. We are looking for sentences that have low mean similarity score i.e, low likelihood score.

5.2 Likelihood Approximation Model (TSI-A)

The process described in section 5.1 for calculating likelihood of each sentence would be quadratic in the total number of sentences and computationally expensive at runtime. Therefore, we train a BERT model on likelihood scores computed on contracts from five categories collected from SEC EDGAR. (Table 5) refers to contracts distribution in data slice of 5000 randomly selected files. This model would learn to predict the likelihood of a sentence given the contract type.

In the TSI-A model we extend BERT (LEGAL-BERT-BASE) for this regression. Documents are segmented into paragraphs and tokenized using

Contract Type	Number of Contracts
Employment	2200
Incentive	650
Severance	500
Purchase	750
Software License	600

Table 5: Contract distribution

BERT tokenizer, adding special tokens [CLS] and [SEP] at the beginning and end of the input sequence respectively. The input sequences are padded to a maximum length of 256 tokens. The final hidden states output is passed through linear layers with an activation layer in between for non-linearity. The last layer returns a score which serves as the sentence likelihood score. The model is trained for 15 epochs with learning rate of 1e-05. The loss criteria is MSE (Means Squared Error) and the objective is to minimise the loss between the predicted scores and the training scores. Pearson correlation scores are calculated between the test scores generated by using Equation 2 and the trained BERT regression model.

6 Human Evaluation

To assess the effectiveness of the TSI model we conducted an experimental study that compares the performance of the model against a human annotated corpus. Two annotators were asked to read the contract set provided to them and then label the sentences as rare or familiar. We chose to make this a binary classification task for the humans in order to reduce cognitive load.

Model generated scores in the test set were converted to labels based on their likelihood scores thresholded by the *knee-point* value for each class in Figure 4. If the sentence score is below the threshold set for rare sentences, then the sentences are labeled rare (0) . If the sentence score is above the threshold (set for rare sentences), then it is labeled familiar (1) . Table 6 details the precision, recall and f1 scores of both the annotators on selected contract types.

	Annotator 1			**Annotator 2**		
Contract Type	P	R	F_1	P	R	F_1
Employment	0.94	0.94	0.94	0.88	0.93	0.91
Incentive	0.92	0.97	0.94	0.92	0.90	0.91
Severance	0.99	0.90	0.94	0.99	0.83	0.90
Software License	0.99	0.94	0.96	0.99	0.87	0.93

Table 6: Human Evaluation Statistics

7 Evaluation

For evaluation, the masked contracts in the test set are divided into paragraphs, tokenized using BERT tokenizer and padded with special tokens ([CLS] and [SEP]).

7.1 CSI Model Evaluation

	F_1	*P*	*R*
ContractParties	0.92	0.89	0.95
ContractTitle	0.81	0.72	0.94
EffectiveDate	0.84	0.80	0.89
GoverningLaw	0.55	0.40	0.86
EmploymentRole	0.42	0.42	0.42
SalaryCompensation	0.49	0.43	0.57
TerminationDate	0.40	0.60	0.30

Table 7: Evaluation of Contract Elements

The table 7 shows micro-averaged metrics F_1, precision and recall across the selected contract elements. By examining these results, we can infer that common elements like ContractTitle, ContractParties, EffectiveDate which occur in all documents are well generalised by the BERT model and so have higher precision and recall values. The precision and recall scores are low for contract elements like TerminationDate, SalaryCompensation which have not commonly occurred in the test contracts sampled. The primary reason contributing to these low values is that contracts are sometimes amendments to pre existing contracts and they may not have all the contract elements that a new contract would mention. Table 8 shows the frequencies of the contract elements in both train and test bins after deduplication. The low representation of TerminationDate and SalaryCompensation samples in the train and test data explains low precision and accuracy values. The positives from this result is BERT is able to generalise commonly occurring contract elements with samples as low as 100 contracts. For uncommon contract elements, it requires more data.

7.2 Pearson Correlation Evaluation

Fig 4 shows the plots for sorted likelihood scores of sentences for each contract type. We observed that the plot is similar across contract types mentioned in Table 9 under contract types. From the plot we inferred that likelihood scores of sentences follow a trend. For all contract types, there exists sentences that have low likelihood and sentences which are more probable.

Contract Element	Frequency in Train Data	Frequency in Test Data
ContractParties	218	62
EmploymentRole	179	52
EffectiveDate	131	32
GoverningLaw	83	22
ContractTitlle	80	15
TerminationDate	38	3
SalaryCompensation	12	2

Table 8: Frequency of Contract Elements in Train and Test data

Contract Type	Pearson Correlation on Kfold
Employment	0.996
Incentive	0.998
Severance	0.990
Software License	0.997
Purchase	0.987

Table 9: Averaged K-Fold Validation for Pearson Correlation of test and predicted likelihood scores

i) lower likelihood score : these sentences map to rare sentences, not normally present in all the contracts of that category.

ii) average and above likelihood score : these sentences map to boilerplate sentences which uniformly occur in all the contracts with a minor change in wordings or expression and core sentences that contain named entities. Masking the named entities increases the likelihood scores of the templatised sentences.

Table 4 identifies few examples and compares original unmasked sentences with sentences masked using spaCy, where 'lscore' refers to the likelihood score. We observe that masking entities has shown impact on the sentence likelihood scores.

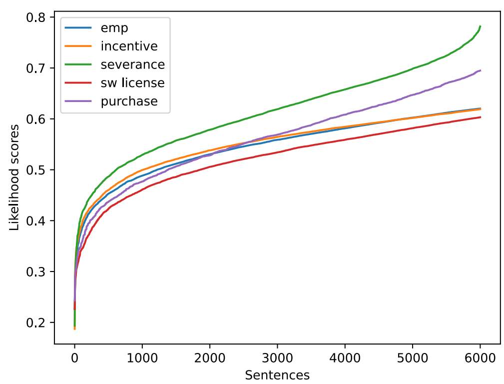

Figure 4: Sorted Likelihood Scores of Sentences

$$C = \frac{(N \sum_{i=1}^{N} T_i P_i - (\sum_{i=1}^{N} T_i)(\sum_{i=1}^{N} P_i))}{\sqrt{N \sum_{i=1}^{N} {T_i}^2 - (\sum_{i=1}^{N} T_i)^2} \sqrt{N \sum_{i=1}^{N} {P_i}^2 - (\sum_{i=1}^{N} P_i)^2}} \tag{3}$$

To measure the performance of our proposed model in predicting the likelihood score, we compute the Pearson product-moment correlation (C) (Benesty et al., 2009) between likelihood scores computed by mean pooling LaBSE similarity scores (calculated using Equation) (T) and likelihood scores generated by the TSI-A model (P), for a sample of 10000 sentences (N) using (Equation 3). Pearson correlation estimates the degree of statistical relationship between two independent variables. A high positive correlation between the actual and predicted values implies that the model can be trusted to work reasonably well on new unseen contracts of that category. For calculating the sentence Likelihood using TSI model, K-fold validation (with k=3) was performed. Table 9 has the Pearson correlation scores averaged for K-fold data sets on contract types considered. The high Pearson correlation values instill confidence that the model can identify rare sentences with reasonable accuracy.

7.3 TSI-A Model Evaluation

To the best of our knowledge, there are no publicly available corpora for rare sentence identification. But, rare sentence identification can be considered as either a ranking task or as an outlier detection task. Therefore, TSI-A model was evaluated against TextRank and LOF outlier detector applied on the sampled data. To keep the evaluation on similar grounds, we converted the likelihood scores obtained using TSI model to labels (0,1) based on the knee-point. On the 100 contracts sampled for test from each contract type, the contracts were split into train, test and validation bins in the ratio 7:2:1.

The performance of TextRank model was measured by considering the first 15 , 25 and 50 sentences as rare. Results were compared with human labeled data and Table 10 shows the precision, recall and f_1 values for all the thresholds considered. The metrics (precision, recall and f_1) were calculated for a document and then averaged for all the contracts. From the Table 10 we can observe that TextRank with threshold as 15 performs the best.

Although anomaly detection techniques are famous for identifying rare components, its applica-

	Text Rank								
	P_{15}	R_{15}	$F1_{15}$	P_{25}	R_{25}	$F1_{25}$	P_{50}	R_{50}	$F1_{50}$
Employment	0.83	0.90	**0.87**	0.84	0.85	0.85	0.84	0.74	0.79
Incentive	0.90	0.91	**0.91**	0.90	0.85	0.87	0.9	0.71	0.79
Severance	0.82	0.84	**0.83**	0.82	0.74	0.78	0.82	0.52	0.64
Software License	0.83	0.88	**0.86**	0.83	0.82	0.82	0.82	0.64	0.72
Purchase	0.90	0.88	**0.89**	0.90	0.8	0.85	0.89	0.61	0.72

Table 10: Evaluation of TextRankScores

	TSI			TextRank			Lof Outlier		
	P	R	F_1	P	R	F_1	P	R	F_1
Employment	0.89	0.93	**0.911**	0.83	0.90	0.87	0.92	0.44	0.59
Incentive	0.93	0.98	**0.96**	0.9	0.91	0.91	0.91	0.28	0.44
Severance	0.84	0.98	**0.91**	0.82	0.84	0.83	0.85	0.21	0.33
Software Licence	0.89	0.99	**0.93**	0.83	0.88	0.86	0.86	0.26	0.36
Purchase	0.92	0.97	**0.94**	0.9	0.88	0.89	0.92	0.44	0.59

Table 11: Evaluation of TSI, TextRank and LoF

tions on legal data are less prevalent. The main idea of unsupervised anomaly detection algorithms is to detect data instances in a dataset, which deviate from the norm. However, there are a variety of cases in practice where this basic assumption does not hold true. The anomalies could be local, global or anomalous when compared with its close-by neighborhood and determining a single approach that would work well for all data instances is difficult.

Table 11 compares the metrics of TSI model, TextRank and LoF outlier with the human labels. From the table it can be observed that TSI model performs better than unsupervised TextRank and LOF approaches.

The TSI model performs better at identifying the rare sentences than the best TextRank model as it is designed based on the semantics and structural features of legal contracts.

8 Conclusion and Future work

Our work is an attempt to study the structure of contracts and harness the semantic "strictness" of these contracts in order to extract "significant" pieces of information contained therein. Here, significance is defined by two distinct ideas: rarity in a type of contract and commonality across types. We find that this view of contracts removes a need to review elements which are boilerplate and would, in turn, reduce the effort required to find critical content in a given contract. We show that our models can achieve reasonable accuracy with relatively low training data. This work can be extended in future to a query based model by taking input from the users in the form of a query and highlight text most relevant to a given query. Since the task is novel and there exists no parallel corpora, we wish to release sentences, that are rare and sentences that contain contract specific elements from the sampled contracts.

9 Limitations

Our study aims at capturing significant components of a legal contract with an emphasis on identifying information that is specific and unique to a contract. While the approach successfully highlights and identifies significant components, there were a few limitations. The dataset contains contracts as well as amendments made to the existing contracts. These amendments contribute to low coverage of contract elements. Increasing the data for each contract type might yield in better coverage and results.

References

Narendra Andhale and LA Bewoor. 2016. An overview of text summarization techniques. In *2016 international conference on computing communication control and automation (ICCUBEA)*, pages 1–7. IEEE.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ilias Chalkidis and Ion Androutsopoulos. 2017. A deep learning approach to contract element extraction. In *JURIX*, pages 155–164.

Ilias Chalkidis, Ion Androutsopoulos, and Achilleas Michos. 2017. Extracting contract elements. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 19–28.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*, pages 363–370.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *arXiv preprint arXiv:2103.06268*.

Zhiheng Huang and Wei Xu. 2015. Kai yu. *Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991*.

Prathamesh Kalamkar et al. 2021. Indian legal nlp benchmarks: A survey. *arXiv preprint arXiv:2107.06056*.

Rahul C Kore, Prachi Ray, Priyanka Lade, and Amit Nerurkar. 2020. Legal document summarization using nlp and ml techniques. *Int. J. Eng. Comput. Sci*, 9:25039–25046.

Sibusiso Kubeka and Abejide Ade-Ibijola. Automatic comprehension and summarisation of legal contracts. *contract*, 9:10.

Tuyen Le et al. 2020. Automated requirements identification from construction contract documents using natural language processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 12(2):04520009.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Sudip Mittal, Karuna P Joshi, Claudia Pearce, and Anupam Joshi. 2015. Parallelizing natural language techniques for knowledge extraction from cloud service level agreements. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2831–2833. IEEE.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*.

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

Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. 2016. A redundancy-aware sentence regression framework for extractive summarization. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 33–43, Osaka, Japan. The COLING 2016 Organizing Committee.

Paulo Silva, Carolina Gonçalves, Carolina Godinho, Nuno Antunes, and Marilia Curado. 2020. Using natural language processing to detect privacy violations in online contracts. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 1305–1307.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Puyang Xu and Ruhi Sarikaya. 2013. Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding*, pages 78–83. IEEE.

Dan Yang, Christina Leber, Luis Tari, Aravind Chandramouli, Andrew Crapo, Richard Messmer, and Steven Gustafson. 2013. A natural language processing and semantic-based system for contract analysis. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, pages 707–712. IEEE.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Ruixue Zhang, Wei Yang, Luyun Lin, Zhengkai Tu, Yuqing Xie, Zihang Fu, Yuhao Xie, Luchen Tan, Kun Xiong, and Jimmy Lin. 2020b. Rapid adaptation of bert for information extraction on domain-specific business documents. *arXiv preprint arXiv:2002.01861*.

Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 159–168.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.

Markus Zopf, Eneldo Loza Mencía, and Johannes Fürnkranz. 2018. Which scores to predict in sentence regression for text summarization? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1782–1791, New Orleans, Louisiana. Association for Computational Linguistics.