# Deeper Analysis and Comprehension of Documents including Contracts

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science*
*in*
*Computer Science and Engineering*
*by Research*

by

Hiranmai Sri Adibhatla
2018900044
hiranmai.sri@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2023

International Institute of Information Technology
Hyderabad, India

# CERTIFICATE

It is certified that the work contained in this thesis, titled “**Deeper Analysis and Comprehension of Documents including Contracts**” by Hiranmai Sri Adibhatla, has been carried out under my supervision and is not submitted elsewhere for a degree.

| | |
|---|---|
| Date | Adviser: Prof. Manish Shrivastava |

To Family and Friends

# Acknowledgments

This thesis is a culmination of the efforts, guidance, and support of numerous people around me. I would like to express my deep gratitude to my thesis advisor, Dr. Manish Shrivastava, for his invaluable guidance, insightful feedback, and unwavering support throughout the process. His expertise and mentorship have been instrumental in shaping the direction and quality of this work.
I would also like to thank the faculty and staff of IIIT Hyderabad for providing a stimulating academic environment and for their willingness to share their knowledge and expertise. I thank Dr. C.V Jawahar, Prof. Dipti Misra Sharma, Dr. Radhika Mamidi, and Dr. Vasudev Varma for their insightful lectures, assignments, and quizzes. The concepts I learned in the courses helped me enhance my technical skills, and the projects I completed provided the basis for my research work.
Furthermore, I am indebted to my family, friends, and my lab mates for their unwavering support, encouragement, and for their belief in me. Without their love and motivation, this thesis would not have been possible.

# Abstract

**Keywords:** Natural language processing, Document analysis, Legal Documents, Contracts, Information Extraction, Cause, Effect, Summarization

The exchange of information between humans involves the use of speech and text and is known as Natural Language. Each day, people communicate with each other in various languages through speech or text, sharing a vast amount of information. The data produced through natural language communication can offer valuable insights despite being ambiguous, unstructured, and noisy. However, computers cannot interpret natural language on their own yet. To fully comprehend this data and respond intelligently, computers need the ability to understand and emulate human language. This is where Natural Language Processing (NLP) comes in - it is a branch of Artificial Intelligence that focuses on enabling machines to read, comprehend and derive meaning from human languages. NLP integrates the disciplines of linguistics and computer science. It decodes language, its structure, rules, and creates models capable of comprehending, analyzing, and extracting important information from both text and speech.

An abundance of information is available in the form of text, including books, documents, articles, social media posts, and more. A document, one of the oldest forms of information exchange, refers to written, printed, or electronic material that is created to facilitate the exchange of information from its author to its intended audience. These files contain valuable information that can significantly benefit business activities. With the use of NLP applications, insights can be extracted from text data. Enterprises utilize NLP applications for various purposes, ranging from document understanding, information extraction, or providing answers to common questions. In this thesis, we develop techniques for deeper analysis and understanding of documents that are commonly used in enterprises.

A contract is a frequently used type of document in the corporate world. Contracts are agreements between two or more parties, that govern what each party can or cannot do and are usually dense in information. Automatically extracting key components or components that contain rare or novel information from these large documents makes reviewing contracts easier. Nevertheless, it can be a challenging task as the key and novel components are not present in isolation within the contract. Extraction of significant components (key components + novel components) from contracts aims to simplify the end user's comprehension and reduce dependency on legal experts for reviewing contracts. In this thesis, we introduce approaches for the automatic identification and extraction of significant components from a contract. We propose a Bidirectional Encoder Representations from Transformers (BERT) based model that automatically identifies or highlights significant components of a contract.

In the corporate world, reports are also a frequently encountered type of document. A report is a document that provides information and analysis on a particular topic or issue. Reports are used to convey important information to stakeholders, such as managers, executives, investors, and customers. The vast data available in these reports have the potential to revolutionize data-driven analysis. Causality identification and span detection is one such data-driven task. The relationship between two entities where one causes another event to happen is known as cause and effect. We explored various transformer-based models that help in classifying sentences as well as identifying spans in a sentence.

Both significant component extraction and causality identification tasks help us comprehend large documents and reports for downstream tasks and data-driven analysis.

# Contents

# List of Figures

Figure Page

# List of Tables

*Chapter 1*

# Introduction

A document is more than a collection of sentences. Its structure and organization are also crucial for comprehension and interpretation. While humans can easily make sense of written material, machines struggle with understanding natural language and processing the information contained therein. Even the human brain's mechanisms for comprehending language are not fully understood, making it challenging to replicate this process in machines. The most common document types are contracts and reports.

**Contract** is a legally binding agreement between two or more parties that establishes the rights and obligations of each party. Contracts are used to establish the terms of business relationships between companies, as well as to govern the relationship between a company and its employees, contractors, customers, and suppliers. Contracts can take many forms, such as employment agreements, sales agreements, licensing agreements, and service agreements. These contracts typically include provisions related to the scope of work, payment terms, intellectual property rights, warranties and representations, liability, and dispute resolution.

**Report** is a document that provides information and analysis on a particular topic or issue. Reports are typically used to convey important information to stakeholders, such as managers, executives, investors, and customers. Reports can take many forms, including written documents, presentations, and dashboards. Some of the common types of reports used in corporate offices include financial reports, sales reports, and project reports.

Although contracts and reports share a specified structure for organizing information, their purposes differ, requiring distinct techniques to comprehend them for downstream tasks.

## 1.1 Contract

Given that language is central to many domain-specific fields like medicine and law, it is not surprising that there has been an increasing interest in utilizing NLP applications to address a broader array of problems in recent years. The legal sector is being transformed by artificial intelligence (AI). The most recent developments in the legal field have primarily focused on areas such as contract element extraction, clause identification, legal document summarization, and judgment prediction. Legal work is known for being time-consuming, so any degree of automation can provide significant relief for lawyers.

Although AI adoption in legal is still in its early stages, lawyers now have access to a diverse range of intelligent tools powered by natural language processing (NLP). While NLP on its own is not an automation technology, it can facilitate automation in certain areas. NLP can assist in reducing the amount of time needed for completing certain tasks. Therefore, the development of NLP tools tailored to the legal field is highly necessary. The significance of word choice and syntax in law cannot be overstated. Even a slight ambiguity in a legal document, such as a contract, can result in unintended interpretations. Contracts are an essential tool for managing business relationships and minimizing risk in corporate transactions. It is important to ensure that contracts are drafted and reviewed carefully to ensure that they accurately reflect the parties intentions and protect their interests. The objective of this thesis is to establish the drive to create tools that can assist in contract review and enhance the understanding of end users by automatically identifying and highlighting significant components from information-dense documents.

### 1.1.1 Significant Components

In essence, “significant” pieces of information have:

1. information pertaining to material/practical details about a specific contract.

2. information that is novel or comes as a “surprise” for a specific type of contract.

The significance of a component is defined both 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 that deviate significantly from the norm for the type may be considered significant (type-specific information (TSI)).

### 1.1.2 Need for Significant Components Extraction

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 the significant components do not occur in isolation and 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 thesis, we introduce approaches for the automatic identification and extraction of significant components in documents.

### 1.1.3 Structure and Content of a Contract

When compared with the corpora on which most pre-trained deep models are based, the structure and vocabulary of texts in contracts differs significantly. Contracts frequently take constrained forms, sometimes even "template-like" for the sake of ensuring legal unambiguity. The structure of the document is defined by the relationships between the sentences it contains. 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 the same category, although the wording and sentence structure differ between individual contracts, the information conveyed remains almost the same,
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:

1. Templatised sentences
2. Boilerplate sentences
3. Rare sentences

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

#### 1.1.3.1 Templatised sentences

Templatised sentences are those that adhere to a specific template structure. Typically, in such sentences, a certain phrase or section may differ while the remaining content remains semantically identical throughout contracts. Examples include contract elements [12] such as the title of the contract, parties involved in the contract, dates, governing law, and more.

As observed in Table 1.1, 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 "effective 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.

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

**Table 1.1** Sentences with a Template Structure

#### 1.1.3.2 Boilerplate sentences

Boilerplate sentences are a set of standard phrases or formulations that are uniformly present in all contracts of a particular type. They make up a significant portion of the contract and are found in huge numbers. This work proposes an extension to the definition of boilerplate sentences to include those that convey the same semantic meaning across contracts but vary in lexical and structural composition. As illustrated in Table 1.2, these clauses are commonly used in contracts belonging to a specific type. Business and technical documents often rely on boilerplate sentences to enhance efficiency and standardize language and structure. In the case of boilerplate sentences, the degree of information divergence between contracts of a particular type remains relatively constant.

| **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 1.2** Sentences with Standardized Clauses

#### 1.1.3.3 Rare Sentences

In contracts, rare sentences are those that contain content that is not commonly found in contracts of that particular type, hence are conspicuous by their presence in the current contract. For instance, Table 1.3's first example mentions a hypothetical tax rate that applies to employees working on-site, while the second example states that no additional stock units are granted if there's a change in the organization's control. These clauses are situational and aren't typically found in most contracts of that category. Intuitively, these sentences are of interest to anyone examining the contract as they bring in novelty. Rare sentences are identified by the contract type to which they belong.

| **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 1.3** Sentences with Rare elements

In summary, a contract has

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

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

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

### 1.1.4 Significant component extraction (SConE) VS Summarization

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 [92] techniques would inadvertently change the semantics of the contract. Even when compared to an extractive setting [60], in this study, our main focus is to accentuate rare and templatised sentences as significant components in comparison to boilerplate sentences. We are not aware of any publicly available comparable corpora.

## 1.2 Reports

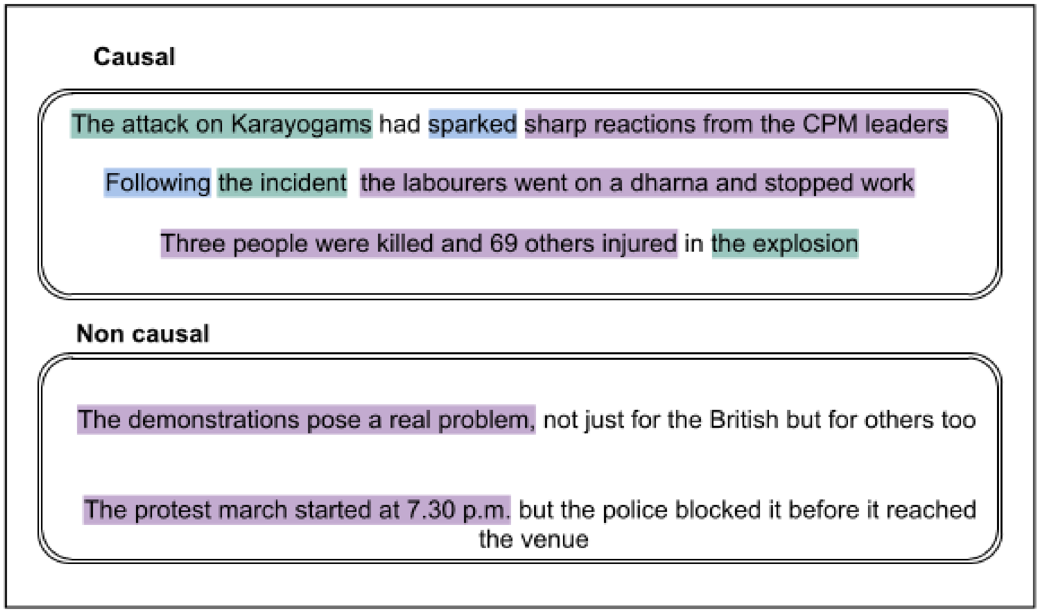
Reports are an essential tool for decision-making and communication in corporate offices. They help stakeholders stay informed, identify opportunities for improvement, and make strategic decisions for the future of the company. The vast information present in these reports has the potential to revolutionize data-driven analysis [4]. In order to make strategic decisions, it is imperative to recognize the relationship between the entities within a sentence. Different types of relations can exist between the entities of a sentence, such as causal, temporal, and conditional relations. These relations assist in enhancing our understanding, making predictions and inferences. Causality identification and span detection [19] is one such data-driven task. In this thesis, we identify the causal relations in socio-political events from news articles and the model can be extended to any document or report that contains sentences with cause and effects.

### 1.2.1 Causal Text Mining

Causal text mining involves identifying causal relationships within the text. It is one of the many natural language processing (NLP) studies that attempts to address inference and comprehension. A causal relation is a semantic relationship between cause argument and effect argument such that the occurrence of one contributes to the occurrence of the other.Figure 1.1 contains a few annotated examples from the causal news corpus. The causes are highlighted in green, the effects in purple, and the signals in cyan. Both cause and effect must be present in the same sentence to mark it as causal. The goal is to locate instances where causal information is present within a given input text. Once the causal information has been extracted, researchers can use it as a knowledge base for downstream tasks like summarization [40, 32], question answering [42, 26] and making inferences [21].

Cause is a span of text that results in the occurrence of an effect event. An effect is a span of text that is the consequence of the cause event and a signal is a span of text that binds both the cause and effect events. Together the study of cause and effect can help in understanding what agents contribute to an event and the effects they create. Along with reports, causality identification, and span detection find their use cases in many domains like climate and news.

In the climate science domain, it helps analyze the rapid climate changes [39]. Similarly analysis on financial domain [54] aids in better risk management, decision-making, customer segmentation, and market analysis. Further examples include socio-political news events where the effects created by causes such as a change in policy can be identified over a period of time



Causal

The attack on Karayogams had sparked sharp reactions from the CPM leaders

Following the incident the labourers went on a dharna and stopped work

Three people were killed and 69 others injured in the explosion

Non causal

The demonstrations pose a real problem, not just for the British but for others too

The protest march started at 7.30 p.m. but the police blocked it before it reached the venue

**Figure 1.1** Annotated examples from Causal News Corpus. Causes are in highlighted in green, Effects in purple and Signals in cyan.

and serve as critical information for analysts.

Since causality is an integral part of human cognition, causal text mining also has important downstream natural language understanding use cases. It also motivates and shows the need for causality detection models in the news domain.

## 1.3 Document Comprehension

Document comprehension and analysis encompasses comprehending the connections between sentences and paragraphs within a document, as well as inferring the general meaning of the document from its contents and title. Achieving this requires two important elements:

1. Sentence comprehension: comprehending the meaning of individual sentences within the document, both in isolation and in the context of the larger document.

2. Understanding the relationships between sentences: understanding the relationship between sentences within the document as well as comprehending the relationship between the different components of a sentence.

Both sentence comprehension and identifying relationships between sentences hinge on a critical step: sentence representation. Representation, in this context, involves converting text-based information into numerical formats, such as vectors or matrices, which can be processed by computers. Sentence representation involves encoding information from distributed word/sub-token representations into a single unit, representing the entire sentence.

### 1.3.1 Sentence Representation

A complete idea or thought is expressed by a sentence and is a fundamental building block of language. A sentence is composed of words that must adhere to specific rules, and a random grouping of words cannot form a valid sentence in a language. Humans apply different rules while constructing sentences to aid in creating coherent and significant statements. For comprehending a sentence, when using deep learning networks, the sentence is typically converted into a vector known as the ”sentence representation.” This representation is generated by calculating the numerical representations of the individual units of the sentence, which include words or character n-grams.

Word2Vec [57] and Glove [61] are two techniques that demonstrate how words can be expressed as vectors. However, FastText [8] takes a different approach by using n-grams as the basic building blocks. A word's representation is formed by averaging the representations of its constituent n-grams. To obtain a sentence representation, the sentence is first tokenized into individual units and then each unit is assigned a numerical representation. This process results in a two-dimensional matrix for each sentence. Additional information such as positional data can be included in this matrix if desired. Finally, this matrix is converted into a vector that represents the sentence. There are various methods for converting the matrix into a sentence vector. Some examples include encoding using CNN [62, 15, 94], LSTM Networks [34, 95], and Transformer layers [82, 18].

These sentence representations at the end are utilized as inputs for NLP models to perform a range of downstream tasks.

## 1.4 Contributions

This thesis examines the comprehension and analysis of documents and articles in a comprehensive manner. The contributions made in this regard are outlined below:

1. Identifying the features of a sentence and classifying them into three broad categories namely Templatized, Boilerplate, and Rare.

2. Highlighting and extracting significant (key + novel) components from contracts.

3. Analyze clauses in a sentence and their dependencies for classifying them as the cause, effect, or signal and also identifying spans.

## 1.5 Thesis Outline

The thesis is structured into five distinct chapters, each chapter independent of the other. The organization of the thesis is as follows:

1. In Chapter 1, we introduce and present the problems that we aim to tackle. We outline the motivation and need for confronting these problems, and provide a brief overview of the approaches taken to address them.
2. In Chapter 2, we describe efforts and related work on legal domain and causality detection.
3. In Chapter 3, we detail our efforts in identifying significant components of a contract using BERT-based methods.
4. In Chapter 4, we describe our various attempts in identifying causes and effects in a sentence.
5. In Chapter 5, we conclude the document, summarizing its contributions and providing an overview of potential future enhancements.

*Chapter 2*

# Literature Review

NLP, a subset of AI, assists machines in comprehending and processing human language, enabling them to perform tasks automatically. Some of the downstream applications of NLP are machine translation, summarization, question-answering systems, sentiment analysis, classification, and spell check. NLP systems typically take in the text as input. The input text is first transformed into a vector representation of real values. This vector is processed by a model that produces text or a class label, depending on the downstream application.

This thesis presents two tasks that facilitate the analysis and comprehension of contacts and report documents. The first task involves extracting significant components from contracts, while the second task involves identifying and determining causal relationships in a report. In this section, we discuss the work progress on contracts and causal text mining from reports.

## 2.1 Related Work on Contract Analysis

In recent years, there has been an increase in research activity focused on document and text processing. This surge in interest has led to the development of numerous datasets, tasks, and applications, including but not limited to prior case retrieval [2, 41], summarization [7], events and named entities extraction [43, 47], and judgment prediction [87, 11, 53].

### 2.1.1 Contract Elements Extraction

A considerable amount of work has been done in contract analysis and information extraction from contracts [89, 72, 58]. Extracting contract elements from legal documents, such as contracts, has been a long-standing challenge for the legal and artificial intelligence (AI) communities. The extraction of contract elements can include identifying parties, dates, payment terms, obligations, warranties, and other relevant provisions that are essential for understanding the content of a contract. Contract elements extraction is well studied and the most obvious approach to automatic contract element extraction is to model it as a sequence labeling task.

#### 2.1.1.1 Rule-based approaches

One of the earliest approaches to contract element extraction involved using rule-based methods, where experts manually created rules with hand-crafted features, word embeddings, and part-of-speech tag embeddings to identify and extract specific elements from contracts [12]. While these methods were effective in certain cases, they were limited by their inability to handle the complexity and variability of natural language. This is likely because they use linear classifiers, which may not be able to capture complex relationships between the input features and output labels. The other main issue with this approach is the encoded form of the contracts, where each vocabulary word has been replaced by a unique integer. This could limit the effectiveness of specific methods that rely on character-level embeddings. Additionally, not being able to study the actual texts of the contracts could limit the ability to gain insights into the language used in legal documents and may also hinder the ability to identify and address any biases or inconsistencies in the dataset.

#### 2.1.1.2 Statistical approaches

Use of deep learning methods for contract element extraction [10] along with conditional Random Fields [23, 88] was popular for sequence labeling tasks prior to neural networks. The authors proposed a Bidirectional LSTM (BILSTM) model that operates on word, part-of-speech (POS) tag, and token shape embeddings. This model was tested against the linear sliding-window classifiers [12] that were proposed in their previous work. They found that the BiLSTM [27] model outperformed the linear classifiers, without the need for any manually written rules. The authors further improved the BILSTM model by adding an additional LSTM layer on top of it, which reduced the number of misclassified tokens. They also experimented with adding a Conditional Random Field (CRF) layer on top of the BILSTM model, which resulted in even better performance, particularly in cases where multi-token contract elements needed to be extracted. The advantages of this approach are that it does not rely on manually written rules and it can handle multi-token contract elements. Additionally, the use of deep learning models allows for more complex and flexible feature representations that can capture important patterns in the data.

#### 2.1.1.3 Neural Netwok based approaches

Recently, neural networks [35, 52, 11] based approaches were employed for contract elements extraction. This approach [11] investigates the task of contract element extraction and compares the performance of several neural network-based models such as LSTM-based encoders, dilated CNNs, Transformers, and BERT. The study finds that LSTM-based encoders perform better than the other models for this task, and domain-specific WORD2VEC embeddings outperform generic pre-trained GLOVE embeddings. The study also highlights the importance of task-specific choices, as several choices that work well for generic sequence labeling tasks do not improve performance for contract element extraction. However, the study does not provide a comprehensive analysis of the reasons for the observed performance differences between the different models and embeddings. Further investigation and analysis could provide more insights into the underlying factors that influence the performance of contract element extraction methods.

BERT [18] based approaches [93, 14] were developed for sequence contracts elements extraction. Formulating contract elements and as a sequence labeling task by adapting BERT [93] uses BERT, a state-of-the-art language model, which shows promising results in extracting important content elements from business documents. The study demonstrates that even with a modest amount of annotated data, the model can achieve reasonable accuracy, which is valuable for practical applications. The development of an end-to-end cloud platform that provides an easy-to-use annotation interface and an inference interface for users to upload documents and inspect model outputs is useful for addressing real-world business needs. The model is limited by its focus on only two types of business documents, regulatory filings, and property lease agreements, and may not generalize to other types of business documents. The study also fails to compare the performance of BERT to other models or approaches for content extraction from business documents. A joint intent classification and slot-filling model based on BERT [14] shows significant improvements in intent classification accuracy, slot-filling F1 score, and sentence-level semantic frame accuracy on several public benchmark datasets, compared to other models like attention-based recurrent neural networks and slot-gated models. The results demonstrate that the use of BERT in natural language understanding tasks can effectively address the problem of small-scale human-labeled training data, resulting in improved generalization capability, especially for rare words. The study does not explore the potential limitations or challenges of using BERT for joint intent classification and slot filling, such as computational complexity or model interpretability. The experiments are conducted on a limited set of public benchmark datasets, and the results may not generalize to other domains or languages. All the above studies show that contract elements can be extracted with a decent accuracy from contracts with little annotated data.

### 2.1.2 Clause Analysis

The introduction of the Contract Understanding Atticus Dataset (CUAD) [] addresses the bottleneck of requiring expensive expert annotators for many specialized domains that remain untouched by deep learning. The dataset consists of over 13,000 contracts and 500 annotations by legal experts across 41 labels, making it a valuable resource for research on contract review and for understanding how well NLP models can perform in highly specialized domains. The task is formulated to predict the substrings of a contract related to each label category. The task only focuses on predicting the start and end token positions of the substring of each segment that should be highlighted. This may not capture all the relevant information in the contract and may have annotation biases.

### 2.1.3 Extracting rights and obligations

The paper [24] addresses a practical problem of contract understanding and proposes a solution in the form of an annotated corpus for language processing systems to recognize rights and obligations in contracts. The corpus is built based on a substantial number of real English and Japanese contracts, which increases its usefulness for real-world applications. The study lacks an analysis of the effectiveness of the annotated corpus in improving the performance of language processing systems for contract understanding and making inferences from contracts.

### 2.1.4 Contract NLI

Document-level natural language inference (NLI) [46] system for contracts has the potential to significantly reduce the time and cost associated with contract review, making it more accessible to individuals and companies. The proposed NLI system may not be able to capture all the nuances and context-specific meanings in contracts, which could lead to errors and misinterpretations. The dataset used to train and evaluate the system consists of only 607 contracts, which may not be representative of all types of contracts.

### 2.1.5 Clause Recommendation

There is comparatively less emphasis on generating contracts. There is a gap in NLP research on contracts for clause recommendation to aid in contract authoring [1]. The proposed pipeline uses BERT, a state-of-the-art pre-trained model, for prediction and recommendation. The study does not address ethical or legal concerns that may arise with the automation of legal tasks and the evaluation is limited to a few clause types which may not generalize well to other types of contracts.

Our work, focused on identifying sentences in a contract that deviate from the norm, would add depth to contract analysis along with contract elements extraction, clause identification, and inference tasks. Identifying novel or rare clauses will help alert legal teams to the presence of such sentences without the burden of going through each sentence of the contract. This is important as these statements legally bind the contracting parties.

## 2.2 Related Work on Causal text mining

Causality has been extensively studied across a wide range of fields, such as Psychology [86, 85], Linguistics [74, 33], Philosophy [84], and Computer Science. The main objective of causal text mining is to identify whether an object, an event, or a series of events can be deemed as the cause of a preceding event. One of the most straightforward ways to convey cause-and-effect relationships is by using propositions such as ‘A causes B’ or ‘A is caused by B’. Causality can be expressed using various types of propositions (e.g., active, passive, subject-object, nominal, or verbal) and has diverse syntactic representations. The current body of research on the extraction of causal relationships can be broadly categorized into two groups: (1) methods that rely solely on linguistic, syntactic, and semantic pattern matching, and (2) techniques that utilize statistical approaches and machine learning. The primary aim of this collaborative task is to ascertain whether a quantifiable fact is linked to a causality. Extracting causal information from the text can be a crucial step for many downstream natural language processing (NLP) tasks.

### 2.2.1 Linguistic approaches

A reliable causal analysis tool should be based on a knowledge representation model that includes a semantic framework and domain knowledge to correctly identify significant causal relations from any text. An early work on causal analysis [44] involved encoding the text into prepositions consisting of a verb predicate and its syntactic role-marked arguments. The study utilized hand-coded features of prepositions that indicate causality, such as ”because” and ”due to,” and also incorporated manually constructed constraints for domain-specific concepts that may imply causality. In addition, the model identified cause-effect pairs [20] by examining pairs that were both temporally and spatially related. his approach of knowledge-based inference has several limitations. It heavily relies on manual pre-processing, including the encoding of the text into propositions and the creation of propositional clues to detect explicit causal relations, as well as the design of constraints to identify implicit causalities. Furthermore, these pre-processing steps are mostly domain-specific, requiring expert knowledge to manually design a limited set of clues and constraints. Automatically extracting causal relations from the text by combining syntactic and semantic features along with explicit causal indicators [45] has proven to improve precision in information extraction and retrieval tasks. The proposed model was evaluated on articles from Wall Street Journal. Due to its heavy reliance on explicit causal indicators, the model was unable to handle implicit causalities and lexical ambiguities that arise from domain independence design.

### 2.2.2 Statistical and Machine Learning approaches

The requirement for a significant amount of labeled textual data that is independent of domain and type, along with the need to automatically extract implicit patterns from text containing ambiguous constructions, pave way for machine learning techniques that have the potential to improve performance over purely linguistic methods. It led to models constructed with sophisticated feature engineering, and carefully designed and adjusted handcrafted features that incorporate rich lexical, syntactic, or semantic features [6, 73]. The success of these models were heavily dependent on the tool kits used (POS tagger, dependence parser, and named entity parser) and the errors from the tool kits were propagated to the causality extraction models. Deep learning has become increasingly popular in recent years and minimized the need for complicated feature engineering and external NLP tool kits. The Convolutional Neural Network (CNN) [62] is recognized as one of the most prominent deep learning models for relation extraction. In comparison to rule-based and rich feature-based methods, CNN models with pre-trained word embeddings [8] that encode semantic and syntactic information of words into fixed-length vectors have demonstrated superior effectiveness in extracting complex causal relations. Semantic lexicons like Wordnet [31], Verbnet [71], and Framenet [3] have proven to be useful for deep learning approaches as they can be used for feature extraction.

### 2.2.3 Applications of Causal text mining

By identifying the causal relationships between different events or concepts in a text, NLP models can generate summaries that highlight the most important causal links, or make predictions [28, 65] about what might happen in the future based on past causal events.

Causal information can be used to answer questions about the cause-and-effect relationships between different entities or events [29, 16]. For example, given a passage about a medical condition, an NLP model might be able to answer questions based on the causes the condition or effects of the condition.

Understanding causal relationships between different entities or events can help NLP models make more accurate inferences about the meaning of a text [21]. The ability to recognize causal relationships [25] within and between sentences can help illuminate the outcomes of significant events, particularly in cases where these relationships are not explicitly stated in the document. The utilization of external descriptive knowledge and relational graphs [9] has been proven efficient. There are ongoing efforts to develop benchmark datasets [76, 79, 55] for mining causal relationships in text.

In summary, extracting causal information from text is an important task that can help improve the accuracy and usefulness of many NLP applications.

*Chapter 3*

# Extraction of Significant Components

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 has

1. information pertaining to material/practical details about a specific contract

2. information that is novel or comes as a “surprise” for a specific type of contract.

It indicates that the 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:

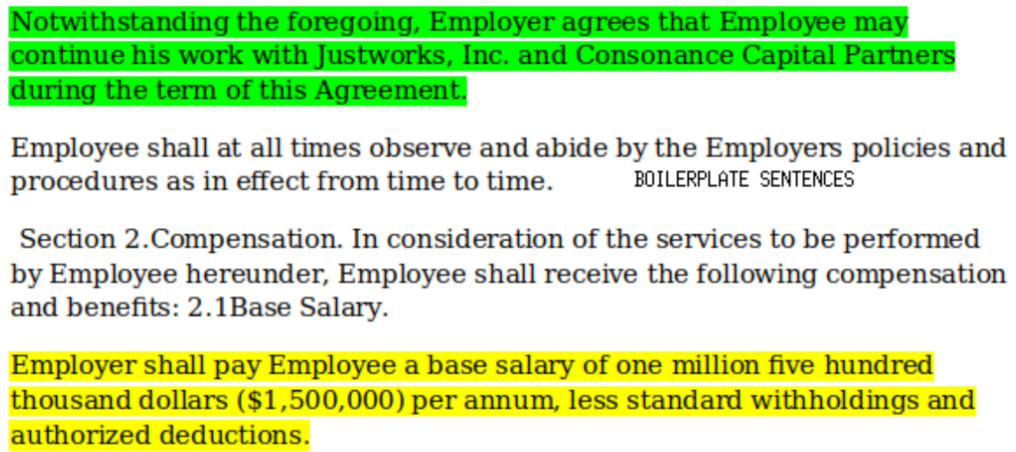
1. identifying CSI using a BERT based contract-specific information extractor.

2. identifying TSI by scoring sentences in a contract for their likelihood.

In this work, 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. The outcome of our approach is presented in two formats:

a *highlighted input document*, where sections of interest are highlighted within the overall contract(Figure 3.1). This helps in vizualizing the significant components of the contract.

b *cover-page*, a consolidated page containing the extracted significant components.



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.

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

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 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.

## 3.1 Approach

Drawing from our observations that legal contracts of the same category contain recurring 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 were calculated using LaBSE [22] while a BERT-based [18] regression model was adapted to learn and predict these likelihood scores. In the regression technique, sentences are evaluated and scored based on their significance, and the model acquires the ability to incorporate sentences into a summary [96] by predicting these scores.

Significant component extraction is accomplished in two stages:

1. Identifying CSI by processing each sentence of the document and identifying sentences with contract elements [12].

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

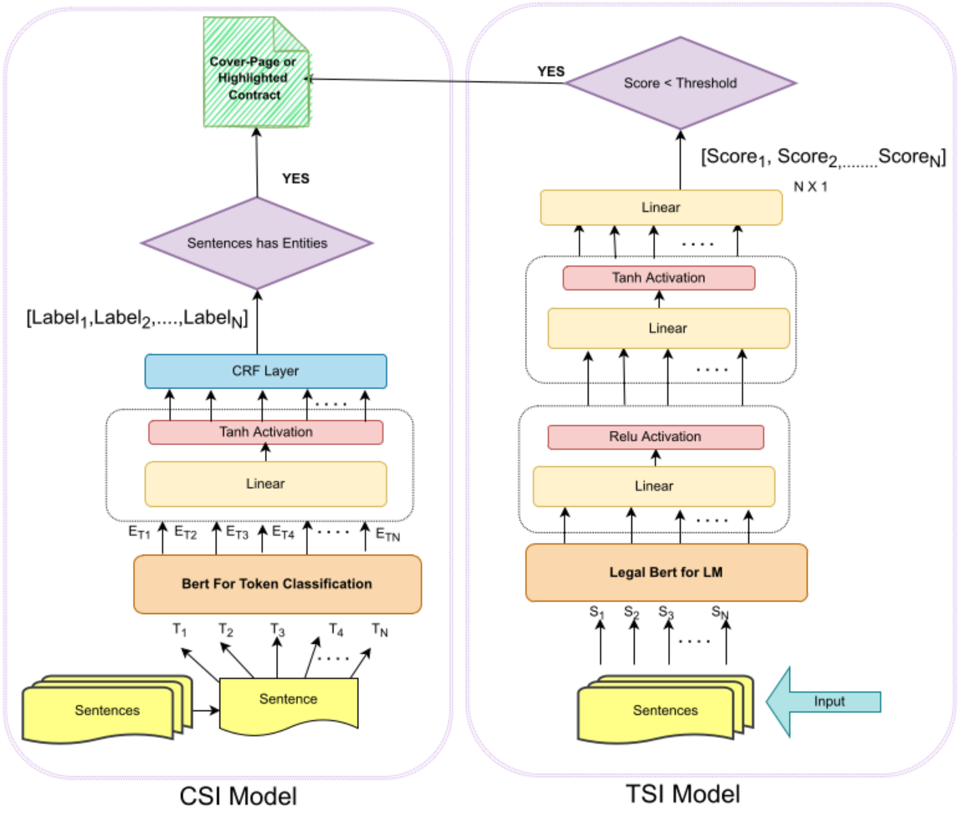


**Figure 3.2** SCoNE Architecture for CSI and TSI

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 [13] which is fine-tuned on BERT [18] 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 3.6.

The overall architecture is shown in Figure 3.2. 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 $P \in S$.

## 3.2 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 them 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, organizations, or locations 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 that contain contract elements, yields better results.

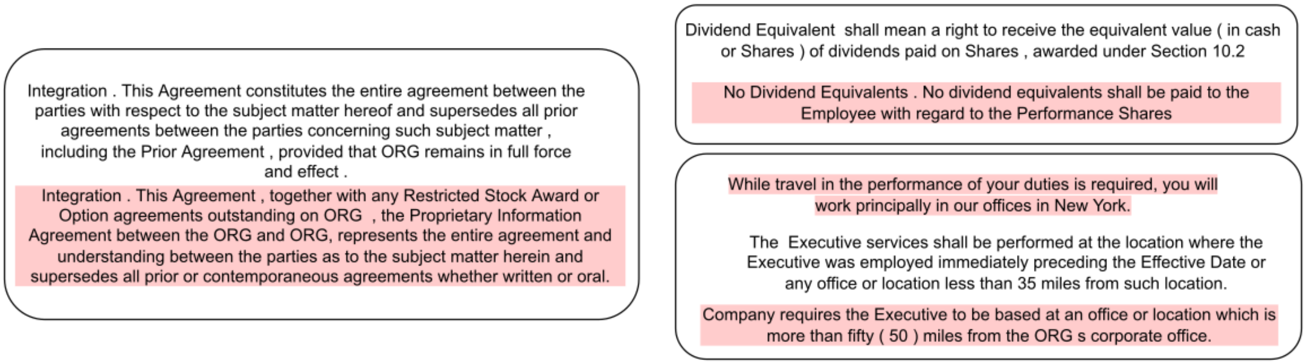
We sampled 500 legal documents (100 documents of each category mentioned in Table 3.6). These documents are then pre-processed into paragraphs. A paragraph as a unit might be of a higher value than an isolated sentence. The documents are split into train, test, and validation bins in the ratio 7:2:1. 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 [81]. 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 model's generality. The annotated contract elements are listed in Table 3.4.

### 3.2.1 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.

## 3.3 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 3.3 highlights a few clauses that may seem ordinary but are different from their usual construction in contracts.



Integration . This Agreement constitutes the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior agreements between the parties concerning such subject matter , including the Prior Agreement , provided that ORG remains in full force and effect .

Integration . This Agreement , together with any Restricted Stock Award or Option agreements outstanding on ORG , the Proprietary Information Agreement between the ORG and ORG, represents the entire agreement and understanding between the parties as to the subject matter herein and supersedes all prior or contemporaneous agreements whether written or oral.

Dividend Equivalent shall mean a right to receive the equivalent value ( in cash or Shares ) of dividends paid on Shares , awarded under Section 10.2

No Dividend Equivalents . No dividend equivalents shall be paid to the Employee with regard to the Performance Shares

While travel in the performance of your duties is required, you will work principally in our offices in New York.

The Executive services shall be performed at the location where the Executive was employed immediately preceding the Effective Date or any office or location less than 35 miles from such location.

Company requires the Executive to be based at an office or location which is more than fifty ( 50 ) miles from the ORG s corporate office.

**Figure 3.3** Novel sentences snippets, highlighted in pink

Based on our observations, legal contracts of the same category have repetitive information (Boilerplate). This requires ranking the sentences based on a metric for rarity. Scoring sentences [67, 96] based on both importance and redundancy among sentences was attempted for summarization [60] tasks. The approach, however, does not guarantee the 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 [56] 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 subunits of text by identifying relations among them. The Local Outlier Factor (LOF) algorithm [63] is an unsupervised anomaly detection method that computes the deviation of a given data point with respect to its neighbors. We applied both TextRank and LOF algorithms 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 the TSI scores with respect to the contract type and use these scores to identify TSI. The TSI score of a sentence here indicates the confidence with which a given sentence is part of a specific contract type.

| 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 3.1** Sentence Likelihood Scores (lscore) with and without Masking Entities

### 3.3.1 Masking Sentences

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 [12]. In order to ignore these sentences and to make sentences more comparable across contracts we mask all the named entities in contracts using spaCy[1] 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 3.1 shows examples of a few sentences whose TSI score has increased after masking named entities.

### 3.3.2 Mean-Max Pooling

Though contracts of a type contain repetitive information, the vocabulary, and structure might change. 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 [22]. We consider the maximum semantic overlap depicted by LaBSE as the indicator of semantic presence of the

[1] http://spacy.io

concept expressed by a sentence. Thus, we are approximating the expected count of a sentence (concept) occurring for a type using the LaBSE score as a proxy.

Let, $S_{\mathrm{ij}}$ be the $j^{\mathrm{th}}$ sentence in document $D_{\mathrm{i}}$ and $S_{\mathrm{kl}}$ be the $l^{\mathrm{th}}$ sentence in document $D_{\mathrm{k}}$. 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,

$$Count_k(S_{\mathrm{ij}}) = \max_{1 \leq l \leq p}(LaBSE(S_{\mathrm{ij}}, S_{\mathrm{kl}})) \tag{3.1}$$

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

The "count" obtained for the sentence $\mathrm{S_{ij}}$ 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 $S_{\mathrm{ij}}$ is calculated using Equation 3.2.

$$TSIScore(S_{\mathrm{ij}}) = \frac{(\sum_{i=1}^{N} Count_k(S_{\mathrm{ij}})}{N} \tag{3.2}$$

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 a low mean similarity score i.e, a low likelihood score.

### 3.3.3 Likelihood Approximation Model (TSI-A)

The process described in section 3.3.2 for calculating the 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 3.2) refers to contracts distribution in a 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 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 that serves as the sentence likelihood score. The model is trained for 15 epochs with a learning rate of 1e-05. The loss criteria is MSE (Means Squared Error) and the objective is to minimize the loss between the predicted scores and the training scores. Pearson correlation scores are calculated between the test scores generated by using Equation 3.2 and the trained BERT regression model.

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

**Table 3.2** Contract distribution

## 3.4 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 **??**. 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 3.3 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 3.3** Human Evaluation Statistics

## 3.5 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]).

### 3.5.1 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 3.4** Evaluation of Contract Elements

The table 3.4 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 3.5 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.

### 3.5.2 Pearson Correlation Evaluation

Fig 3.4 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 3.6 under contract types. From the plot we inferred that likelihood scores of sentences follow a trend. For all

| 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 3.5** 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 3.6** Averaged K-Fold Validation for Pearson Correlation of test and predicted likelihood scores

contract types, there exists sentences that have low likelihood and sentences which are more probable.

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 3.1 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.

$$C = \frac{(N\sum_{i=1}^{N} T_{\mathrm{i}}P_{\mathrm{i}} - (\sum_{i=1}^{N} T_{\mathrm{i}})(\sum_{i=1}^{N} P_{\mathrm{i}}))}{\sqrt{N\sum_{i=1}^{N} {T_{\mathrm{i}}}^2 - (\sum_{i=1}^{N} T_{\mathrm{i}})^2}\sqrt{N\sum_{i=1}^{N} {P_{\mathrm{i}}}^2 - (\sum_{i=1}^{N} P_{\mathrm{i}})^2}} \tag{3.3}$$

To measure the performance of our proposed model in predicting the likelihood score, we compute the Pearson product-moment correlation $(C)$ [5] between likelihood scores computed

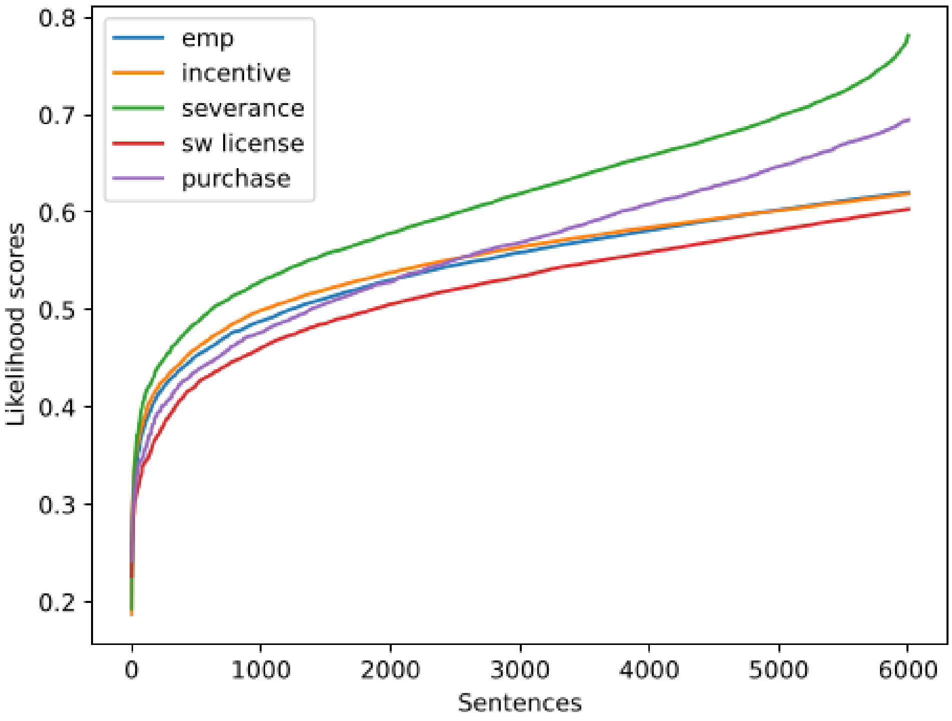


**Figure 3.4** Sorted Likelihood Scores of Sentences

by mean pooling LaBSE similarity scores (calculated using Equation 3.3) $(T)$ and likelihood scores generated by the TSI-A model $(P)$, for a sample of 10000 sentences $(N)$ using (Equation 3.3). 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 3.6 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.

### 3.5.3 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

|  | Text Rank |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | $P_{15}$ | $R_{15}$ | $F_{1\,15}$ | $P_{25}$ | $R_{25}$ | $F_{1\,25}$ | $P_{50}$ | $R_{50}$ | $F_{1\,50}$ |
| 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 3.7** Evaluation of TextRankScores

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 3.7 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 3.7 we can observe that TextRank with threshold as 15 performs the best.

|  | 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 3.8** Evaluation of TSI, TextRank, and LoF

Although anomaly detection techniques are famous for identifying rare components, its applications 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 3.8 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 rare sentences than the best TextRank model as it is designed based on the semantics and structural features of legal contracts.

## 3.6 Conclusion

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: a rarity in a type of contract and commonality across types. We find that this view of contracts removes a need to review elements that 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 the future to a query-based model by taking input from the users in the form of a query and highlighting text most relevant to a given query. Since the task is novel and there exist no parallel corpora, we wish to release sentences, that are rare and sentences that contain contract-specific elements from the sampled contracts.

## 3.7 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.

## *Chapter 4*

# Causality Detection and Identification

Causality detection and identification is centered on identifying semantic and cognitive connections in a sentence. In this work, we describe the effort of team LTRC for Causal News Corpus - Event Causality Shared Task 2022 at the 5th Workshop on Challenges and Applications of Automated Extraction of Sociopolitical Events from Text (CASE 2022) [77]. The shared task consisted of two subtasks:

1. identifying if a sentence contains a causality relation,
2. identifying spans of text that correspond to cause, effect, and signals.

We fine-tuned transformer-based models with adapters for both subtasks. Our best-performing models obtained a binary F1 score of 0.853 on held-out data for subtask 1 and a macro F1 score of 0.032 on held-out data for subtask 2. Our approach is ranked third in subtask 1 and fourth in subtask 2. The paper describes our experiments, solutions, and analysis in detail.

The causal news corpus [75, 78] comprises 3,559 event sentences, extracted from protest event news, that have been annotated with sequence labels on whether it contains causal relations or not. Subsequently, causal sentences are annotated with cause, effect, and signal spans.

For both tasks, we use a Transformer-based model [82]. We use adapters [64], a parameter-efficient fine-tuning method, in conjunction with a pre-trained model with strong language understanding and generation abilities [51]. Recent research has shown that this method is robust to over-fitting in low-resource settings [30]. In this way, the large pre-trained model RoBERTa remains frozen, and only small modules of the model parameters are optimized. This effectively retains acquired knowledge in the pre-trained language model. The first task was treated as a binary classification task with a single label for the input sentence, while for the second task, the label was predicted for each input word of the sentence.

## 4.1 Dataset

The data consists of English news in the socio-political and crisis context, extracted from Automated Extraction of Socio-political Events from News (AESPEN) in 2020 [38] and Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) in 2021 [37] .

Both cause and effect must be present in the same sentence to mark it as causal. The organizers made 3 datasets available for both the subtasks: *train, dev, and test*. Later UniCausal, a Causal Text Mining data [80] was released to be used for both subtasks. The labels for test data were not announced for both subtasks.

For subtask 1, around 869 news documents and 3559 English sentences were annotated with labels on whether they contained causal relations. Table 4.1 presents the sentence counts per data split.

| | Labels | | |
|---|---|---|---|
| | Causal | Non-causal | Total |
| Train | 1603 | 1322 | 2925 |
| Dev | 178 | 145 | 323 |
| Test | 176 | 135 | 311 |
| Total | 1975 | 1602 | 3559 |

**Table 4.1** Data split for sentences in subtask 1

For subtask 2, positive causal sentences from subtask 1 were retained and annotated with cause-effect-signal spans. From the total positive sentences, 180 sentences were annotated and there could be multiple relations per sentence. The data splits were: 130 train and 13 development.

After combining the causal news corpus and UniCausal corpus, the total number of unique samples on adding train and dev datasets is 6767 for subtask 1 and 1249 for subtask 2. We used 20% of the combined dataset for validation.

## 4.2 Approach

Transformer-based language models [82] that have been pre-trained on massive amounts of text data and then fine-tuned on target tasks has resulted in significant advances in NLP [50, 90], with state-of-the-art results across the board. However, models like BERT [18] and

RoBERTa [51] have millions of parameters, making sharing and distributing fully fine-tuned models for each individual downstream task prohibitively expensive.

Adapters [64], which consist of only a small set of newly introduced parameters at each transformer layer, is a lightweight alternative to full model fine-tuning. Because of their modularity and compact size, adapters overcome several limitations associated with full model fine-tuning: they are parameter-efficient, they speed up training iterations, and they are shareable and composable. Furthermore, adapters typically outperform state-of-the-art full fine-tuning [68].

### 4.2.1 Subtask 1

Three transformers-based language models [82] were considered for subtask 1 and fine-tuned on the causal news corpus dataset. The models experimented are BART (large) [48], RoBERTa (base and large) [17] with an additional linear layer on top, RoBERTa (base and large) with adapter [64] and a classification head. Adapters are small learned bottleneck layers inserted within each layer of a pre-trained model to avoid full fine-tuning of the entire model. The adapters framework enables them to be small, and scalable, particularly in low-resource scenarios. It freezes all weights of the pre-trained model so only the adapter weights are updated during training. It activates the adapter and the prediction head such that both are used in every forward pass. As NLP tasks become more complex and necessitate knowledge that is not readily available in a pre-trained model [69], adapters will provide a plethora of additional sources of relevant information that can be easily combined in a modular and efficient manner. We added a task-specific layer which is a classification head adapter. RoBERTa with a classification adapter head and a linear layer added on top of RoBERTa (base) performed better than the BART-large model.

### 4.2.2 Subtask 2

Subtask 2 was modeled as a token classification task in the lines of named entity recognition [49, 59] and parts-of-speech tagging [70, 83]. Each token of the cause-effect sentence should be labeled as either cause, effect, signal, or other. In the annotated data shared, a span of text for cause was between ARG0 opening and closing tags, a span of text for effect was between ARG1 closing and opening tags, and a span of text corresponding to signal enclosed between SIG0 opening and closing tags. The labeled annotations were pre-processed to be written in the Inside-Outside-Beginning (IOB) format [66] to aid in the identification of the sequences during inference. BertForTokenClassification model from BERT (base) [18] was used for obtaining the contextual embeddings for the token and trained to predict the most probable label sequence. Since we saw a slight boost in performance on using adapters, we added an adapter head to RoBERTa (base) to predict the label sequence. In spite of using IOB format and contextual embeddings of BERT in modeling the problem as a token labeling task, inference of predicted labels is difficult. A limitation that the model has is, that it can make an incorrect prediction in the middle of a cause/effect sequence or predict a cause/effect token in the middle of O tags. Few heuristics were employed to address the issue:

1. If a cause or effect sequence has a length lower than 2, it is ignored.
2. If a token is being preceded by a beginning-tag[1] and followed by either 'O' (for other) or the inside-tag [2], then the label is changed to its corresponding inside-tag.
3. If a token is predicted as (other) 'O', the sequence length of 'O' is less than 2, and is surrounded by beginning and inside tags of a single kind, then the label is changed to its corresponding inside-tag.
4. If a token is predicted as a beginning or inside tag of a kind, the whole sequence length is less than 2 and is surrounded by beginning and inside tags of another category, then the current category is changed to match the surrounding labels.

## 4.3 Evaluation and Experimental Setup

We fine-tune pre-trained transformers: BERT, BART and RoBERTa provided by huggingface [3]. The maximum sequence length for base models was 256 and for 512 for large model. The learning rate was 1e-4 and the models were fine tuned for 10 epochs for subtask 1 and 20 epochs for subtask 2. Adam optimizer was used with a dropout of 0.2 in each transformer layers. The train and validation batch sizes are 8 and 4 respectively.

### 4.3.1 Results

Table 4.2 shows the performance of our transformer based models for subtask 1 on the dev data set. All the transformer variants have surpassed the baseline scores. RoBERTa (base) with adapters was our best-performing model. The slight improvement in precision and F1 scores

[1] The beginning-tags could be B-E for effect, B-C for cause and B-S for signal

[2] The inside-tags could be I-E for effect, I-C for cause and I-S for signal

[3] https://huggingface.co/docs/transformers

| Model | R | P | F1 |
|---|---|---|---|
| BART-large | 0.85 | 0.81 | 0.84 |
| RoBERTa-large+Adapter | 0.82 | 0.84 | 0.83 |
| RoBERTa-base+Adapter | **0.87** | **0.86** | **0.87** |
| RoBERTa-base+linear layer | 0.86 | 0.83 | 0.84 |
| Baseline | 0.86 | 0.80 | 0.83 |

**Table 4.2** Performance on Devset for subtask 1

| Model | R | P | F1 |
|---|---|---|---|
| BERT+Adapter | **0.056** | **0.023** | **0.032** |
| Baseline | 0.003 | 0.009 | 0.005 |

**Table 4.3** Performance on Devset for subtask 2

for RoBERTa (base) with adapters over RoBERTa base with linear layer could be because, in the adapters framework, the adapters are added within each transformer layer while in the other approach, the linear layer is added to the output of the last layer of RoBERTa.

Table 4.3 shows the results obtained by using adapter on BERT (base). the predictions were post edited employing the heuristics discussed above. The results have improved marginally over the baseline model.

### 4.3.2 Error Analysis

| **Samples** | **Actual Label** | **Predicted Label** |
|---|---|---|
| The one-day fast attracted a “motley crowd” according to Sumitra M. Gautama, a teacher with the Krishnamurthi Foundation of India ( KFI ) | 1 | 0 |
| Both sides were raining bombs on each other and Mondal was hit by one of the bombs , Murshidabad district magistrate Pervez Ahmed Siddiqui said . | 1 | 0 |
| Another ‘TP’ issue may also leave a blot on the CPM , as public opinion is heavily pitted against the assault made upon former diplomat T P Srinivasan by SFI activists | 0 | 1 |
| The police did not grant a permit for the march – the second time authorities have rejected a protest request – following a ban on the Saturday rally in Yuen Long | 0 | 1 |

**Table 4.4** Misclassified samples of subtask 1

While reviewing and analyzing the errors made by our models, we discovered few patterns where the models failed. Table 4.4 shows a few samples that were misclassified for subtask 1. We observed that the model fails to identify effects and causes that are not explicit. For

| Samples | Predicted Cause | Actual Cause | Predicted Effect | Actual Effect |
|---|---|---|---|---|
| The treating doctors said Sangram lost around lost around 5 kg due to the hunger strike. | due to the hunger strike | due to the hunger strike | The treating doctors said Sangram lost around 5 kg | Sangram lost around 5 kg |
| The Sadtu protest was a call for the resignation of Motshekga and her director general Bobby Soobrayan. | resignation of Motshekga | the resignation of Motshekga and her director general Bobby Soobrayan | The Sadtu protest was a call | The Sadtu protest |
| Troops also killed two militants making infiltration bids in Gurez sector today. | making infiltration bids in Gurez sector | making infiltration bids in Gurez sector today | Troops ('also' predicted as 'O') killed two militants | Troops also killed two militants |

**Table 4.5** Misclassified samples of subtask 2

the first example in Table 4.4, the effect is*"attracted a motley crowd"* and the cause *"the one-day fast"*. The cause phrase contains polysemous word *"fast"*, that could be misleading. In the second example *"raining bombs"* is a simile and in NLP tasks similies, idioms and proverbs have always been tough to comprehend. The model fails to identify phrases with length of less than four words without signal words. To check this further, we reordered the phrases in the second example and added a signal. The modified sentence we tested our model on was *"Mondal was hit by one of the bombs because both sides were raining bombs on each other, Murshidabad district magistrate Pervez Ahmed Siddiqui said"*. This sample does not change the meaning of the original sentence, but is reorganised and the conjunction is changed from a joining conjunction ('and') to a causal one ('because') and the model could classify the modified sentence as a causal sentence. False positives were also observed, the third and fourth examples contained an event or action, but the cause is not explicitly mentioned in the sentences. These incorrect predictions are a result of frequently encountering similar sentence structures in causal sentences. Longer sentences, having multiple clauses were also misclassified as causal sentences even when they are missing a cause of effect for the same reason.

Errors in subtask 2 were mainly because of incorrect and inconsistent predictions of cause and effect. The number of samples containing signals are very few in the dataset and therefore not well generalized by the model. As observed in Table 4.5, either the complete sequence is not predicted, or a few tokens in the middle are incorrectly predicted.

## 4.4 Conclusion

With the rapid growth in information from news portals, automated solutions to analyze data and draw inferences from the data play a pivotal role. Our solution for both subtasks involved adding an adapter layer which improves the performance by avoiding full fine-tuning of the entire model and instead adding additional newly initialized weights at every layer of the transformer trained during fine-tuning. Though the solutions work well, they could be further improved by using an ensemble model for subtask 1 and by adding an LSTM [34] and CRF [91, 36, 35] on top of the contextual embeddings layer for proper alignment of tokens and labels for subtask 2.

In our experiments on the causal news corpus and on analyzing the misclassified samples we feel that the models for both subtasks can also benefit from having extra syntactic and semantic information. For subtask 1, verbs and signal arguments like conjunctions play a significant role in determining if the sentence is causal or non-causal. Similarly for subtask 2, having part-of-speech tags information for all the tokens along with contextual embedding from BERT might work well. The current models have good contextual representations, but appending them with an extra embedding of the main verbs, conjunctions, and parts-of-speech tags might steer the task inference in a better direction.

## *Chapter 5*

# Conclusion and Future Work

Document processing, comprehension, analysis, and key information extraction play a critical role in modern information management systems. With the exponentially growing volume of data and advancements in technology, the importance of these tasks has become increasingly evident. Information extraction and causality detection and identification are well-studied problems, with vast amounts of literature and research devoted to their study.

In the thesis, we address two problems. The first is extracting information from contract documents. Specifically, we aim to automatically extract or highlight significant components of contracts for user review. Due to the high skill and time required to review each line of a contract manually, the automatic highlighting of crucial information is essential for simplifying the end user's comprehension and for enabling effective decision-making. However, the lack of publicly available datasets containing contracts and their key information, including user-specific and novel information, creates a major obstacle in designing a solution using current state-of-the-art deep learning techniques. Despite this challenge, we demonstrate the practical implications of these problems by addressing them in a highly real-world application-driven, domain-specific environment.

In the second problem, we look at causality detection and identification in news reports. Causality detection is an important tool for text analysis, allowing for a better understanding of the relationships between events and actions in a text. It enables more accurate predictions and better decision-making. NLP and machine learning techniques can be used to improve the accuracy of sentiment analysis, event prediction, and other applications that require an understanding of the relationships between different entities in a text. Causality detection and identification find their application in various domains and require the time and effort of experts in that field. This problem becomes challenging as it involves identifying the semantic and cognitive connections in a sentence. We present an approach to automatically identify the cause-effect relation in texts by exploiting transformer variants with an effective fusion strategy.

## 5.1 Future Work

There are various avenues for future research. One direction is to focus on extracting crucial components by emphasizing the extraction of key information. Our study concentrated on the user or contract reviewer's viewpoint, but this assumption can be loosened, and the problem can be considered in a more generic context. Additionally, when it comes to using the information provided in contracts, there are other categories of entities involved, such as the legal team or internal stakeholders within the company. These parties may have different objectives than those assumed in this thesis. Similarly, the analysis of contracts can also be extended from identifying the novel and key elements to identifying the obligations and the consequences of the contract.

Likewise, the detection of causality in news reports could be expanded to encompass enterprise financial and sales reports. This analysis could prove advantageous to the company in managing its customers more effectively and monitoring the impact of its policies and modifications. Additionally, other relationships between entities, such as purpose and temporal relation, could be extracted from the reports to reinforce data-driven analysis.

# Related Publications

## Relevant Publications

1. **Adibhatla, Hiranmai**, and Manish Shrivastava. ”SConE: Contextual Relevance based Significant CompoNent Extraction from Contracts.” Proceedings of the 19th International Conference on Natural Language Processing (ICON). 2022.161-171.

2. **Adibhatla, Hiranmai Sri**, and Manish Shrivastava. ”LTRC@ Causal News Corpus 2022: Extracting and identifying causal elements using adapters.” Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE). 2022.50-55

## Other Publications

Vaidya, Shalaka, **Hiranmai Sri Adibhatla**, and Radhika Mamidi. ”Samajh-Boojh: a reading comprehension system in Hindi.” Proceedings of the 16th International Conference on Natural Language Processing. 2019.239-248.

# Bibliography

[1] V. Aggarwal, A. Garimella, B. V. Srinivasan, A. N, and R. Jain. ClauseRec: A clause recommendation framework for AI-aided contract authoring. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8770–8776, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[2] K. Al-Kofahi, A. Tyrrell, A. Vachher, and P. Jackson. A machine learning approach to prior case retrieval. In *Proceedings of the 8th international conference on Artificial intelligence and law*, pages 88–93, 2001.

[3] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998.

[4] B. Barik, E. Marsi, and P. Øzturk. Event causality extraction from natural science literature. 2016.

[5] J. Benesty, J. Chen, Y. Huang, and I. Cohen. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[6] S. Bethard and J. H. Martin. Learning semantic links from a corpus of parallel temporal and causal relations. In *Proceedings of ACL-08: HLT, Short Papers*, pages 177–180, 2008.

[7] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh. A comparative study of summarization algorithms applied to legal case judgments. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41*, pages 413–428. Springer, 2019.

[8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[9] P. Cao, X. Zuo, Y. Chen, K. Liu, J. Zhao, Y. Chen, and W. Peng. Knowledge-enriched event causality identification via latent structure induction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4862–4872, 2021.

[10] I. Chalkidis and I. Androutsopoulos. A deep learning approach to contract element extraction. In *JURIX*, pages 155–164, 2017.

[11] I. Chalkidis, I. Androutsopoulos, and N. Aletras. Neural legal judgment prediction in english. *arXiv preprint arXiv:1906.02059*, 2019.

[12] I. Chalkidis, I. Androutsopoulos, and A. Michos. Extracting contract elements. In *Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law*, pages 19–28, 2017.

[13] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, Nov. 2020. Association for Computational Linguistics.

[14] Q. Chen, Z. Zhuo, and W. Wang. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*, 2019.

[15] Y. Chen. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo, 2015.

[16] D. Dalal, M. Arcan, and P. Buitelaar. Enhancing multiple-choice question answering with causal knowledge. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 70–80, Online, June 2021. Association for Computational Linguistics.

[17] P. Delobelle, T. Winters, and B. Berendt. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online, Nov. 2020. Association for Computational Linguistics.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, June 2019. Association for Computational Linguistics.

[19] Q. Do, Y. S. Chan, and D. Roth. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 294–303, 2011.

[20] R. J. Doyle. Hypothesizing and refining causal models. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, 1984.

[21] J. Dunietz, G. Burnham, A. Bharadwaj, O. Rambow, J. Chu-Carroll, and D. Ferrucci. To test machine comprehension, start by defining comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7839–7859, Online, July 2020. Association for Computational Linguistics.

[22] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, and W. Wang. 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, May 2022. Association for Computational Linguistics.

[23] J. R. Finkel, T. Grenager, and C. D. Manning. 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, 2005.

[24] R. Funaki, Y. Nagata, K. Suenaga, and S. Mori. A contract corpus for recognizing rights and obligations. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2045–2053, Marseille, France, May 2020. European Language Resources Association.

[25] L. Gao, P. K. Choubey, and R. Huang. Modeling document-level causal structures for event causal relation identification. 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 1808–1817, 2019.

[26] R. Girju. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, page 76–83, USA, 2003. Association for Computational Linguistics.

[27] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610, 2005.

[28] C. Hashimoto, K. Torisawa, J. Kloetzer, M. Sano, I. Varga, J.-H. Oh, and Y. Kidawara. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 987–997, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[29] O. Hassanzadeh, D. Bhattacharjya, M. Feblowitz, K. Srinivas, M. Perrone, S. Sohrabi, and M. Katz. Answering binary causal questions through large-scale text mining: An evaluation using cause-effect pairs from human experts. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5003–5009. International Joint Conferences on Artificial Intelligence Organization, 7 2019.

[30] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J. Low, L. Bing, and L. Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online, Aug. 2021. Association for Computational Linguistics.

[31] M. A. Hearst. Automated discovery of wordnet relations. *WordNet: an electronic lexical database*, 2, 1998.

[32] C. Hidey and K. McKeown. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany, Aug. 2016. Association for Computational Linguistics.

[33] J. R. Hobbs. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22(2):181–209, 2005.

[34] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[35] Z. Huang and W. Xu. Kai yu. *Bidirectional LSTM-CRF models for sequence tagging. CoRR, abs/1508.01991*, 2015.

[36] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[37] A. Hürriyetoğlu, Ali, H. Tanev, V. Zavarella, J. Piskorski, R. Yeniterzi, and E. Yörük. Challenges and applications of automated extraction of socio-political events from text (case 2021): Workshop and shared task report. *arXiv preprint arXiv:2108.07865*, 2021.

[38] A. Hürriyetoğlu, Ali, V. Zavarella, H. Tanev, E. Yörük, A. Safaya, and O. Mutlu. Automated extraction of socio-political events from news (aespen): Workshop and shared task report. *arXiv preprint arXiv:2005.06070*, 2020.

[39] M. Ionescu, A.-M. Avram, G.-A. Dima, D.-C. Cercel, and M. Dascalu. UPB at FinCausal-2020, tasks 1 & 2: Causality analysis in financial documents using pretrained language models. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*, pages 55–59, Barcelona, Spain (Online), Dec. 2020. COLING.

[40] K. Izumi, H. Sano, and H. Sakaji. Economic causal-chain search and economic indicator prediction using textual data. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 19–25, Lancaster, United Kingdom, 15-16 Sept. 2021. Association for Computational Linguistics.

[41] P. Jackson, K. Al-Kofahi, A. Tyrrell, and A. Vachher. Information extraction from case law and retrieval of prior cases. *Artificial Intelligence*, 150(1-2):239–290, 2003.

[42] S. Jain. Question answering over knowledge base using factual memory networks. In *Proceedings of the NAACL Student Research Workshop*, pages 109–115, San Diego, California, June 2016. Association for Computational Linguistics.

[43] P. Kalamkar, A. Agarwal, A. Tiwari, S. Gupta, S. Karn, and V. Raghavan. Named entity recognition in indian court judgments. *arXiv preprint arXiv:2211.03442*, 2022.

[44] R. M. Kaplan and G. Berry-Rogghe. Knowledge-based acquisition of causal relationships in text. *Knowledge Acquisition*, 3(3):317–337, 1991.

[45] C. S.-G. Khoo. *Automatic identification of causal relations in text and their use for improving precision in information retrieval*. Syracuse University, 1995.

[46] Y. Koreeda and C. D. Manning. Contractnli: A dataset for document-level natural language inference for contracts. *arXiv preprint arXiv:2110.01799*, 2021.

[47] N. Lagos, F. Segond, S. Castellani, and J. O'Neill. Event extraction for legal case building and reasoning. In *Intelligent Information Processing V: 6th IFIP TC 12 International Conference, IIP 2010, Manchester, UK, October 13-16, 2010. Proceedings 6*, pages 92–101. Springer, 2010.

[48] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[49] J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70, 2020.

[50] Y. Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[51] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[52] X. Ma and E. Hovy. 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, Aug. 2016. Association for Computational Linguistics.

[53] V. Malik, R. Sanjay, S. K. Nigam, K. Ghosh, S. K. Guha, A. Bhattacharya, and A. Modi. Ildc for cjpe: Indian legal documents corpus for court judgment prediction and explanation. *arXiv preprint arXiv:2105.13562*, 2021.

[54] D. Mariko, H. Abi Akl, K. Trottier, and M. El-Haj. The financial causality extraction shared task (fincausal 2022). In *Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022*, pages 105–107, 2022.

[55] D. Mariko, H. A. Akl, E. Labidurie, S. Durfort, H. De Mazancourt, and M. El-Haj. Financial document causality detection shared task (fincausal 2020). *arXiv preprint arXiv:2012.02505*, 2020.

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

[57] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[58] S. Mittal, K. P. Joshi, C. Pearce, and A. Joshi. 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, 2015.

[59] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[60] R. Nallapati, F. Zhai, and B. Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[61] J. Nivre, M.-C. De Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. McDonald, S. Petrov, S. Pyysalo, N. Silveira, et al. Universal dependencies v1: A multilingual treebank

collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, 2016.

[62] K. O'Shea and R. Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

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

[64] J. Pfeiffer, A. Rücklé, C. Poth, A. Kamath, I. Vulić, S. Ruder, K. Cho, and I. Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

[65] K. Radinsky, S. Davidovich, and S. Markovitch. Learning causality for news events prediction. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 909–918, New York, NY, USA, 2012. Association for Computing Machinery.

[66] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer, 1999.

[67] P. Ren, F. Wei, Z. Chen, J. Ma, and M. Zhou. 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, Dec. 2016. The COLING 2016 Organizing Committee.

[68] A. Rücklé, G. Geigle, M. Glockner, T. Beck, J. Pfeiffer, N. Reimers, and I. Gurevych. Adapterdrop: On the efficiency of adapters in transformers. *arXiv preprint arXiv:2010.11918*, 2020.

[69] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pages 15–18, 2019.

[70] H. Schmid. Part-of-speech tagging with neural networks. *arXiv preprint cmp-lg/9410018*, 1994.

[71] K. K. Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.

[72] P. Silva, C. Gonçalves, C. Godinho, N. Antunes, and M. Curado. 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, 2020.

[73] A. Sorgente, G. Vettigli, and F. Mele. Automatic extraction of cause-effect relations in natural language text. *DART@ AI* IA*, 2013:37–48, 2013.

[74] L. Talmy. *Toward a cognitive semantics*, volume 2. MIT press, 2000.

[75] F. A. Tan, D. Hazarika, S.-K. Ng, S. Poria, and R. Zimmermann. Causal augmentation for causal sentence classification. In *Proceedings of the First Workshop on Causal Inference and NLP*, pages 1–20, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[76] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, T. Caselli, O. Uca, F. F. Liza, and N. Oostdijk. Event causality identification with causal news corpus–shared task 3, case 2022. *arXiv preprint arXiv:2211.12154*, 2022.

[77] F. A. Tan, H. Hettiarachchi, A. Hürriyetoğlu, T. Caselli, O. Uca, F. F. Liza, and N. Oostdijk. Event causality identification with causal news corpus - shared task 3, CASE 2022. In *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2022)*, Online, Dec. 2022. Association for Computational Linguistics.

[78] F. A. Tan, A. Hürriyetoğlu, T. Caselli, N. Oostdijk, T. Nomoto, H. Hettiarachchi, I. Ameer, O. Uca, F. F. Liza, and T. Hu. The causal news corpus: Annotating causal relations in event sentences from news. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2298–2310, Marseille, France, June 2022. European Language Resources Association.

[79] F. A. Tan, X. Zuo, and S.-K. Ng. Unicausal: Unified benchmark and model for causal text mining. *arXiv preprint arXiv:2208.09163*, 2022.

[80] F. A. Tan, X. Zuo, and S.-K. Ng. Unicausal: Unified benchmark and model for causal text mining, 2022.

[81] E. F. Tjong Kim Sang. 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)*, 2002.

[82] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[83] A. Voutilainen. *Part-of-speech tagging*, volume 219. The Oxford handbook of computational linguistics, 2003.

[84] P. A. White. Ideas about causation in philosophy and psychology. *Psychological bulletin*, 108(1):3, 1990.

[85] P. Wolff. Representing causation. *Journal of experimental psychology: General*, 136(1):82, 2007.

[86] P. Wolff and G. Song. Models of causation and the semantics of causal verbs. *Cognitive psychology*, 47(3):276–332, 2003.

[87] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, et al. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*, 2018.

[88] P. Xu and R. Sarikaya. 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, 2013.

[89] D. Yang, C. Leber, L. Tari, A. Chandramouli, A. Crapo, R. Messmer, and S. Gustafson. 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, 2013.

[90] W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*, 2019.

[91] N. Ye, W. Lee, H. Chieu, and D. Wu. Conditional random fields with high-order features for sequence labeling. *Advances in neural information processing systems*, 22, 2009.

[92] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR, 2020.

[93] R. Zhang, W. Yang, L. Lin, Z. Tu, Y. Xie, Z. Fu, Y. Xie, L. Tan, K. Xiong, and J. Lin. Rapid adaptation of bert for information extraction on domain-specific business documents. *arXiv preprint arXiv:2002.01861*, 2020.

[94] Z. Zhao and Y. Wu. Attention-based convolutional neural networks for sentence classification. In *Interspeech*, volume 8, pages 705–709, 2016.

[95] W. Zheng, P. Zhao, K. Huang, and G. Chen. Understanding the property of long term memory for the lstm with attention mechanism. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2708–2717, 2021.

[96] M. Zopf, E. Loza Mencía, and J. Fürnkranz. 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, June 2018. Association for Computational Linguistics.