Monsoon 2024 CS7.501

Advanced Natural Language Processing

Project Report on

Contract NLI

By Team

Semantic Scode

Shubham Kathiriya (2023201050)

Srivatsa Sinha (2024701003)

Ishita Bansal (2022114004)

❖ Abstract

Contract review is a crucial but labor-intensive and costly process for businesses, often requiring extensive legal expertise to interpret complex obligations and restrictions within documents. This work introduces *ContractNLI*, a novel dataset and task formulation for document-level Natural Language Inference (NLI) tailored to automate and streamline contract analysis. The *ContractNLI* dataset consists of 607 annotated Non-Disclosure Agreements (NDAs), each evaluated against 17 standardized hypotheses covering common contractual obligations, such as confidentiality, data-sharing limitations, and termination clauses. For each hypothesis, entailment labels (entailment, contradiction, or neutral) and corresponding evidence spans are provided. To address the challenges of this task, we present *Span NLI BERT*, a BERT-based model optimized for evidence span detection and entailment classification. Span NLI BERT frames evidence identification as multi-label classification over predefined spans and employs dynamic context segmentation to process lengthy documents effectively. Experimental results show that Span NLI BERT outperforms baseline models in evidence identification and NLI accuracy, though contradiction detection remains challenging due to label imbalance. Additionally, we explore a decoder-only model, *Phi3*, which uses instruction fine-tuning with model quantization. Phi3 demonstrates superior performance to Span NLI BERT, offering an efficient solution with reduced computational overhead.

❖ Motive of the Problem

The *ContractNLI* paper addresses a critical and specialized need in legal contract analysis: automating the review of legal documents, specifically contracts, by leveraging Natural Language Inference (NLI) techniques. Reviewing contracts manually is time-consuming, costly, and susceptible to human error, especially for lengthy or complex contracts. The **primary problem** this paper aims to solve is making contract review faster, more accessible, and less resource-intensive through automated understanding of contract obligations and terms. Here's a breakdown of the underlying motivation and the specific challenges the paper seeks to address:

1. High Cost and Inefficiency of Manual Contract Review

Contracts, such as Non-Disclosure Agreements (NDAs) or service agreements, govern a large percentage of business interactions. A study cited in the paper estimates that a typical Fortune 1000 company maintains 20,000 to 40,000 active contracts at any time. Reviewing each contract thoroughly can cost companies millions in legal fees and resources annually. Additionally, companies with limited budgets may lack access to legal professionals and might forgo comprehensive contract reviews, potentially leaving them vulnerable to unfavorable terms or legal liabilities.

The **primary motivation** is to reduce this overhead by automating contract reviews using an NLI approach, where hypotheses relevant to contract terms can be verified against the document content to classify clauses as *entailed*, *contradicted*, or *not mentioned*. This automation can potentially:

- Lower the cost and time involved in contract review.
- Increase accessibility, allowing smaller firms or individuals to assess contracts without requiring legal expertise.

2. Challenges of Traditional Information Extraction in Legal Contexts

This tackles the limitations of traditional information extraction approaches, which can identify contract clauses but lack the ability to understand their meaning and implications in context. The paper frames contract analysis as a document-level NLI task, which introduces new challenges compared to sentence-level NLI, such as handling long contract texts, identifying relevant evidence, and dealing with the complexities of legal language.

❖ Goal

By automating this process, the paper aims to reduce the cost and time involved in contract review, as well as increase accessibility for smaller organizations that may lack legal expertise. The NLI-based approach enables a more interactive, user-driven contract analysis process that provides evidence to support the system's conclusions.

❖ Objective

Two Main Tasks:

- **Contract NLI Task**: This task involves classifying the relationship between a given hypothesis and contract content. The hypotheses can either be **entailed**, **contradicted**, or **not mentioned** in relation to the contract text. The goal is to develop a model that can understand and classify these relationships accurately.
- **Evidence Identification Task**: This task involves identifying spans of text in the contract that support or contradict a hypothesis. The model needs to locate relevant spans of evidence in the document and determine their relevance to the hypothesis.
- Benchmarking the different Models and making a compressed and more robust Large language model for contract NLI

❖ Dataset and EDA

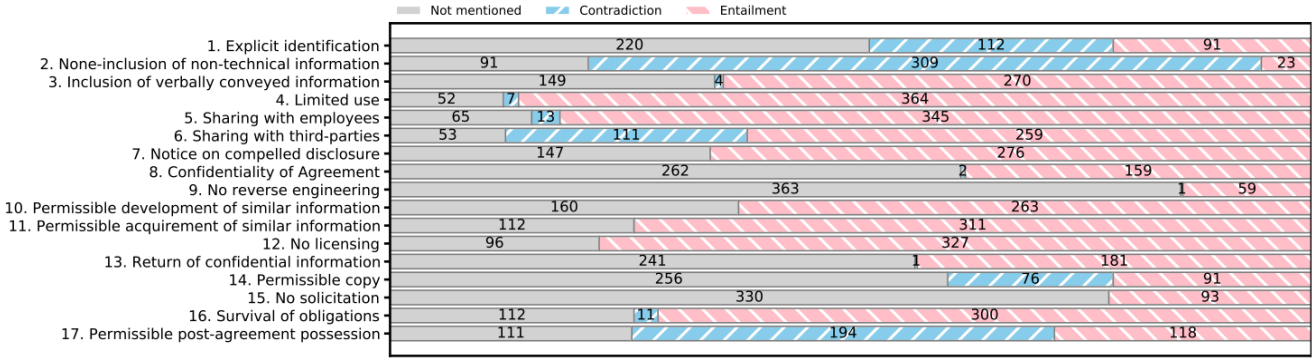


Figure (1) Dataset visualization hypothesis vise

Format	Source	Train	Development	Test	Total
Plain Text	EDGAR	83	12	24	119
HTML	EDGAR	79	11	23	113
PDF	Search engines	261	38	76	375
Total		423	61	123	607

Table 1: Data split

	Number per a document			Tokens per an instance		
	Average	Min.	Max.	Average	Min.	Max.
Paragraph	43.7	9	248	52.8	1	1209
Span	77.8	18	354	29.5	1	289
Token	2,254.3	336	11,503	—	—	—

Table 2: Basic statistics of the training dataset

The ContractNLI dataset consists of 607 NDA documents, which are split into training, development, and testing sets at a 70:10:20 ratio, stratified by document format (Table 1). As shown in Table 2, the documents have an average length of 2,254 tokens, which exceeds the 512-token maximum context length of BERT. In fact, 86% of the documents exceed this limit, highlighting the need for techniques to handle long-form legal text.

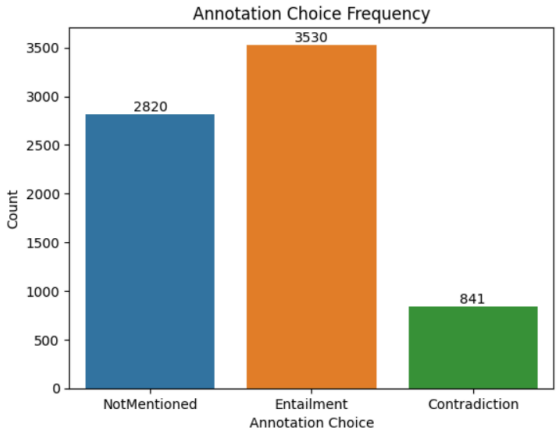
The distribution of NLI labels in the dataset is provided in Figure 1. ENTAILMENT and NOT MENTIONED account for a significant portion of the annotations, but interestingly, around half of the hypotheses contain both ENTAILMENT and CONTRADICTION labels. This suggests the dataset captures nuanced relationships between contract terms and the associated hypotheses.

Further analysis of the evidence span annotations reveals that most entailed or contradicted hypotheses have one or two supporting spans, though some go up to nine spans (Figure 3). This variability in the quantity of evidence required underscores the challenges in identifying the most salient information to support NLI predictions on legal text.

❖ EDA - Exploratory data analysis

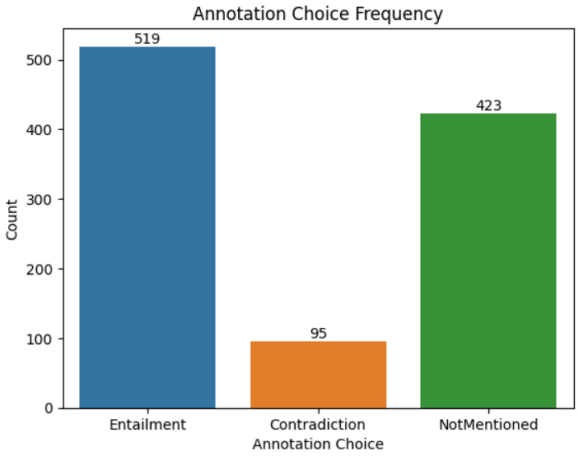
- ➢ There are a total of 17 hypotheses in the train, dev, and test split.
- ➢ There is not a single entry with a NULL value in all three splits so the data is clean.
- ➢ Most of the contract is length 5000 to 15000 (in character). The average contract length is 11,050 (in character) in train data
- ➢ Also for dev and test data, the average contract length is around 11,200 (in character).
- ➢ Most of the span length is less than 400. Same in dev and test data.
- ➢ The annotation count per contract is 17 for all contracts in train, dev, and test.
- ➢ There are only 3 types of contracts:- 'search-pdf', 'sec-text', and 'sec-html'.
- ➢ ‘Search-pdf’ is the most frequent

➢ **Train Dataset Annotation distribution**



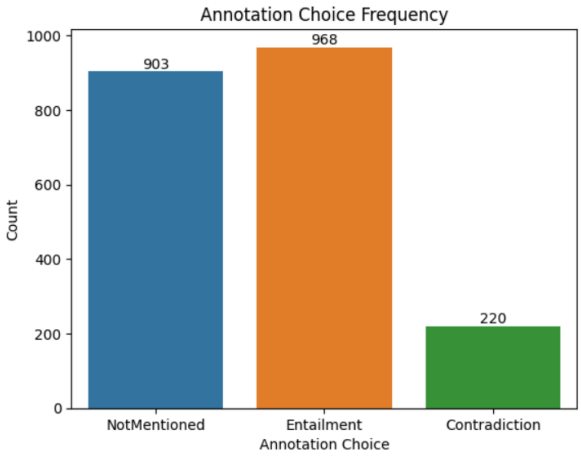
- The train data is imbalanced because the ratio of classes Entailment: Contraction: Not Mentioned is almost 4:1:3.

➢ **Dev Dataset Annotation distribution**



- The dev data is imbalanced because the ratio of classes Entailment: Contraction: Not Mentioned is almost 5:1:4.

➢ **Test Dataset Annotation distribution**



- The test data is imbalanced because the ratio of classes Entailment: Contraction: Not Mentioned is almost 4:1:4.

❖ Methodology

Part I - Approach Based on Suggestions from Paper

Baselines:-

1. Majority Votes

➔ In this approach, we use a majority voting strategy to infer the correct class labels based on annotations within each contract. Majority voting is a simple ensemble method that assigns a final label based on the most common label across multiple annotations.

➔ In many contract analysis scenarios, multiple annotations or predictions for each hypothesis might exist. By using majority voting, we aggregate these labels to choose the one with the highest frequency, assuming that this frequency represents the most likely correct label. This approach can enhance robustness by mitigating individual annotation errors and aligning with the majority opinion.

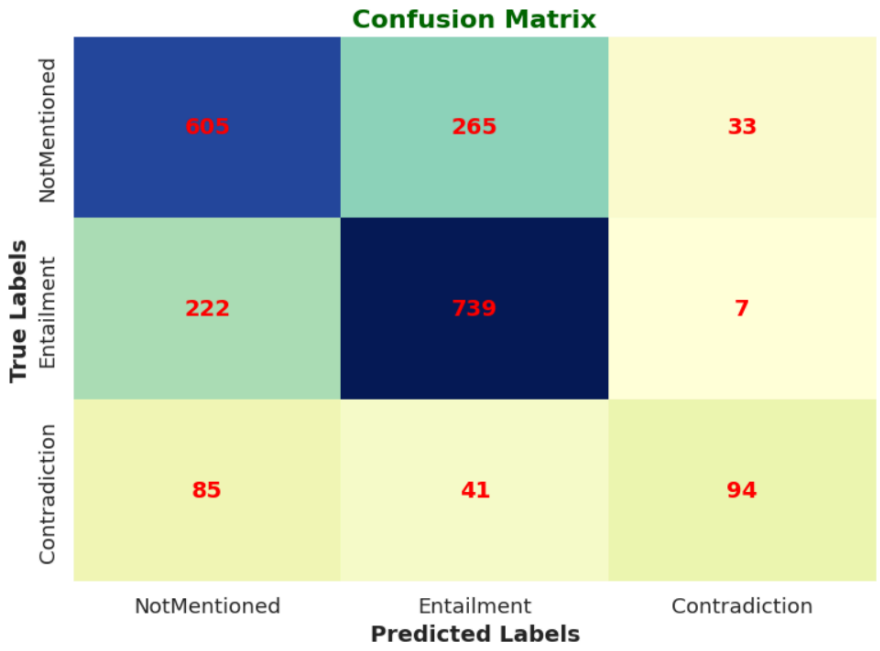
- **Majority Voting Mechanism:** For each hypothesis, maintain a count of labels (Entailment, Contradiction, Not Mentioned) and select the label with the highest count.

➔ Results

- Accuracy: 68.77 %
- Classification Report

	precision	recall	f1-score	support
0	0.66	0.67	0.67	903
1	0.71	0.76	0.73	968
2	0.70	0.43	0.53	220
accuracy			0.69	2091
macro avg	0.69	0.62	0.64	2091
weighted avg	0.69	0.69	0.68	2091

- Confusion Matrix



➔ **Drawback :**

a. Bias Toward Majority Class :

- In imbalanced datasets, where some classes dominate others, majority voting tends to favor the majority class.This can lead to poor performance on the minority class, as the ensemble model will predict the majority class more often, even if the minority class is of more interest or importance.

b. No Consideration for Class Probabilities

- Majority voting does not consider the probabilities of each class predicted by the models. It simply looks at the final predicted class, ignoring the model's confidence in that class. In many cases, this could reduce the performance of the ensemble compared to methods that weigh votes based on confidence or probability

2. TF-IDF + SVM

- The primary intuition behind this method is to represent contract text and hypotheses using **TF-IDF (Term Frequency-Inverse Document Frequency)** vectors, a common approach for converting textual data into numerical representations that capture the importance of each term. Then, a **Support Vector Machine (SVM)** with a linear kernel is employed to classify these pairs as per the entailment task. This combination leverages the TF-IDF's capability to highlight significant terms in each document and the SVM's robustness in finding an optimal hyperplane for separating classes in high-dimensional space.

★ Procedure

➔ **Text Cleaning**: To ensure consistency, the text is preprocessed by:

- Replacing newline characters and other escape sequences with spaces.
- Removing repeated characters to simplify word variations (e.g., "noooo" to "no").
- Tokenizing and lowercasing text.

➔ **Vectorization**:

- For each contract clause (**premise**) and each associated **hypothesis**, TF-IDF is applied separately.
- The resulting vectors for the premise and hypothesis are concatenated, forming a single feature vector that represents their relationship.

➔ **SVM Initialization**: A linear kernel SVM model is chosen due to its efficiency in handling high-dimensional sparse vectors from TF-IDF. SVMs are well-suited for text classification as they maximize the margin between classes.

★ Results:

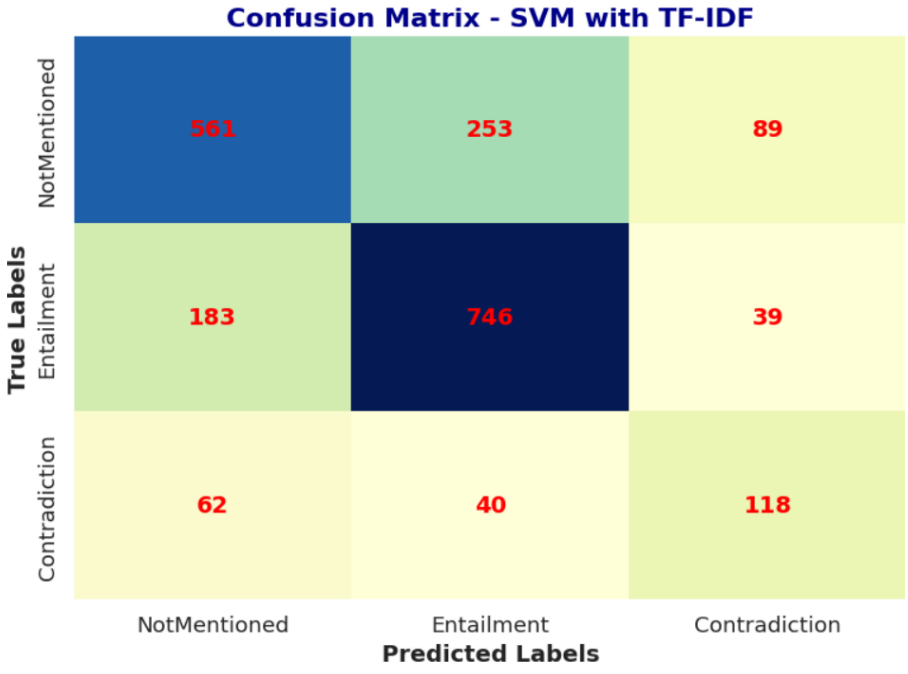
- Accuracy : 68.15 %
- Classification Report

```
Classification Report:
                precision    recall  f1-score   support

 NotMentioned       0.70      0.62      0.66       903
   Entailment       0.72      0.77      0.74       968
Contradiction       0.48      0.54      0.51       220

     accuracy                           0.68      2091
    macro avg       0.63      0.64      0.64      2091
 weighted avg       0.68      0.68      0.68      2091
```

- Confusion Matrix



★ Limitations :

a. Lack of Semantic Understanding

- TF-IDF only captures the frequency of words, not their meaning or relationships. This means that synonyms, paraphrases, or words with similar meanings are treated as entirely separate features, which limits the model's understanding of text semantics.

b. Sparse and High-Dimensional Feature Space

- The TF-IDF representation creates a very high-dimensional, sparse matrix, with each unique word (or n-gram) in the vocabulary being a separate feature. This leads to large memory and computational requirements, especially for datasets with extensive vocabularies or long documents.

c. Inability to Handle Long-Distance Dependencies

- TF-IDF treats words as independent of each other, ignoring the order and structure of words in sentences. This can be problematic for texts where word order or long-distance dependencies are important for understanding context.

3. Span TF-IDF + Cosine Similarity

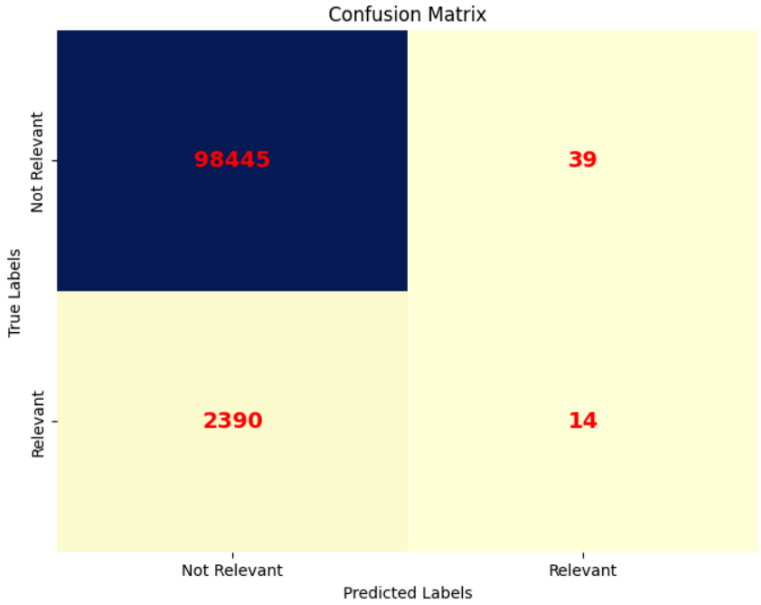
➔ This method is a baseline approach focused specifically on **evidence identification** in contract NLI tasks. Here's a breakdown of its workings, advantages, limitations, and performance.

★ Procedure

1. **TF-IDF Vectorization**:
    - For each contract document, this approach represents both hypotheses and spans as **unigram-level TF-IDF vectors**.
    - TF-IDF (Term Frequency-Inverse Document Frequency) captures the importance of each word in a span relative to its frequency across the document. Words that appear frequently in a span but rarely in other documents are given higher weights.
2. **Cosine Similarity**:
    - After vectorizing the hypotheses and spans, **cosine similarity** is used to measure the lexical similarity between each hypothesis and every span within the document.
    - Cosine similarity evaluates the angle between the hypothesis and span vectors; the closer the angle to zero (i.e., the closer the vectors), the higher the similarity score.
    - By comparing each span's cosine similarity score with the hypothesis, the model identifies the span with the **highest similarity** as evidence for supporting or refuting the hypothesis.
3. **Span Selection**:
    - Only the span with the highest cosine similarity score is selected as evidence, meaning the model picks the span most lexically similar to the hypothesis.
    - This span is considered the primary evidence supporting or contradicting the hypothesis.

★ Results

- Precision @ 80% Recall: 0.0597
- Mean Average Precision: 0.1536



★ Limitations:

- **Limited Contextual Understanding**: This approach is purely similarity-based, relying on lexical matching without considering the broader context of the hypothesis. It lacks the nuanced, learned understanding that a classifier could provide.
- **No Handling of Paraphrasing**: The cosine similarity approach is effective for spans that closely match the hypothesis in wording, but it struggles with paraphrased or conceptually related text that doesn't share exact terms.
- **No Contract-Specific Nuance**: Contracts often use complex legal language that may not align lexically with the hypotheses. Without semantic understanding, the model might miss the true intent or context of the contract language.

State of the Art:- Span NLI BERT

- The Span NLI BERT approach is designed to handle Natural Language Inference (NLI) tasks within contract documents by identifying entailment relationships (entailment, contradiction, or not mentioned) and evidence spans. Here's a detailed breakdown of its motive, intuition, architecture, training objective, task design, token usage, and benefits.

- **Motive**: The primary motivation behind Span NLI BERT is to enhance the understanding of legal contracts by automatically determining whether specific clauses in a contract support, contradict, or remain irrelevant to a given hypothesis. Contracts are often dense with legal terminology, and manually analyzing them is time-consuming and error-prone. Span NLI BERT aims to automate this process by accurately identifying spans of text that either support or contradict specific hypotheses within the document.

- **Intuition**: The intuition is that by using BERT's contextualized embeddings and **NLI training,** the model can learn to capture nuanced relationships between hypotheses and complex contract text. BERT's self-attention mechanism enables it to consider the contextual importance of each word within a span relative to the hypothesis, helping it to identify evidence spans even in the presence of indirect language or legal terminology. Additionally, it allows the model to represent the semantics of both the hypothesis and the spans more effectively than similarity-based approaches, which are limited to surface-level lexical similarity.

- ## Training Objective

The training objective for Span NLI BERT consists of two main tasks:

1. **NLI Classification Task**:
    - This task aims to classify the relationship between the hypothesis and the contract text as Entailment, Contradiction, or Not Mentioned.
    - **Loss Function**: Cross-entropy loss is used on the `[CLS]` token output, with a label indicating the relationship between the hypothesis and document segment.
2. **Evidence Span Identification Task**:
    - This task determines which spans within the contract chunk serve as evidence for the classified relationship.
    - **Loss Function**: Binary cross-entropy loss is applied to each `[SPAN]` token's output, where a 1 indicates that the span is evidence and a 0 indicates it is not.

The overall training loss is a weighted sum of the NLI and span identification losses, allowing the model to optimize for both objectives simultaneously.

- ## Architecture

The **Span NLI BERT** model is built on a modified BERT architecture, tailored to handle the specific requirements of contract NLI and evidence span identification. Key components include:

1. **Input Setup**:
    - Each input comprises a **contract chunk** and a **hypothesis** concatenated with special tokens.
    - The structure is: `[CLS] hypothesis [SEP] contract_chunk [SEP]`.

2. **Span Markers**:
    - **[SPAN] tokens** are used to mark potential evidence spans within the contract chunk.
    - Each `[SPAN]` token is associated with text that could serve as evidence. The presence of multiple `[SPAN]` tokens allows the model to independently assess multiple spans in the same input.
3. **Two-Level Classification**:
    - **NLI Classifier**: A standard classifier using the `[CLS]` token's output to classify the relationship (Entailment, Contradiction, Not Mentioned).
    - **Span Evidence Classifier**: A binary classifier applied to the embeddings at each `[SPAN]` token position, which determines whether each span is relevant (1) or irrelevant (0) as evidence for the hypothesis.

This architecture enables the model to perform **joint classification** of both the relationship type and evidence spans, making it efficient and well-suited to the task's dual objectives.

● Benefits

1. **Dual Objective Handling**:
    - By simultaneously learning NLI classification and span evidence identification, Span NLI BERT is well-suited to complex contract NLI tasks. This dual-objective approach improves both document-level and span-level understanding.
2. **Contextualized Understanding**:
    - Leveraging BERT's transformer-based embeddings, the model captures subtle contextual nuances in contract language, which is essential for interpreting legal terms and phrases.

3. **Efficient Evidence Identification**:
    - The `[SPAN]` token structure allows efficient span-level predictions, highlighting relevant text without needing separate span annotations.
4. **Flexible to Multiple Hypotheses**:
    - The architecture is flexible enough to accommodate multiple hypotheses per contract by adjusting the tokenization scheme, allowing scalable evidence identification across multiple contract aspects.
5. **Strong Lexical and Semantic Matching**:
    - BERT's language understanding capabilities allow Span NLI BERT to go beyond simple lexical similarity (as in TF-IDF) and understand the semantic relationships within legal text, making it more robust in handling paraphrases and indirect evidence.

★ Results :

➔ mAP (Mean Average Precision): 58.4293

➔ Precision @80 Recall: 35.67567

➔ NLI Task Accuracy: 65.54621

➔ F1 (Entailment): 30.61

➔ F1 (Contradiction): 26.6324

Part II - Modern Methods Using Large Language Model

Introduction

The recent success of pre-trained Large Language Models with parameter size greater than one billion like OpenAI's GPT3, Meta's LLaMA, and Microsoft Phi 3 inspired us to explore the avenues of utilizing these LLMs for contract analysis.

Span NLI Bert had numerous challenge that we wished to address:

1. Limited Context Length: Span Bert Utilized dynamic stride to select a chunk of data to feed to the model at a given time. This restricted model's ability to reason over disjoint spans, reference to definition in certain spans, etc
2. Limited view into the model's reasoning capability. While the model gave certain output, there is no indication of the reasoning behind the same. In sensitive domains like Law and Medical, the reliability of such models can't be guaranteed and responses of these models need to be assessed by humans.

To this end, we wished to solve the above problems by utilizing a model capable of running over a wide context and providing reasoning for the same.

With this in mind, we selected Microsoft's Phi 3 model (mini, 128k, instruct). The model consists of 3 billion parameters with a context size of 128,000 tokens. The model has been fine tuned to follow instructions and follows the chat format described in the dataset analysis section. We evaluate a model's performance by

creating a prompt template to trigger the model to generate output similar to annotation in the dataset directly. We further finetune the base model with training data provided using LoRA with a recipe provided in the QLoRA method i.e. quantizing the original model to NF4. We keep the compute data type to FP16 due to hardware Limitations.

Dataset Reason Generation

To effectively train the model within hardware bandwidth we initially filter out datasets with tokens greater than 7181 tokens.

Beyond the initial analysis provided in this report, we further used GPT-4o for generating the train and dev dataset reason for the annotation by providing the contract, hypothesis, and ground truth labels (choice and spans) and prompted the model to give its thoughts on why the ground truth is correct. We also give the model the flexibility to contradict the choice provided in ground truth to assess the quality of the label.

The model identified 381 (5.6%) of the training dataset to have incorrect labels. It gave its thought and after verifying a few of these the contradiction seems reasonable.

E.g.

Document ID: 111

Hypothesis: The receiving Party shall not disclose the fact that the Agreement was agreed or negotiated.

Ground Truth: { "choice": "Entailment", "spans": [9]}

GPT's contradictory thought. Statement_id_9 mentions the Unilateral nature of the agreement where the 2nd Party is prohibited from disclosing confidential and proprietary information released by the 1st Party. This implies that confidentiality is crucial and disclosure is restricted. 2. The hypothesis states that the receiving party shall not disclose the fact that the Agreement was agreed or negotiated. \n3. However, statement_id_9 does not explicitly mention that the existence or negotiation of the Agreement itself is confidential, only the information shared under the Agreement. Therefore, the hypothesis is not directly supported by the chosen statement."

Statement 9 from Contract: - *"Unilateral – This Agreement shall be Unilateral, whereas, 1st Party shall have sole ownership of the Confidential Information with 2nd Party being prohibited from disclosing confidential and proprietary information that is to be released by the 1st Party."*

Conclusion: GPT's reasoning is strong and conclusive the label is indeed incorrect.

We use the reason generated by the model to Finetune Phi3 using the Chain of Thought (CoT) paradigm. The approach effectively tried to perform distillation of Phi3 using GPT-4o as the parent model and also develop the reasoning abilities of the model to solve the second issue mentioned above.

Training Methodology

1. Dataset Statistics:
    a. Total Training Points: 6680

    b. Total Eval Points: 900
    c. Total Test Points: 1921
2. We apply the following optimization to train our model under constrained hardware:
    a. Compute Type: FP16
    b. Backbone Model Data Type: NF4
    c. Model Parameters: 3B
    d. Training Distribution Framework: Deep Speed Stage 3
    e. Gradient (or Activation) Checkpointing Enabled: Yes
    f. Parameter Offloading Enabled: Yes (CPU)
3. Prompt Template:
    a. <|system|>[Instruction]<end>\n<|user|>[Contract,Hypothesis]<end >\n<|assistant|>{choice: “”, spans: []}. For detailed instructions provided in different prompts refer to the attached code.
4. Hyperparameter:
    a. LoRA r: 8
    b. LoRA Alpha: 16
    c. Learning Rate: 0.0002 (With Linear Decay)
    d. Effective Batch Size (Accounting for DPP and Gradient Accumulation): 64

Result:

We trained two different models one without a reasoning prompt and other with a model prompted to generate reason.

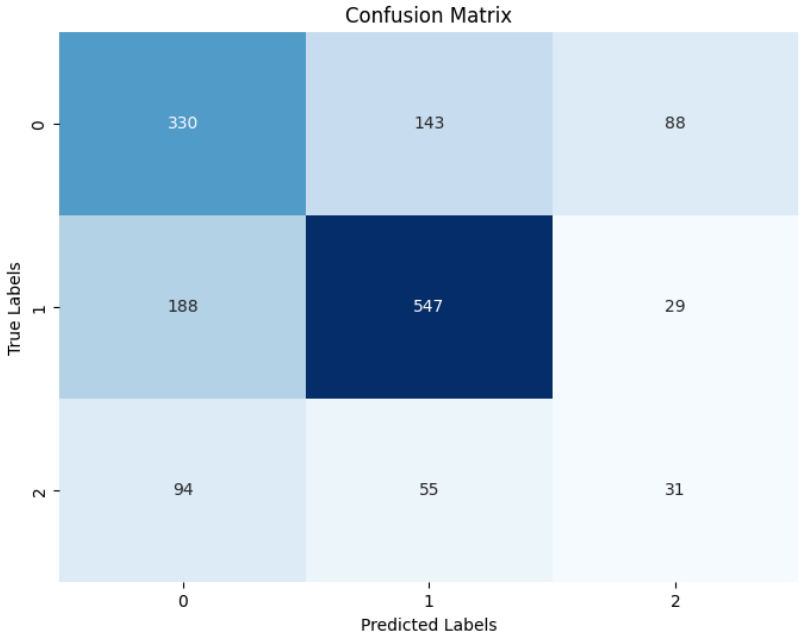
While the paper mentions mAP for Evidence prediction, it doesn't fit in our paradigm hence we selected to use IoU. We found the ratio of intersection between the span of predicted and ground truth and the union of both.

	Choice Accuracy (%)	FI(Contradiction)	F1(Entailment)	Evidence Precision	Evidence Recall
Without Fine Tuning*	47.63	0.1635	0.6748	-	-
Without Reason	86.38	0.6426	0.9015	0.3916	0.4319
With Reason	84.90	0.6529	0.8872	0.3674	0.3954

* Only NLI metric was calculated due to the model's failure to follow the structured output leading to malformed JSON output at the majority of data points

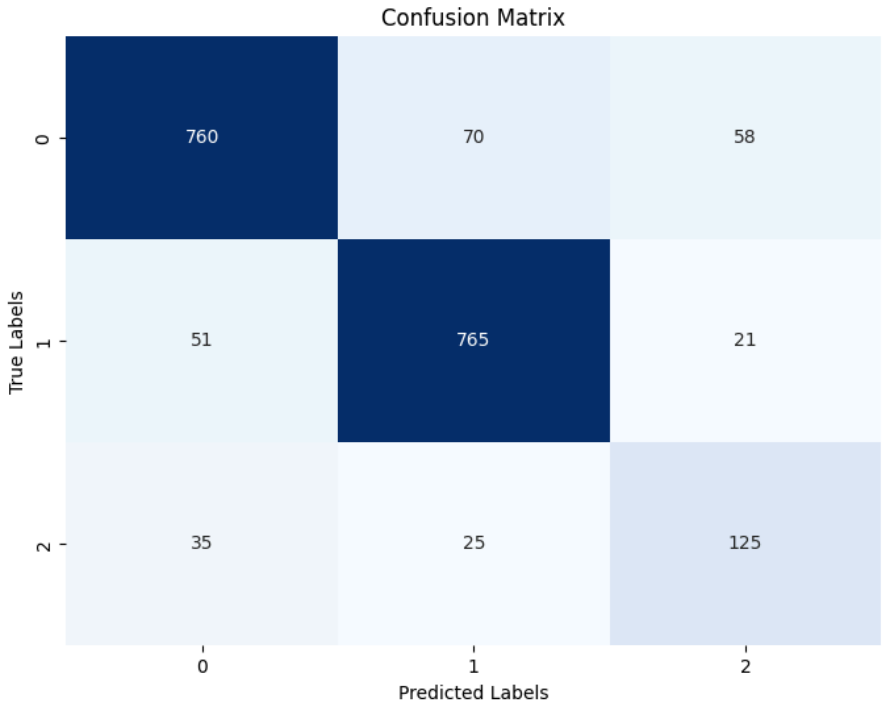
Confusion Matrix for Base Model

0: Not Mentioned, 1: Entailment, 2: Contradiction



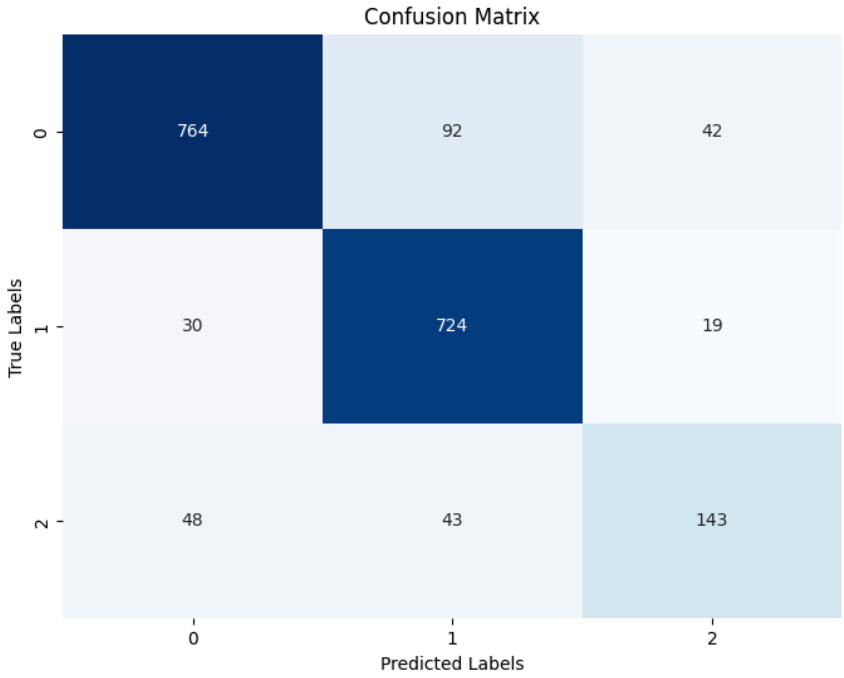
***Confusion Matrix for Model Without Reason*:**

0: Not Mentioned, 1: Entailment, 2: Contradiction



***Confusion Matrix for Model with Reason*:**

0: Not Mentioned, 1: Entailment, 2: Contradiction



Conclusion

1. We observe that the Fine Tuning model performs generally worse on all labels. We observe lower than baseline performance for NotMentioned and Contradictory while the model heavily predicts Entailment. We attribute this to hallucination.
2. We were able to achieve competitive performance by using LoRA finetuning. Although contrary to expectation the model with reason performed worse than the model without reason.
3. We observed an increase in correct prediction for Contradictory and Not Mentioned where the model was asked to reason, while a decrease in Entailment, it appears that while it's generally harder for the model to negate due to hallucination, with proper reasoning, the model was able to reduce this and provide improved prediction for negative examples.