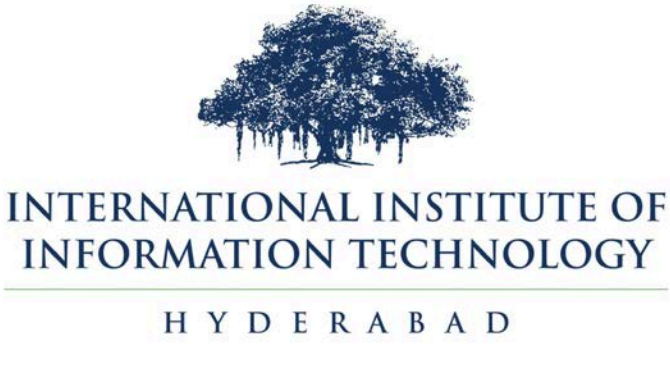


Advanced NLP

Project Report

Project Title : Contract NLI

Team Name : Badhum
Team Number : 19

Viraj Shah - 2023201011
Ronak Patel - 2023201074
Sushrut Naik - 2023201064

Guided By:

Dr. Manish Shrivastava

Teaching Assistant:

Patanjali Bhamidipati

Introduction

The project addresses the need for efficient contract review, a process which is essential to business transactions, yet time-consuming and costly. Contractual obligations and clauses often contain nuanced language, including negations and exceptions, which are challenging to interpret and automate. This project explores Document-level Natural Language Inference (NLI) to automate contract analysis, classifying hypotheses related to contract clauses as "entailed," "contradicting," or "neutral" and pinpointing supporting evidence within the contract text.

Baselines

1. Majority Vote

Task: NLI only

Description: The Majority Vote baseline is a simplistic approach where the model always predicts the majority class label for each hypothesis. Since this is a highly simplistic model, it assumes that the most frequent label in the training set will apply to all test cases.

Analysis: While this method provides a useful reference point for model performance, it tends to produce biased results in datasets where label distribution is unbalanced. This is especially evident in NLI tasks, where certain labels (like neutral or entailment) may dominate. Hence, Majority Vote is expected to perform poorly compared to other models due to its lack of sophistication.

Performance Metrics:

Label	**Precision**	**Recall**	**F1-Score**	**Support**
Contradiction	0.48	0.54	0.51	220
Entailment	0.68	0.78	0.73	968

NotMentioned	0.68	0.56	0.61	903
Accuracy			0.66	2091
Macro Avg	0.62	0.63	0.62	2091
Weighted Avg	0.66	0.66	0.66	2091

Performance Analysis:

- The model performs okay in terms of both precision and recall, especially for the "Entailment" and "NotMentioned" labels, as shown by the high F1-scores.
- However, for the "Contradiction" label, there is some class imbalance, with only one sample leading to a lower precision but perfect recall. This indicates that the majority vote method could be overfitting to prevalent labels due to the small dataset size.

2. Doc TF-IDF + SVM (Support Vector Machine)

Task: NLI only

Description: This baseline uses a linear SVM classifier to predict NLI labels based on document-level TF-IDF features. TF-IDF captures the importance of words in the document, while the linear SVM is a strong classifier for text-based tasks. Here, the contract document is represented as a bag of words using unigrams, and the model aims to predict the correct NLI label (entailment, contradiction, neutral).

Analysis: The Doc TF-IDF + SVM baseline is a classic choice for NLI tasks, and it has shown reasonable success in various domains. However, it does not capture more sophisticated semantic relationships beyond surface-level word frequencies. For contract language, which often includes complex clauses and legal jargon, this method might miss deeper contextual meanings, leading to moderate performance on NLI tasks. Compared to Majority Vote, this model is more robust but still limited by its reliance on simple bag-of-words features.

Performance Metrics:

Label	Precision	Recall	F1-Score	Support
Contradiction	0.70	0.62	0.66	903
Entailment	0.72	0.77	0.74	968
NotMentioned	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

Performance Analysis:

- The performance on a larger dataset drops, with a decrease in both accuracy (0.68) and F1-score (macro avg 0.64). This could be attributed to the complexity of the dataset, where the model struggles with "Contradiction" as seen by the low F1-score (0.51).
- The precision and recall for "Contradiction" are notably lower, showing difficulty in distinguishing contradictions from the other two classes. On the other hand, the "Entailment" class performs comparatively better with higher precision (0.72) and recall (0.77), showing the model's strength in predicting entailments.
- The weighted average F1-score (0.68) reflects the imbalanced nature of the dataset and better performance on the more frequent classes like "Entailment."

3. Span TF-IDF + Cosine Similarity

Task: Evidence identification only

Description: This baseline focuses on identifying evidence spans within contracts. It computes unigram-level TF-IDF vectors for each hypothesis and compares them to each potential span in the document using cosine similarity. The span with the highest similarity score is selected as evidence for supporting or refuting the hypothesis.

Analysis: This model is purely similarity-based and does not leverage any learned classification, relying entirely on TF-IDF for identifying spans. Cosine similarity can highlight the most lexically similar spans, but this method might fail in cases where the most relevant evidence involves paraphrasing or indirect language. While it is lightweight and easy to implement, the performance might be lower due to its lack of deep understanding of contract-specific nuances.

Performance Metrics:

Precision @ 80% recall: 0.030058717670690627

Mean Average Precision: 0.0461261085908014

4. Span TF-IDF + SVM

Task: Evidence identification only

Description: In this approach, a span-level linear SVM is used to classify whether a particular span in the contract is evidence supporting the hypothesis. Like the previous models, it relies on unigram bag-of-words features to represent spans and hypotheses.

Analysis: Span TF-IDF + SVM is an improvement over Cosine Similarity as it introduces a more discriminative classifier (SVM) to determine which spans are relevant evidence. The span-level SVM can learn patterns in the dataset that go beyond surface similarity, improving the accuracy of evidence identification. However, similar to other TF-IDF-based models, it still struggles to capture deeper semantic meanings. This method may provide better results than Span TF-IDF + Cosine, especially in distinguishing subtle differences between evidence and non-evidence spans. The performance below is lower than cosine similarity as the model was

trained on only 100 samples as opposed to the entire training set in cosine similarity. This was done due to very high training times for the svm approach.

Performance Metrics:

Precision @ 80% recall: 0.02521623982193672

Mean Average Precision: 0.02521623982193672

Dataset Analysis

We have used 607 annotated contracts, the largest dataset available.

Split into :

Train Set : 423 Contracts

Dev Set : 61 Contracts

Test Set : 123 Contracts

The basic structure in all files contains a list of documents. Each document has several attributes, such as:

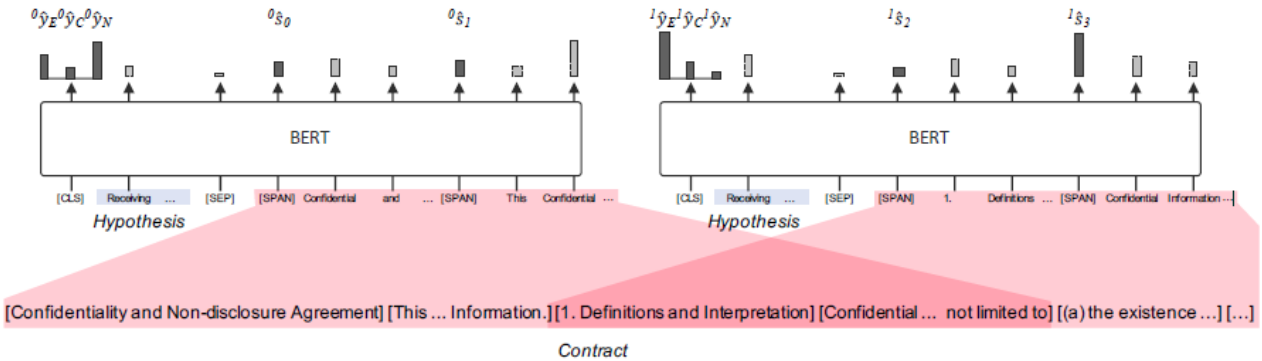
- id: Unique identifier for the document
- file_name: Name of the file (usually a legal document like NDA)
- text: Full text content of the document
- spans: Indices or ranges within the text that are marked for classification purposes
- annotation_sets: Contains the hypotheses data, including the name, the type and the span indices if the type is entailment or contradiction

	Number per Document		
	Average	**Min**	**Max**
Paragraphs	43.7	9	248
Spans	77.8	18	354
Tokens	2254.3	336	11503

Motivation

- Previous works that have predicted evidence spans have predicted the starting and ending index of the evidence using static windows and strides to split the document
- This makes the problem unnecessarily complex as the model now has to solve span boundary detection and evidence identification. This can also cause problems if the evidence is not in contiguous spans or is split across contexts
- To solve this problem, the author proposed inserting special [SPAN] tokens, each of which represents a span and modelled the problem as straightforward multi-label binary classification over the [SPAN] tokens
- Another issue was the small token limit of language models, whereas most documents are larger. As a solution to this issue, the authors propose a technique called Dynamic Context Segmentation. Here the strides are varied dynamically such that each span is fully contained in at least one context and there is enough context for understanding the span

Span NLI BERT Architecture



1. **Input Representation**

   - The input to the model consists of a **hypothesis-context** pair:
     - **Context**: Segments that the document is split into
     - **Hypothesis**: A concise statement or question derived from the contract's clauses
   - Each input segment is tokenized into a sequence using **BERT's tokenizer** and is prepended with [CLS] (classification token) and separated by [SEP] (separator token). The final input format is:

     [CLS] Hypothesis [SEP] [SPAN] Context1 [SPAN] Context2 … [SPAN] Context n

2. **Context Windowing Mechanism**

- Given the 512-token limit of BERT, a sliding window approach is used:
  - The document is split into overlapping chunks, each of up to 512 tokens (including special tokens).
  - Overlaps between chunks are used to ensure that spans crossing window boundaries are not missed.
  - For each chunk, the model makes independent predictions, which are later aggregated.

3. **Transformer Encoder (BERT)**

   - The **BERT encoder** processes each tokenized sequence, producing contextual embeddings for each token.
   - The embedding corresponding to the [CLS] token is used for **hypothesis classification**, while the embeddings for [SPAN] tokens are used for **span identification**.

4. **Span Extraction Module**
    - The embeddings output by the **BERT encoder** are utilized for two distinct tasks, employing both the [CLS] tokens and [SPAN] tokens for classification and evidence identification.
    - The architecture includes two **Multi-Layer Perceptron (MLP)** classifiers:
        - **[CLS] Token Classifier**: The embedding corresponding to the [CLS] token is used for a **multi-class classification task**. It predicts whether the hypothesis is "entailed," "not mentioned," or "contradicted" by the document context
        - **[SPAN] Token Classifier**: The span tokens are used in a **multi-label classification task** through a separate **MLP classifier**. This classifier determines whether each span serves as evidence for any of the 17 hypotheses related to the contract. For each span, it outputs probabilities indicating its relevance to the various hypotheses.
    - The **[CLS] classifier** and the **[SPAN] classifier** operate concurrently, enabling the model to jointly handle document-level classification while simultaneously identifying specific evidence spans.

5. **Multi-Task Learning Framework**
    - The model is trained using **two loss functions**:
        - **Span Identification Loss** (l_{span}): A binary cross-entropy loss for predicting whether a token belongs to a relevant span.

$$l_{span} = \sum_{i} (-s_i \log \hat{s}_i - (1 - s_i) \log(1 - \hat{s}_i))$$

        - **NLI Loss** (l_{NLI}): A standard cross-entropy loss for predicting entailment, contradiction, or neutrality.

$$l_{NLI} = \begin{cases} -\sum_{L \in \{E,C,N\}} y_L \log \hat{y}_L, & if\ s_i = 1 \\ 0, & otherwise \end{cases}$$

    - The multitask loss is

$$l = l_{span} + \lambda l_{NLI}$$

where λ is a hyperparameter that controls the balance between the two losses.

Results

The Span NLI BERT achieved the following metrics with bert-base-uncased:

- **Mean Average Precision (mAP)**: 0.5432
- **Precision at 80% Recall**: 0.1015
- **NLI Accuracy**: 0.6054
- **F1 Score for Entailment**: 0.2704
- **F1 Score for Contradiction**: 0.3024

The Span NLI BERT achieved the following metrics with bert-large-uncased:

- **Mean Average Precision (mAP)**: 0.5883,
- **Precision at 80% recall**: 0.2567,
- **NLI Accuracy**: 0.6554,
- **F1 Score for Entailment**: 0.2621,
- **F1 Score for Contradiction**: 1.0

```
[109…  {'mAP': 0.5883051888367965,
        'precision_at_80_recall': 0.25675675675675674,
        'nli_acc': 0.6554621848739496,
        'f1_score_for_entailment': 0.26212590299277605,
        'f1_score_for_contradiction': 1.0}
```

In comparison, baseline models showed significantly lower performance:

- **Span TF-IDF + SVM**: mAP of 0.0252, Precision @ 80% Recall of 0.0252
- **Span TF-IDF + Cosine Similarity**: mAP of 0.0461, Precision @ 80% Recall of 0.0301

- **Doc TF-IDF + SVM**: Accuracy of 0.68

Analysis

What Works Better and Why Span NLI BERT surpasses baseline models in all areas due to its ability to adapt context segmentation and process spans as independent units. Unlike TF-IDF and SVM-based approaches, which rely on shallow lexical representations, Span NLI BERT benefits from contextual embeddings, making it more adept at capturing nuanced contractual language.

Areas of Usefulness

Use Span NLI BERT (bert-large-uncased):

- For tasks where evidence span identification is crucial (e.g., explainable AI, contract review), Span NLI BERT with bert-large-uncased is the most effective option due to its high mAP and precision.

2. Fine-Tune for NLI Accuracy:
    - While Span NLI BERT performs well on spans, further fine-tuning may help close the accuracy gap with the Doc TF-IDF + SVM baseline.
3. Address Imbalanced Performance:
    - The F1 Score for Contradiction in the bert-large-uncased variant is perfect (1.0), but this might indicate overfitting to certain patterns.

Limitations and Areas for Improvement Despite improved performance, bert-base-uncased struggles with "contradiction" labels due to imbalanced label distributions. Discontinuous spans and negation by exceptions—where clauses alter meaning in later sections—pose challenges as the model requires significant contextual understanding. Enhancing attention mechanisms to better integrate distant contextual spans may improve performance in such cases.

References

Koreeda, Yuta, and Christopher D. Manning. "ContractNLI: A dataset for document-level natural language inference for contracts." *arXiv preprint arXiv:2110.01799* (2021).

Models

Bert-base-uncased - https://iiitaphyd-my.sharepoint.com/:u:/g/personal/ronak_patel_students_iiit_ac_in/ESfSN7g0J4ZMlT5jTk1wRNQBdfopI3zY52GH442owqdpkA?e=aDabLY

Bert-large-uncased - https://iiitaphyd-my.sharepoint.com/:u:/g/personal/ronak_patel_students_iiit_ac_in/EVHrFx5pxetIrU-_YheYl0sBhzzNLK13AZsn_PRxNlMQog?e=6CzuDD