Critical Sentence Identification in Legal Cases Using Multi-Class Classification

Sahan Jayasinghe, Lakith Rambukkanage,
Ashan Silva, Nisansa de Silva, Amal Shehan Perera

Table of Contents

Introduction

1. Importance of NLP in the Legal Domain
2. Important notions in the Legal domain
3. Problem Statement

NLP in the legal domain

- Legal domain is an area with growing amounts of textual data.
- Also these texts are read and analyzed by professionals on a daily basis.
- Interpreting legal texts is an analytically demanding task.

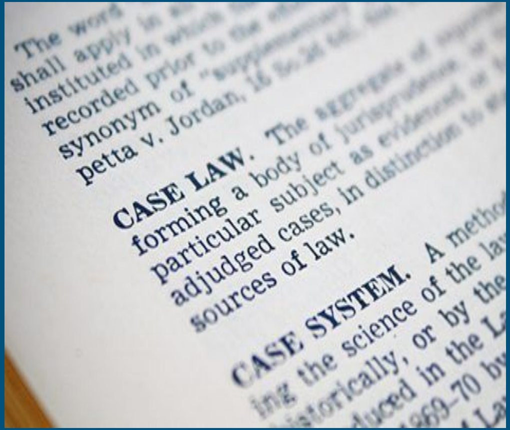
NLP in the legal domain : Research

- Party Based Sentiment Analysis
- Party Identification
- Domain Specific Embeddings
- Identifying Discourse Relations among sentences
- Critical Sentence Identification
- Winning Party Prediction

Important notions in the Legal domain

<u>Case Law</u>

- Past court cases are documented for future use.
- They are used to support ongoing court cases.
- Accounts to a large amount of text data in the legal domain.
- Legal professionals use them on a daily basis.
- Good source for training data to apply NLP



Important notions in the Legal domain

Legal Party

Two main parties :

1. Petitioner
2. Defendant

Extracting these parties is not the same as Name Entity Recognition (NER)

- **Other entities are present in the text:** Requires Party Extraction
- **Use of different names:** Requires coreference resolution

Related Work

1. Critical Sentence identification
2. Party based Sentiment Analysis Dataset

Critical Sentence identification

In the work of Glaser et al. [1], they have used German Civil code and Legal contracts. 9 semantically different classes have been used.

Researchers Jagadeesh et al. [2] have used a scoring mechanism based on features to identify the important sentences. Scoring has enabled query based ranked retrieval.

Hirao et al. [3] have explored the important sentence extraction aspect by using SVMs. They have defined the two classes as important and unimportant.

[1] I. Glaser, E. Scepankova, and F. Matthes, “Classifying semantic types of legal sentences: Portability of machine learning models,” in Legal Knowledge and Information Systems. IOS Press, 2018, pp. 61–70.
[2] Jagadeesh, P. Pingali, and V. Varma, “Sentence extraction based single document summarization,” International Institute of Information Technology, Hyderabad, India, vol. 5, 2005.
[3] T. Hirao, H. Isozaki, E. Maeda, and Y. Matsumoto, “Extracting important sentences with support vector machines,” in COLING 2002: The 19th International Conference on Computational Linguistics, 2002.

Party based Sentiment Analysis Dataset

“SigmaLaw-absa: Dataset for aspect-based sentiment analysis in legal opinion texts” [4] addresses the need for proper datasets for party based sentiment analysis (PBSA).

Nearly 2000 sentences extracted from United States Supreme Court cases have been annotated per each party in the sentence and the overall sentence sentiment.

Sentiment labels :

- positive (+1)
- neutral (0)
- negative (-1)

[4] C. R. Mudalige, D. Karunarathna, I. Rajapaksha, N. de Silva, G. Ratnayaka, A. S. Perera, and R. Pathirana, “Sigmalaw-absa: Dataset for aspect-based sentiment analysis in legal opinion texts,” arXiv preprint arXiv:2011.06326, 2020.

Methodology

1. Preparation of the Dataset
2. Multi-class Classification Model
3. Task Specific Loss Function

Preparation of the Dataset

- Used the PBSA dataset which consists of 1822 sentences from 25 US Supreme Court cases.
- Also, it consists of a sentiment label (Positive/Negative/Neutral) for each party member mentioned in a sentence, based on the sentence context.
- Extended this dataset by calculating the overall sentiment (impact) for petitioner party for each sentence.
- Also labeled each sentence with the decision of the court case where the sentence belonged (Whether the petitioner party won or lost).

Preparation of the Dataset

The following example court case sentence elaborates how the overall sentiment related to petitioner is derived.

After obtaining a warrant, the officials searched Lee's house, where they found drugs, cash, and a loaded rifle.

Defendant Party
- officials: Positive (+1)
- they: Positive (+1)

Petitioner Party
- Lee: Negative (-1)

Overall Sentiment for Petitioner party

= ((Avg. sentiment for petitioner) - (Avg. sentiment for defendant)) / 2

= ((-1) / 1 - (+2) / 2) /2

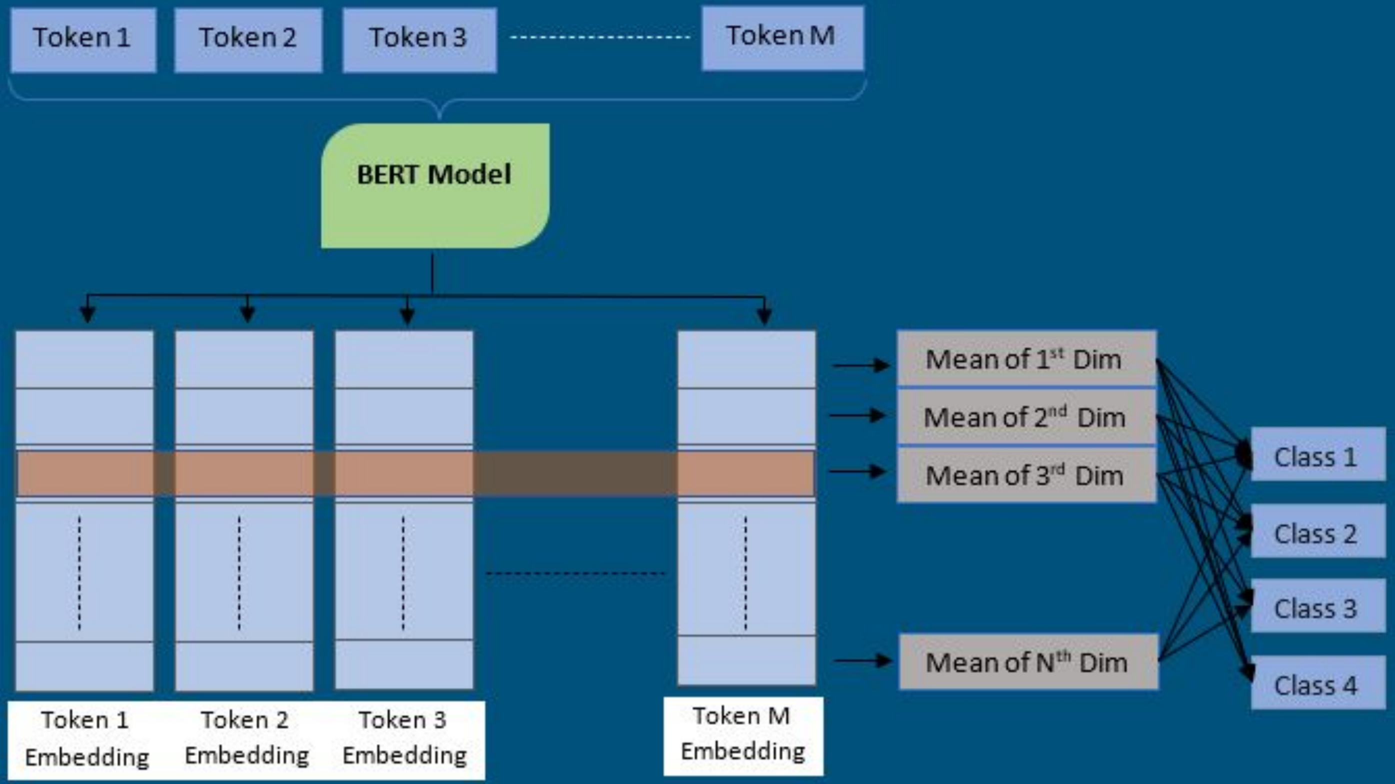
= -1

Preparation of the Dataset

Four categories of sentences within the dataset could be identified from the combination of case decision and overall sentiment towards petitioner.

Decision (with respect to Petitioner)	Overall Sentiment towards Petitioner	Sentence Count
lose	negative	226
lose	positive	230
win	negative	687
win	positive	465

Multi-class Classification Model



Task Specific Loss Function

A loss function is implemented to exploit the polarity within the four classes during the model training process.

Petitioner lose & negative	Petitioner win & negative
Petitioner lose & positive	Petitioner win & positive

Goal of the loss function is to penalize more on the probabilities of opposite classes.

Ex: When the label is ***petitioner win and positive***, the probabilities of the classes ***petitioner lose & negative*** and ***petitioner lose & positive*** should be considered more severely for the training loss.

Task Specific Loss Function

$$P_{c_i} = Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad \text{--------} \; (1)$$

x_i : *output of node i*

$P(c_i)$: *probability of class i*

$$L_{c_i} = W_{c_i} \times \log(1 - P_{c_i}) \quad \text{--------} \; (2)$$

W_{c_i} : *Weight for class i*

L_{c_i} : *Loss of class i*

$$Loss = \sum_i L_{c_i} \quad \text{--------} \; (3)$$

$$W_{c_i} = \begin{cases} 0, & for\ labeled\ class \\ 1, & for\ same\ polarity\ class \\ w, & for\ opposite\ polarity\ class \end{cases}$$

Experiments and Results

Class	Original train set	Over-sampled train set	Under-sampled train set	Validation set	Test set
Petitioner lose & negative	176	547	176	25	25
Petitioner lose & positive	180	547	176	25	25
Petitioner win & negative	547	547	176	70	70
Petitioner win & positive	365	547	176	50	50

Experiments and Results

Loss Function	Opposite class loss weight	Over-sampled		Under-sampled	
		Accuracy (%)	Macro-F1 (%)	Accuracy (%)	Macro-F1 (%)
Categorical Cross Entropy	N/A	68.82	67.86	58.24	59.99
Task specific loss function	1	72.94	70.39	63.53	64.68
	2	73.53	72.02	61.18	60.49
	3	71.76	70.97	62.94	63.68
	4	74.12	**73.62**	63.53	64.83
	5	74.12	73.26	**65.88**	**65.66**
	6	**75.29**	73.24	60.00	60.52
	7	74.71	73.57	59.41	60.90
	8	71.18	70.67	59.41	61.11

Conclusion and Future work

- So far we discussed how the identification of critical sentences in court cases can be automated.
- The Task Specific Loss Function outperforms the Categorical cross entropy loss.
- More annotated data and domain specific embeddings would improve the model.
- Future work :
    - A Sentence embedding for the legal domain.
    - Predicting the winning party of court cases.

Thank you!

Q & A