LDSI W2021 Project Report

Name: Md Siyam Sajeeb Khan (03720299)
Gloss ID: tum_ldsi_61

Summary

In this experimental NLP project, the task of sentence classification in BVA decisions was carried out on a dataset of 141 annotated decisions containing 15349 annotations in several phases. First, after experimenting with different sentence segmenters, the law-specific sentence segmenter LUIMA [2] was used to segment all the decisions of an unlabeled corpus, which were then tokenized using a Python-based NLP module Spacy. Two types of featurization, TFIDF, and word embedding were developed and after experimenting with several linear and nonlinear classifiers with these two features, a nonlinear SVM classifier with radial basis function kernel produced the highest accuracy of 0.84.

Dataset Splitting

Here are the 14 IDs of the documents (7 from each granted and denied decisions) which constitute the dev and test set, created by adhering to the project guideline instructions:
Devset ids:
1222885.txt, 1423581.txt, 0730252.txt, 1402660.txt, 0915952.txt, 1550623.txt, 1522066.txt, 0730523.txt, 0825725.txt, 0904118.txt, 0828967.txt, 0811472.txt, 0724576.txt, 0916473.txt

Testset IDs:
0918958.txt, 1003646.txt, 1226035.txt, 1309811.txt, 1420688.txt, 0628729.txt, 0918858.txt, 0941862.txt, 0901508.txt, 0920940.txt, 1242455.txt, 1600990.txt, 0930721.txt, 0724458.txt

Sentence Segmentation and Analysis

Definition of the error metrics
For the segmentation task, I matched the true start position of a segment with the start position of the predicted segments. To find a match, I first determined the closest start position to the true start position from all the generated segments and then considered a generated split a match if the **|true_start - predicted_start| <= 3**. All the matched true splits were considered as TPs, the unmatched true splits were considered as FNs and all the unmatched generated splits were considered as FPs. Based on this scheme, I calculated error metrics, precision, recall, and F1 score. I experimented with three segmenters.

Standard segmentation analysis
I achieved a precision of 0.60, recall of 0.62, and an F1 score of 0.61 with the standard segmenter of Spacy [1]. I observed the standard segmenter often over and under splits the documents. Below I present three document ids with the lowest precision values:

Document ID	**Precision**	**Recall**	**F1 Score**
61aea55e97ad59b4cfc412de	0.38	0.42	0.4
61aea55e97ad59b4cfc412f0	0.4	0.32	0.36
61aea55c97ad59b4cfc41299	0.47	0.57	0.52

After inspecting the errors in the generated segments, I observed the following patterns:

- The segmenter struggles when it encounters empty spaces.
- Fails to properly segment the CaseHeaders containing words Docket No., Date, etc., and CaseFooters

- The segmenter also fails to split the single-word and multi-word headers
- Tends to over segment when encounters periods, for example, consecutive citations were divided into separate segments, and also special words such as Vet. App., etc.
- Often misses a correctly segmented sentence due to empty spaces

Some of the examples of segmentation error are presented in the table below:

Fault type	**True segment**	**Generated segment**
Failing to segment single header	REPRESENTATION	REPRESENTATION Veteran represented by: … …
Failing to segment due to empty spaces	FINDING OF FACT	<empty_spaces> FINDING OF FACT
Over Segmentation when encountering Vet. App.	see also Chelte v. Brown, 10 **Vet. App.** 268, 271, 272 (1997) …	268, 271, 272 (1997)

Improved segmentation analysis

To tackle the issues explained above, several exceptions and special cases were added to the standard segmenter of Spacy. Some of the changes include treating single and multi-word headers as separate segments, adding special cases for Vet. App., preventing splits on empty spaces (“\n”, “\t”, “\r”), adding special conditions for recognizing CaseHeaders and CaseFooters. After all these changes the overall precision improved to 0.71, recall to 0.77, and F1 score to 0.74. Moreover, the metrics of the three documents with the lowest precision also improved significantly and are now as follows:

Document ID	**Precision**	**Recall**	**F1 Score**
61aea55e97ad59b4cfc412de	0.53	0.6	0.56
61aea55e97ad59b4cfc412f0	0.64	0.84	0.72
61aea55c97ad59b4cfc41299	0.6	0.7	0.65

Law-specific sentence segmenter [2]

The law-specific sentence segmenter, LUIMA from Savelka [2] performs better than the above two segmenters and achieves an improved precision of 0.83, recall of 0.99, and F1 score of 0.90. The performance of the three documents with the lowest precisions also improved as follows:

Document ID	**Precision**	**Recall**	**F1 Score**
61aea55e97ad59b4cfc412de	0.88	0.99	0.93
61aea55e97ad59b4cfc412f0	0.71	0.96	0.81
61aea55c97ad59b4cfc41299	0.73	0.96	0.83

Observations and error analysis:

- LUIMA splits the sentences more often and it produces the highest number of splits among the three segmenters (true splits: 12450, standard: 12450, extended: 13560, LUIMA: 14842).
- High recall due to high number of splits and fewer false negatives
- It splits the single and multi-word headers successfully into separate segments
- Handles the case of Vet. App. well

- Sometimes over segments whenever upper case letters appear and this might be influenced by the fact that most of the headers are in upper case
- Splits numbers like “1.”, “2.” into separate sentences unlike Spacy
- Fails to segment the CaseHeaders properly, also cannot segment the CaseFooters properly
- The “________” at the end of the document for signatures can not be segmented by LUIMA
- LUIMA exhibits inconsistent splitting in the case of newline(“\n”) character
- Exhibits problems while handling citations

Decision on a sentence segmenter
Comparing the performance of the above three metrics, I decided to use the LUIMA segmenter for this project since it produces the best metrics and outperformed the other three segmenters in terms of precision, recall, and f1 scores. Although it sometimes oversplits the sentences and exhibits errors in splitting, still these errors are less compared to the other segmenters. The other segmenters often fail to identify different headers and subheaders and do not treat them as separate segments, whereas LUIMA mostly recognizes them. Since LUIMA is built for law-specific sentence segmentation, it will capture the essence of the law documents better than a generalized English language segmenter. Considering all these factors, I have decided to proceed with LUIMA for the rest of the task.

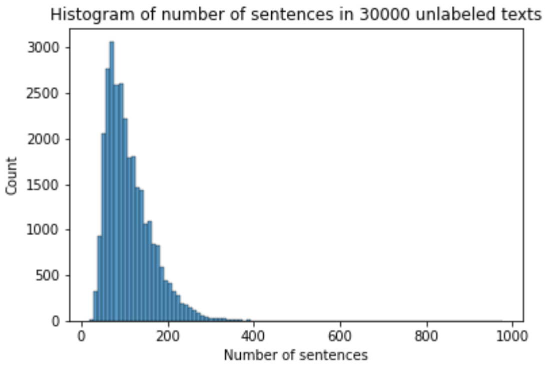
Data Preprocessing and Tokenization

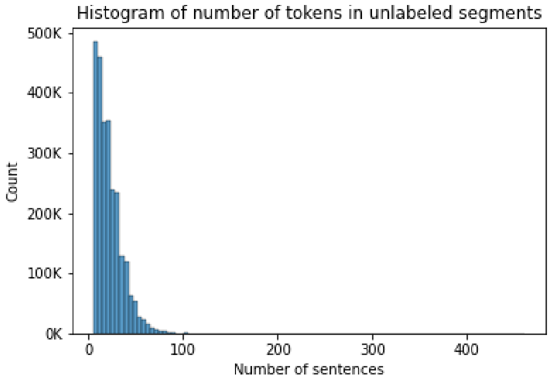
Splitting unlabeled data
The unlabeled corpus containing 30,000 documents was segmented with the LUIMA segmenter which resulted in a total number of **3,380,863** sentences.

Sentence-wise preprocessing and tokenization
Spacy’s tokenizer function was used and extended for tokenizing the sentences produced by the LUIMA segmenter. In addition to the basic functionalities of Spacy’s tokenizer, the following few preprocessing steps were applied for reducing uninformative noises from the data:

- Every token was converted to lowercase, punctuations and empty space characters (“\n”, “\r”, “\t”) were not tokenized.
- Different symbols and characters such as parentheses, �, etc. were removed and only alphanumerics were considered.
- Punctuations from the abbreviations were removed, for example, ‘U.S.C.A.’ was converted to ‘usca’.
- Numbers were simplified as <NUMd> where d is the number of digits. For instance, 3350 was represented as <NUM4>. Moreover, numbers such as (1980) were also converted to <NUM4>.
- Possessive cases such as the “Veteran’s”, “Defendant's”, “Veterans’” were preserved to capture the essence of the possessivity.
- Words with hyphens in between were broken down into separate tokens. ‘service-connection’ becomes ‘service’ and ‘connection’.
- Tokens containing both letters and numbers were removed as they bear little meaning, for instance ‘5107(b)’ after removing the parentheses becomes 5107b, which was not preserved.
- Special tokens such as ‘Vet. App.’ and ‘Fed. Cir.’, which appear together were preserved as a single token.

I tested my extended tokenizer first on the four different sentences from the workshop. Later, I also handpicked some sentences from the corpus as well as some custom-made sentences, which I deemed to be challenging, and tested the consistency of the tokenizer on them. After testing the tokenizer rigorously and being convinced of its consistency, I decided





to leverage it for tokenizing the entire unlabeled corpus. Examples of different sentences are provided in the project notebook and I am refraining from listing the produced tokens here due to the limited scope of this project report.

Tokenizing unlabeled data
The tokenizer was applied to the segmented unlabeled data and tokens of the sentences with a token count > 5 were saved into a text file. Considering this threshold of sentences containing greater than 5 tokens yielded a sentence count of **2,593,800**.

Developing Word Embeddings

Custom FastText embeddings
To develop word embeddings, I used FastText's python module. I trained an unsupervised FastText word embedding model on the tokens for producing a 100-dimensional vector model with an n-gram parameter (1,1) and a minimum word occurrence count of 20. I then trained the model for 10 epochs. I also experimented with a higher minimum word occurrence count such as 25 and also experimented by training the model for higher iterations, but it did not have any major impact on the word embeddings.

Evaluating custom embeddings manually
The model produced a vocabulary of 12217 words. To evaluate the performance of the trained model, I investigated the nearest neighbors of the provided test strings and also of some manually chosen strings. I present the nearest neighbors of a few interesting words below:

Chosen string: Veteran
Nearest neighbors: appellant, he, additionally, she, have, his, that, the, furthermore, moreover

The nearest neighbors of the string 'veteran' are contextually similar. The closest neighbor appellant' refers to a veteran. Moreover, neighbors like 'he', 'she', 'his' are also sensible since they are pronouns pointing towards the same individual. These neighbors make the most sense whereas, neighbors like 'additionally', 'have', 'furthermore', 'moreover' are the most repetitive words appearing in the legal texts along with veteran. They do not have much contextual similarity and do not make as much sense as the other neighbors described. The sentence *"Resolving all reasonable doubt in **the appellant's** favor, **the veteran's** … exposure."* also bolsters the validity of the neighbors:

Chosen string: ptsd
Nearest neighbors: pstd, depressive, mdd, anxiety, dysthymia, bipolar, depression, psychiatric, dysthymic, mst

Neighbors such as 'depressive', 'mdd', 'anxiety', 'dysthymia' of another chosen string from the test strings, 'ptsd' are all related to Posttraumatic stress disorder (PTSD). If we look at

the following sentence we can see how these strings contextually appeared in BVA decisions:
*“ … for an acquired **psychiatric** disorder, to include posttraumatic stress disorder (**PTSD**), **anxiety**, …, **dysthymic** disorder, **depression**, atypical **depression**, … disorder with psychotic feature.”*
However, the closest neighbor ‘pstd’ does not make sense and no matching sentence was found in the whole train, dev and test texts. This phenomena is hard to explain and can be caused due to mistakes of writing PSTD instead of PTSD in texts.

Chosen string: korea
Nearest neighbors: korean, dmz, demilitarized, germany, demilitarize, vietnam, station, panama, seoul, rvn

The third example from the test words is ‘korea’ and the neighbors like ‘korean’, ‘dmz’ short for demilitarized zone, ‘demilitarized’ are all related to veteran’s history in Korea. The following instance provides a strong argument in favor of this:
*“The Veteran's squamous cell carcinoma, ... …near the **Korean Demilitarized** Zone (**DMZ**).”*
Other neighbors such as names of countries and places like ‘germany’, ‘vietnam’, ‘panama’, ‘seoul’ also depict contextual similarity.

Chosen string: v.
Nearest neighbors: mercy, sulfur, sulfa, berry, harry, split, retired, dmd, larry, furnish

The last example from the provided test strings is ‘v.’. This is the string for which the neighbors do not make much sense. The string ‘v.’ mostly appears while citing cases like *Pelegrini v. Principi, 18 Vet. App. 112 (2004)*. One obvious reason behind this is the removal of punctuations while tokenizing the texts and a single letter ‘v’ bears no significant meaning for the embedding model.

Chosen string: Hickson
Nearest neighbors: hickson, shedden, caluza, davidson, wallin, pond, savage, gutierrez, table, boyer

Apart from the given test words, I chose a string containing a name, ‘Hickson’ to test its neighbors and it shows that almost all of these neighbors are strings containing names of individuals and thus makes sense.

Chosen string: testimony
Nearest neighbors: swear, lay, sworn, undersigned, occurrence, corroborative, undersign, lie, video, videoconference

Another handpicked word for the evaluation is ‘testimony’, and the neighbors such as ‘swear’, ‘lay’, ‘sworn’, etc. are contextually proximal to ‘testimony’. One interesting neighbor is ‘videoconference’ and I found that this also appeared with testimony in a sentence where the veteran presented his testimony in a videoconference. The following sentences are the proofs of this:

- *A Veteran's **lay testimony** … to establish the **occurrence** of his claimed in-service stressor …was not related to it.*
- *… presented **testimony** before the **undersigned** … in a **videoconference** hearing in April 2011.*

From the custom embedding analysis on the provided test and the handpicked string, it is evident that the embedder successfully captures the essence of the legal domain and the neighbors strongly reflect this fact.

Training and Tuning Classifiers

TFIDF and word embedding featurization
I extended the classifier workshop code to enhance the TFIDF feature vectors by adding normalized start position and token count of a sentence to the original TFIDF features to produce an enhanced feature of size 3052. I also enhanced the word embedding features by producing an averaged token feature of all the tokens in a sentence and adding the normalized start position and token count to it.

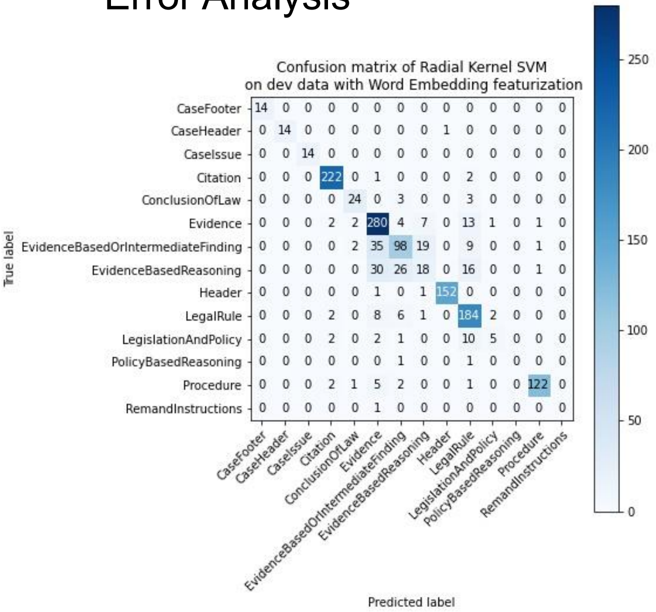
Performance of different linear and nonlinear models on the dev set

Featurizer	Model Type	Model	Kernel	Max Iter	Max depth	No of trees	F1 Score	Comments on the model
TFIDF	Linear	Linear SVM	linear	5000			0.81	Slightly Overfitted
		Log Reg		5000			0.82	Balanced
	Non linear	SVM	rbf				0.83	Best model
		SVM	poly				0.73	Under Fitted
		SVM	sigmoid				0.58	Under Fitted
		DT			None		0.72	Highly overfitted
		DT			10		0.72	Balanced but under fitted
		DT			15		0.73	Overfitted
		RF			10	50	0.61	Under Fitted
		RF			30	100	0.77	Overfitted
		RF			20	50	0.74	Slightly overfitted
Word embedding	Linear	Linear SVM	linear	5000			0.82	Balanced
		Log Reg		5000			0.82	Balanced
	Non linear	SVM	rbf				0.83	Best model
		SVM	poly				0.81	Close to the best
		SVM	sigmoid				0.65	underfitted6
		DT			None		0.70	Highly overfitted
		DT			15		0.72	under fitted
		DT			5		0.66	Highly under fitted
		RF			10	50	0.79	under fitted
		RF			30	100	0.81	Highly overfitted
		RF			5	50	0.65	Highly under fitted

Model performance comparison and decision on the best model
After experimenting with several linear and nonlinear models and putting considerable effort in tuning the hyperparameters of the nonlinear models to counter overfitting, I found SVM

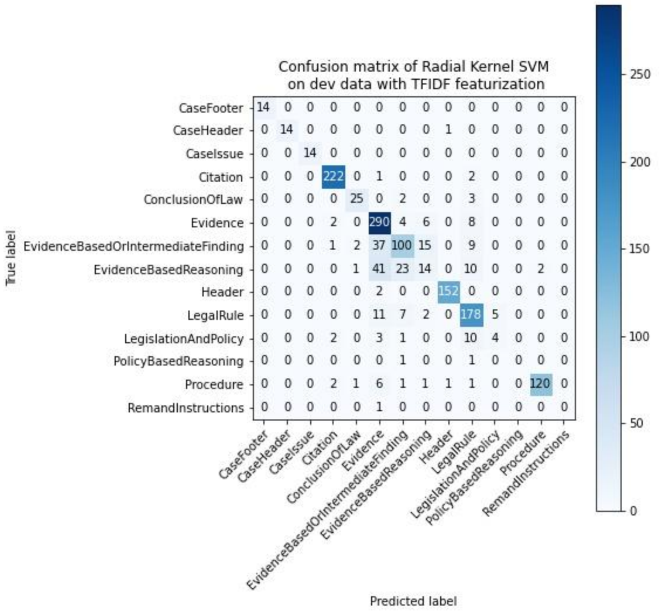
with radial basis function kernel to be the best performing model for both TFIDF and word embedding featurization with similar F1 scores on dev set (0.83) and test set (0.84). However, the word-embedding-based model exhibited balanced performance, whereas the TFIDF based model exhibited overfitting. Although both models produce the same F1 score on the dev set, their performances vary across different classes. For instance, both models exhibit difficulties in classifying types: EvidenceBasedReasoning, EvidenceBasedOrIntermediateFinding, and LegislationAndPolicy, but the embedding-based model performs better than the TFIDF based model on EvidenceBasedReasoning and LegislationAndPolicy classes, and TFIDF based model performs slightly better on EvidenceBasedOrIntermediate Finding. The embedding-based model outperforms the TFIDF based model on ConclusionOfLaw, Evidence, and LegalRule classes. Both models struggle to classify RemandInstructions and PolicyBasedReasoning types due to their very low number of supports. Considering these comparisons I chose the embedding-based model as my best-performing model.

Error Analysis



Confusion matrix of Radial Kernel SVM
on dev data with Word Embedding featurization

	CaseFooter	CaseHeader	CaseIssue	Citation	ConclusionOfLaw	Evidence	EvidenceBasedOrIntermediateFinding	EvidenceBasedReasoning	Header	LegalRule	LegislationAndPolicy	PolicyBasedReasoning	Procedure	RemandInstructions
CaseFooter	14	0	0	0	0	0	0	0	0	0	0	0	0	0
CaseHeader	0	14	0	0	0	0	0	0	1	0	0	0	0	0
CaseIssue	0	0	14	0	0	0	0	0	0	0	0	0	0	0
Citation	0	0	0	222	0	1	0	0	0	2	0	0	0	0
ConclusionOfLaw	0	0	0	0	24	0	3	0	0	3	0	0	0	0
Evidence	0	0	0	2	2	280	4	7	0	13	1	0	1	0
EvidenceBasedOrIntermediateFinding	0	0	0	0	2	35	98	19	0	9	0	0	1	0
EvidenceBasedReasoning	0	0	0	0	0	30	26	18	0	16	0	0	1	0
Header	0	0	0	0	0	1	0	1	152	0	0	0	0	0
LegalRule	0	0	0	2	0	8	6	1	0	184	2	0	0	0
LegislationAndPolicy	0	0	0	2	0	2	1	0	0	10	5	0	0	0
PolicyBasedReasoning	0	0	0	0	0	0	1	0	0	1	0	0	0	0
Procedure	0	0	0	2	1	5	2	0	0	1	0	0	122	0
RemandInstructions	0	0	0	0	0	1	0	0	0	0	0	0	0	0



Confusion matrix of Radial Kernel SVM
on dev data with TFIDF featurization

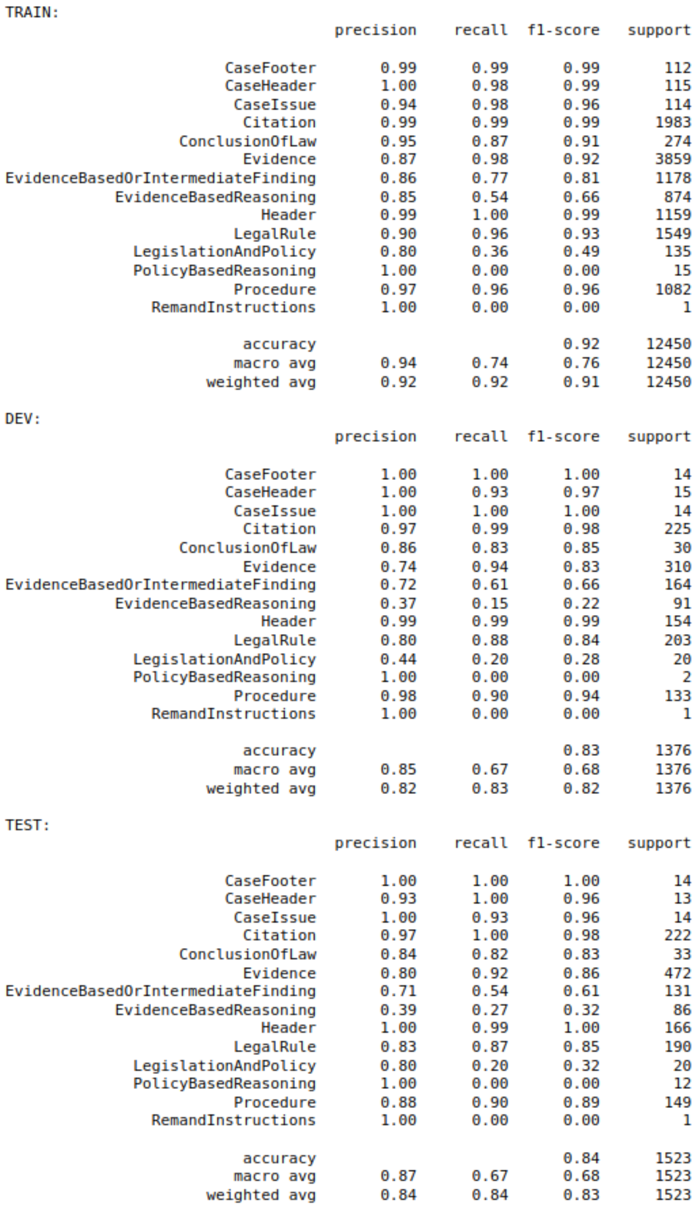
	CaseFooter	CaseHeader	CaseIssue	Citation	ConclusionOfLaw	Evidence	EvidenceBasedOrIntermediateFinding	EvidenceBasedReasoning	Header	LegalRule	LegislationAndPolicy	PolicyBasedReasoning	Procedure	RemandInstructions
CaseFooter	14	0	0	0	0	0	0	0	0	0	0	0	0	0
CaseHeader	0	14	0	0	0	0	0	0	1	0	0	0	0	0
CaseIssue	0	0	14	0	0	0	0	0	0	0	0	0	0	0
Citation	0	0	0	222	0	1	0	0	0	2	0	0	0	0
ConclusionOfLaw	0	0	0	0	25	0	2	0	0	3	0	0	0	0
Evidence	0	0	0	2	0	290	4	6	0	8	0	0	0	0
EvidenceBasedOrIntermediateFinding	0	0	0	1	2	37	100	15	0	9	0	0	0	0
EvidenceBasedReasoning	0	0	0	0	1	41	23	14	0	10	0	0	2	0
Header	0	0	0	0	0	2	0	0	152	0	0	0	0	0
LegalRule	0	0	0	0	0	11	7	2	0	178	5	0	0	0
LegislationAndPolicy	0	0	0	2	0	3	1	0	0	10	4	0	0	0
PolicyBasedReasoning	0	0	0	0	0	0	1	0	0	1	0	0	0	0
Procedure	0	0	0	2	1	6	1	1	1	1	0	0	120	0
RemandInstructions	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Best model: As mentioned above, my chosen best model is SVM with a rbf kernel which produced a F1 score of 0.84 on the test data.

Most prominent examples of mispredictions: Mispredictions by the model were most prominent in three classes:

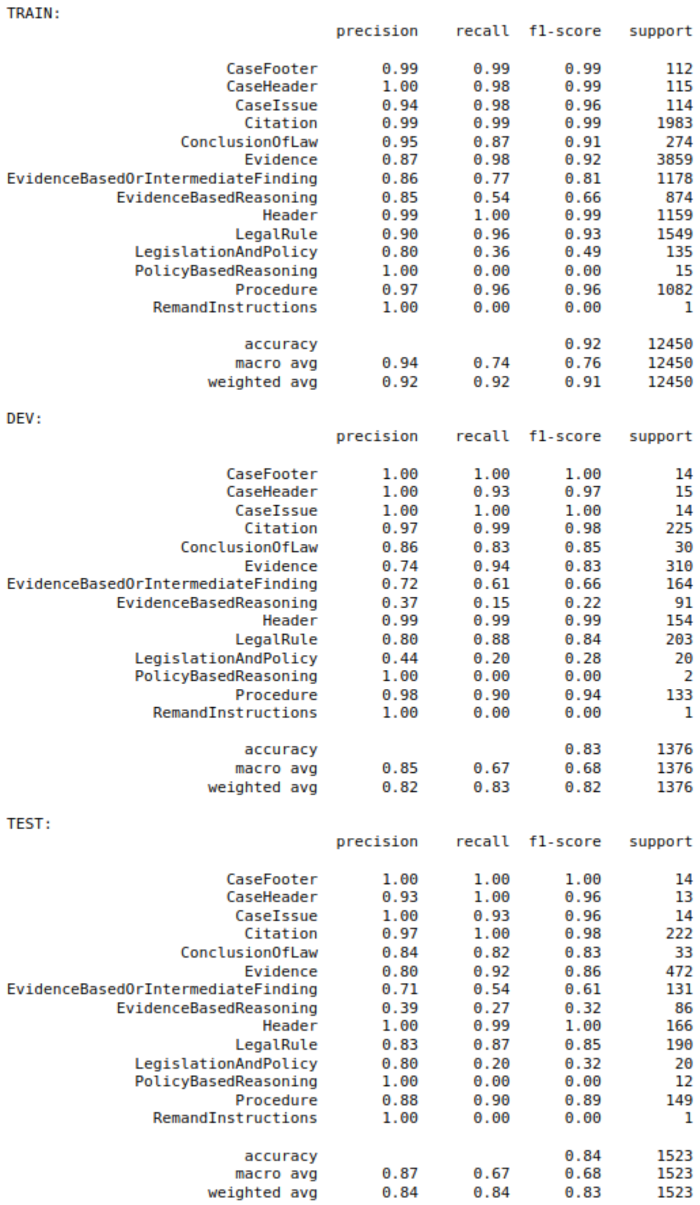
EvidenceBasedReasoning: Most false positives(FP) are with classes Evidence and EvidenceBasedOrIntermediateFinding.

- Misclassification due to annotation error: One of the chosen sentences *“The Veteran is certainly competent to report as to the observable symptoms he experiences, such as an onset of eye problems, and their history.”* was annotated as LegalRule which does not seem to be correct and looks more like an EvidenceBasedReasoning.
- True misclassification: “*Only independent medical evidence may be considered to support Board findings.”,* originally annotated as LegalRule but misclassified as EvidenceBasedReasoning. One common phenomenon observed is the tendency to misclassify if the word ‘evidence’ is present in the sentence. Some common phrases present in the EvidenceBasedReasoning sentences are ‘therefore’, ‘finds’, ‘in this regard’, etc. The model should put more attention to these words which indicate reasoning based on evidence and also the model should not put much focus on the word ‘evidence’ which often result in false positives.



```
TRAIN:
                                      precision    recall  f1-score   support

                          CaseFooter       0.99      0.99      0.99       112
                          CaseHeader       1.00      0.98      0.99       115
                           CaseIssue       0.94      0.98      0.96       114
                            Citation       0.99      0.99      0.99      1983
                     ConclusionOfLaw       0.95      0.87      0.91       274
                            Evidence       0.87      0.98      0.92      3859
EvidenceBasedOrIntermediateFinding       0.86      0.77      0.81      1178
              EvidenceBasedReasoning       0.85      0.54      0.66       874
                              Header       0.99      1.00      0.99      1159
                            LegalRule       0.90      0.96      0.93      1549
                LegislationAndPolicy       0.80      0.36      0.49       135
                PolicyBasedReasoning       1.00      0.00      0.00        15
                           Procedure       0.97      0.96      0.96      1082
                  RemandInstructions       1.00      0.00      0.00         1

                            accuracy                           0.92     12450
                           macro avg       0.94      0.74      0.76     12450
                        weighted avg       0.92      0.92      0.91     12450

DEV:
                                      precision    recall  f1-score   support

                          CaseFooter       1.00      1.00      1.00        14
                          CaseHeader       1.00      0.93      0.97        15
                           CaseIssue       1.00      1.00      1.00        14
                            Citation       0.97      0.99      0.98       225
                     ConclusionOfLaw       0.86      0.83      0.85        30
                            Evidence       0.74      0.94      0.83       310
EvidenceBasedOrIntermediateFinding       0.72      0.61      0.66       164
              EvidenceBasedReasoning       0.37      0.15      0.22        91
                              Header       0.99      0.99      0.99       154
                            LegalRule       0.80      0.88      0.84       203
                LegislationAndPolicy       0.44      0.20      0.28        20
                PolicyBasedReasoning       1.00      0.00      0.00         2
                           Procedure       0.98      0.90      0.94       133
                  RemandInstructions       1.00      0.00      0.00         1

                            accuracy                           0.83      1376
                           macro avg       0.85      0.67      0.68      1376
                        weighted avg       0.82      0.83      0.82      1376

TEST:
                                      precision    recall  f1-score   support

                          CaseFooter       1.00      1.00      1.00        14
                          CaseHeader       0.93      1.00      0.96        13
                           CaseIssue       1.00      0.93      0.96        14
                            Citation       0.97      1.00      0.98       222
                     ConclusionOfLaw       0.84      0.82      0.83        33
                            Evidence       0.80      0.92      0.86       472
EvidenceBasedOrIntermediateFinding       0.71      0.54      0.61       131
              EvidenceBasedReasoning       0.39      0.27      0.32        86
                              Header       1.00      0.99      1.00       166
                            LegalRule       0.83      0.87      0.85       190
                LegislationAndPolicy       0.80      0.20      0.32        20
                PolicyBasedReasoning       1.00      0.00      0.00        12
                           Procedure       0.88      0.90      0.89       149
                  RemandInstructions       1.00      0.00      0.00         1

                            accuracy                           0.84      1523
                           macro avg       0.87      0.67      0.68      1523
                        weighted avg       0.84      0.84      0.83      1523
```

Classification report of the best model with word embedding features



```
TRAIN:
                                      precision    recall  f1-score   support

                          CaseFooter       0.99      0.99      0.99       112
                          CaseHeader       1.00      0.98      0.99       115
                           CaseIssue       0.94      0.98      0.96       114
                            Citation       0.99      0.99      0.99      1983
                     ConclusionOfLaw       0.95      0.87      0.91       274
                            Evidence       0.87      0.98      0.92      3859
EvidenceBasedOrIntermediateFinding       0.86      0.77      0.81      1178
              EvidenceBasedReasoning       0.85      0.54      0.66       874
                              Header       0.99      1.00      0.99      1159
                            LegalRule       0.90      0.96      0.93      1549
                LegislationAndPolicy       0.80      0.36      0.49       135
                PolicyBasedReasoning       1.00      0.00      0.00        15
                           Procedure       0.97      0.96      0.96      1082
                  RemandInstructions       1.00      0.00      0.00         1

                            accuracy                           0.92     12450
                           macro avg       0.94      0.74      0.76     12450
                        weighted avg       0.92      0.92      0.91     12450

DEV:
                                      precision    recall  f1-score   support

                          CaseFooter       1.00      1.00      1.00        14
                          CaseHeader       1.00      0.93      0.97        15
                           CaseIssue       1.00      1.00      1.00        14
                            Citation       0.97      0.99      0.98       225
                     ConclusionOfLaw       0.86      0.83      0.85        30
                            Evidence       0.74      0.94      0.83       310
EvidenceBasedOrIntermediateFinding       0.72      0.61      0.66       164
              EvidenceBasedReasoning       0.37      0.15      0.22        91
                              Header       0.99      0.99      0.99       154
                            LegalRule       0.80      0.88      0.84       203
                LegislationAndPolicy       0.44      0.20      0.28        20
                PolicyBasedReasoning       1.00      0.00      0.00         2
                           Procedure       0.98      0.90      0.94       133
                  RemandInstructions       1.00      0.00      0.00         1

                            accuracy                           0.83      1376
                           macro avg       0.85      0.67      0.68      1376
                        weighted avg       0.82      0.83      0.82      1376

TEST:
                                      precision    recall  f1-score   support

                          CaseFooter       1.00      1.00      1.00        14
                          CaseHeader       0.93      1.00      0.96        13
                           CaseIssue       1.00      0.93      0.96        14
                            Citation       0.97      1.00      0.98       222
                     ConclusionOfLaw       0.84      0.82      0.83        33
                            Evidence       0.80      0.92      0.86       472
EvidenceBasedOrIntermediateFinding       0.71      0.54      0.61       131
              EvidenceBasedReasoning       0.39      0.27      0.32        86
                              Header       1.00      0.99      1.00       166
                            LegalRule       0.83      0.87      0.85       190
                LegislationAndPolicy       0.80      0.20      0.32        20
                PolicyBasedReasoning       1.00      0.00      0.00        12
                           Procedure       0.88      0.90      0.89       149
                  RemandInstructions       1.00      0.00      0.00         1

                            accuracy                           0.84      1523
                           macro avg       0.87      0.67      0.68      1523
                        weighted avg       0.84      0.84      0.83      1523
```

Classification report of the best model with TF-IDF features

EvidenceBasedOrIntermediateFinding: The model often confuses it with EvidenceBasedReasoning, Evidence, etc. due to similarity of languages in these types.

- Misclassification due to annotation error: *“Thus, the representative argues that 38 U.S.C.A. � 1154(b) applies.”* was originally annotated as ‘Evidence’ which seems not correct.
- True misclassification: One example of misclassification is “*There is no indication that there exists any additional evidence that has a bearing on this case that has not been obtained.*“, which was originally labeled as ‘Evidence’ but predicted with this discussed class. This is a quite complicated sentence to classify as it is really hard to distinguish between these classes. The model mostly falls into the trap of words such as ‘indication’, ‘finds’, ‘concludes’, etc. and I think the model’s performance for such cases will be limited to the annotation quality produced by the annotators. I would suggest putting more emphasis on reducing human error while annotating sentences with semantic overlapping with other types.

LegislationAndPolicy: The model mostly produces FPs for the class ‘LegalRule’ and this is understandable since the wording of ‘LegalRule’ sentences have high similarity with ‘LegislationAndPolicy’.

- Misclassification due to annotation error: The same sentence “*The Veterans Claims Assistance Act of 2000 as amended (VCAA) and implementing regulations impose obligations on VA to provide claimants with notice and assistance.”* is classified as ‘LegalRule’ in case **1548317.txt(start index: 2447)** whereas it has been annotated as “LegislationAndPolicy’ in case **0640279.txt (start index 1165)**. Due to this type of human error in misclassification, it is extremely difficult for the model to learn to distinguish between these types.

Mispredictions on other classes

Citation: Some evidence containing sentences are often tagged as ‘Citation’ . For instance, “*The Board has considered the Veterans Claims Assistance Act of 2000, 38 U.S.C.A. �� 5100 et. seq. (West 2002), including the notice and assistance requirements of Dingess/Hartman v. Nicholson, 19 Vet. App. 473 (2006) and Kent v. Nicholson, 20 Vet. App. 1 (2006).”* was classified as ‘Citation’ and the amount of references in this sentence renders it very difficult for the model to classify correctly. Moreover, there are also examples of wrong annotation, such as *“HERTZ 50...LEFT4545506070”,* whose original label was ‘Evidence’ which clearly is not.

Header: There are some instances of labeling a ‘Header’ as ‘CaseHeader’ such as *“THE ISSUE”,* which was predicted as a ‘CaseHeader’, but it should be a ‘Header’.

LegalRule: There have been some instances of classes ‘LegislationAndPolicy’, ‘EvidenceBasedReasoning’, ‘Evidence’, ‘EvidenceBasedOrIntermediateFinding’, ‘ConclusionOfLaw’, etc. being classified as ‘LegalRule’. After investigating some sentences, I am pretty convinced that this is due to the similarity of the wordings among these types of sentences. Also there are some wrong annotations, one example of which was shown while describing ‘LegislationAndPolicy’. There are also some misclassifications of type ‘Citation’ as ‘LegalRule’, which were caused by both incorrect annotation and actual mispredictions. For instance, “*38 C.F.R. � 3.385. Use of 38 C.F.R. � 3.385 to define a disability under 38 U.S.C.A. � 1110 as it pertains to hearing loss has been recognized by the Court as a reasonable interpretation of the statute.”* was originally labeled as ‘Citation’ which clearly is not and most likely is describing a ‘LegalRule’.

Apart from all these classes, the model had 0 F1 score on dev set for ‘PolicyBasedReasoning’ and ‘RemandInstructions’. The reason can be attributed to the fact that these two classes had a very low number of supports and the calculations were based on false negative instances. Moreover, there was an error in the annotation of a ‘RemanInstructions’ sentence, such as, *“However, because service connection is granted by this decision for residuals of a stroke, the RO should reconsider this matter and both claims are referred to the RO for initial consideration.”* which does not seem like a ‘RemandInstructions’ sentence. Same goes for the class 'PolicyBasedReasoning' which has an incomplete sentence *“In considering the evidence of record under the laws and regulations as set forth above, the Board concludes that“* which was annotated as 'PolicyBasedReasoning'. There are also instances of FNs where ‘PolicyBasedReasoning’ instances have been classified as ‘EvidenceBasedOrIntermediateFinding’ and this is often due to their semantic similarities.

Discussion

To fulfill the analytical objective of this project, an extensive multi-step analysis was carried out within a course project scope. First, a sentence segmenter was developed for sentence segmenting the provided unlabeled data corpus. Savelka’s law-specific sentence segmenter LUIMA was used for this which, apart from its some over-segmentation issues, produced the most reliable segmentations. Although LUIMA performed satisfactorily, its segmentation can be improved vastly for punctuations and multi-line segments with empty spaces to reduce over-segmentation problems. After sentence segmenting the corpus, Spacy's tokenizer function was extended for tokenizing the segments as a preprocessing step before feeding them to a word embedding vectorizer for capturing the contextual relations of different words found in the legal texts. In this project, two types of word embedding techniques were used, TFIDF and FastText word embedding model. After analyzing the word vectors with different words and their neighbors, it was observed that the embedding-based word vector captured the essence of legal terms better. Finally, several linear and nonlinear classifiers were trained on the developed feature vectors and it was found that word embedding based nonlinear SVM with rbf kernel performed the best and generalized the unseen data better. However, there are ample scopes for improvements to increase the overall performance. Some recommendations are improving the law-specific segmenter for more reliable segmentations, designing a law-specific tokenizer function by thoroughly examining the exceptions and special cases of legal texts which can save valuable time and improve the quality of word embedding. Also, the idea of building a law-specific meta-sentence embeddings using BERT based models from Xu et. al can be worth exploring [3]. Since a nonlinear model produces the best performance, deep neural network-based models are worth exploring. Finally, a supervised model is as good as the data used for learning, hence the annotation qualities should be improved and more emphasis should be put on producing high-quality annotations. The idea of Westerman et. al of building an annotation scheme based on semantic similarity of documents can assist the annotators and speed up the annotation process by a significant margin [4]. One issue that I have observed is the human annotators as well as the model struggle with classes that have high semantic similarities such as 'Evidence', 'EvidenceBasedReasoning', 'EvidenceBasedOrIntermediateFinding', hence the purpose of creating separate classes for such sentences should be reiterated and if possible these types of ambiguities should be avoided while designing a type system.

Lessons Learned

This project provided an excellent hands-on learning experience of experimental NLP with the challenging task of sentence classification in legal documents. The main challenge for me was improving the sentence segmenter which could have been improved more to beat the performance of LUIMA provided more time. Designing a satisfactory tokenizer and finding the optimal hyperparameters for the classification model were also challenging as they required scrupulous examination of different documents. The workshop notebook and the detailed project instructions helped a lot to achieve the deliverables. One suggestion regarding the project report template is to provide a Latex template in the future as it makes the report writing part more convenient.

Code Instructions

After decompressing just run the following commands inside the **code** folder in the terminal:

```
1. python3 -m venv test #Used Python 3.6.9 originally
2. source test/bin/activate
3. pip3 install -U pip setuptools
4. pip3 install -r requirements.txt
5. python3 -m spacy download en
6. python3 analyze.py <path to txt file>
```

Six notebook files are also provided in the **notebooks** directory to showcase the effort put in analysis. To run the notebooks please follow **steps 1-3**, then run `pip3 install -r requirements-notebook.txt` and finally run **step 5** which should allow you to execute the notebooks successfully.

References

[1] Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python. https://doi.org/10.5281/zenodo.1212303

[2] Savelka, Jaromir, et al. "Sentence boundary detection in adjudicatory decisions in the united states." Traitement automatique des langues 58 (2017): 21.

[3] Xu, Huihui, Jaromir Savelka, and Kevin D. Ashley. "Accounting for Sentence Position and Legal Domain Sentence Embedding in Learning to Classify Case Sentences." Legal Knowledge and Information Systems. IOS Press, 2021. 33-42.

[4] Villata, S. "Sentence embeddings and high-speed similarity search for fast computer assisted annotation of legal documents." Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020. Vol. 334. IOS Press, 2020.