LDSI S2021 Course Project Report

Name: Irtiza Chowdhury
Gloss ID: ldsi_s2_18

Summary

This project aims to explore classification of types in annotated sentences of documents from US Board of Veteran's Appeals' decisions. Different segmenters are observed to select the optimal performing one, followed by tokenization and developing of a custom word embedding model from an unlabeled corpus. Finally, different models are used using TFIDF and word embedding featurizations to find the best performing according to some metrics.

1. Dataset Splitting

The dataset was split into train, dev and test set with the following ids in the respective sets.

DEV: '1631703.txt', '0601461.txt', '1522066.txt', '0806464.txt', '1311391.txt', '1235794.txt', '0805869.txt', '1414169.txt', '1624269.txt'

TEST: '0934845.txt', '1623087.txt', '1114333.txt', '1317440.txt', '1333883.txt', '0824445.txt', '1007840.txt', '0829439.txt', '1635686.txt'

2. Sentence Segmentation

2.1 SpaCy's sentence segmenter was chosen for this task. The starting and ending offsets of the sentences were used to find out the different kinds of False Positives that could arise. The following results (correct to 4 d.p) were observed after applying on the training set.
Precision: 0.2591, **Recall**: 0.5777, **F1 Score**: 0.3578

The documents with the worst scores are given below:

Document ID	Precision	Recall
60b606d7f8611168dd279d16	0.12598425196850394	0.2962962962962963
60b606d9f8611168dd279d44	0.14545454545454545	0.26666666666666666
60b606d8f8611168dd279d2f	0.16822429906542055	0.43902439024390244
60b606cbf8611168dd279cd1	0.18137254901960784	0.3592233009708738

Table 1: Precision and Recall for four worst performing documents

A general pattern across these documents was observed. The segmenter struggled with the upper case headers as well as the extra spaces surrounding them. Punctuations, such as period and parentheses, caused a lot of oversplitting which even resulted in no citation being able to be correctly segmented. The same was observed with the dates. Overall, the majority of false positives arose from oversplitting in between large sentences. This is somewhat

expected as legal texts are known for long sentences. Followed by generated splits starting within a sentence and then by generated ones starting in between true splits.

2.2 Some custom boundaries were added to the segmenter to try to reduce oversplitting. The extra spaces, tabs, carriage returns and quotations were considered. Furthermore, some words in the case header, such as 'DOCKET', 'DATE' were also added as exceptions.
An increase in overall Precision to 0.3929 was observed. The new scores for the same worst performing documents are tabulated below.

Document ID	Precision	Recall
60b606d7f8611168dd279d16	0.20454545454545456	0.3333333333333333
60b606d9f8611168dd279d44	0.2	0.26666666666666666
60b606d8f8611168dd279d2f	0.23880597014925373	0.3902439024390244
60b606cbf8611168dd279cd1	0.24087591240875914	0.32038834951456313

Table 2: Precision and Recall for four previous worst performing documents after exceptions

Even though false positives were reduced, true splits were affected for some documents.

2.3 Using Savelka's law-specific sentence segmenter, a significant increase in overall scores was noticeable. Precision jumped to 0.6671, Recall to 0.8530 and F1 Score to 0.7487. The scores for the previously worst documents also saw an impressive rise.

Document ID	Precision	Recall
60b606d7f8611168dd279d16	0.26666666666666666	0.5185185185185185
60b606d9f8611168dd279d44	0.65	0.8666666666666667
60b606d8f8611168dd279d2f	0.5932203389830508	0.8536585365853658
60b606cbf8611168dd279cd1	0.5899280575539568,	0.7961165048543689

Table 3: Precision and Recall for previous four worst performing documents after exceptions after LUIMA SBD

Observing the documents again, more patterns began to emerge. The segmenter still had difficulty with Case Header, e.g. splitting Citation number and Decision date, and Footer, especially the long line (_____). Numbers, especially those starting a list, e.g. 2., 3., were also incorrectly split. Upper case words, but not abbreviations such as PTSD, were still a problem. For example, splitting "in <split> St. Petersburg". However, a significant improvement occurred in splitting headers and citations. Headers were correctly identified and nearly all of them were self-contained. Previous issues with citations such as splitting at periods were almost non-existent.

2.4 From the scores mentioned above, using the LUIMA SBD for segmentation was the obvious choice. Oversplitting in headers occurred regardless, and the issue with numbers was not as prevalent.

3. Preprocessing

3.1

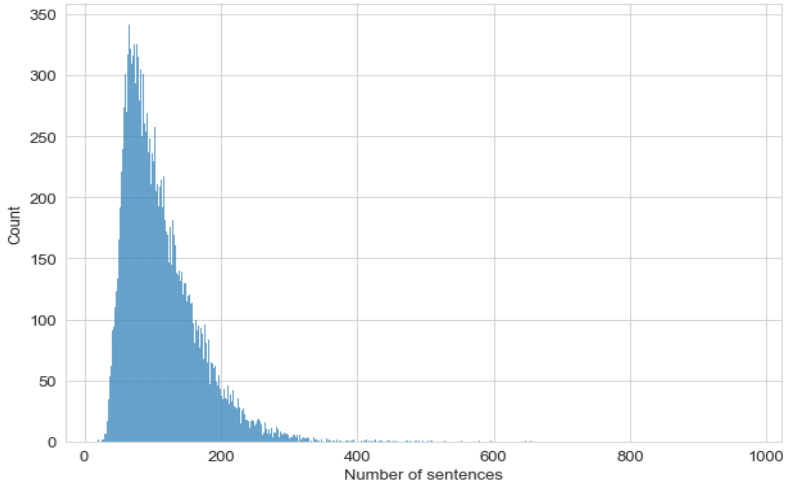


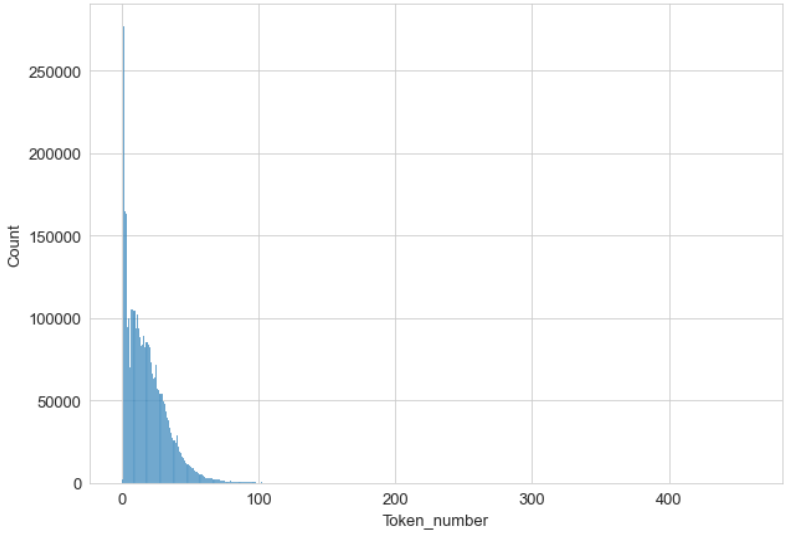
Figure 1: Histogram of number of sentences (bindwidth = 1)

Total number of sentences: 3360513

3.2 When creating the tokenizer, special cases such as ‘Vet. App.' and 'Fed. Cir.' were added for them to be considered together. The tokenizer allowed only alphanumerics and only length of numbers to simplify them with the help of SpaCy’s POS tags. This caused some issues where abbreviations and words with apostrophe were being discarded. Changes were made to discard only the apostrophe from the word to make it a single token. Regular punctuations were discarded accordingly.

The tokenizer was tested on the ‘tough examples’ from the classifier workshop where it produced the desired output such as discarding non-alphanumerics but keeping tokens such as ‘affd’.

3.3 The histogram (bindwidth = 1) of number of tokens in each sentence for the unlabeled data is given below. Tokens of 5 or less were discarded to make a file with randomized sentences.



4. Custom Embeddings

A 100-dimensional word embedding model was trained with minimum word count of 20 for 10 epochs using FastText. During training, over 60M words were shown to be trained on. The vocabulary size was 12262. The top nearest neighbors of the recommended strings were observed along with a few more words to find patterns in several documents.

The closest neighbors of “veteran” were “appellant” and “he”. While it makes sense that the semantic information of the corpus was properly captured as the veteran is also the appellant, it is also interesting to note that “he” had a much higher score than “she”, indicating that a supermajority of the appellants might have been men if not all of them. Connecting words such as “additionally” and “furthermore” also made the list and can be explained by their presence near “veteran” in several sentences related to evidence or reasoning. On the other hand, “vet” had closest neighbors like “app” and “see” which are prevalent in citations.

Predictably, the closest to “service” was “connection”, and other words such as “disability”, “disease” can be explained by medical records and other occurrences during service period.

The strings “cause” and “caused” are quite interesting instances. The closest neighbors for “cause” turned out to be “exacerbate” and “attributable”. Observing few documents, it was noticed that the word is centered around scenarios containing contributing factors of the physical or mental disabilities or disorders the veterans claimed to have had. Typical phrases such as “cause of disability”, “cause of complaints”, “source or the cause of illness”, etc. were present throughout the corpus, sometimes accompanied by the deteriorating conditions. However, this still does not entirely explain why “exacerbate” would be a much nearer neighbor than words like “contributing” or “due”. Their scores were not too far off from each other though. On the other hand, “caused” was close to “cause”, “confused” and “contributor”. The word “confused” appeared in few documents where the veteran claimed to have had some mental disability such as PTSD.

“Vietnam” had the closest neighbors “republic” and “rvn” which is the abbreviation for Republic of Vietnam. These, combined with words like “country” and “thailand”, are self-explanatory within the nature of the corpus.

The word “see” had neighbors such as “vet”, “cf”, several numbers of different lengths which indicates its role in our domain as part of citations. Embedding models not trained on legal domain would most likely have other common synonyms for “see” as its nearest neighbors. A similar word which also reflects our domain is the word “under” as it has nearest neighbors such as “usca”, “code” and “cfr” which also indicates its usage as part of citation in our corpus.

The closest neighbors of “board” were “decision”, “matter”, “appeal”, “bva”, etc. which is expected of it as it is one of the most commonly used words in the corpus along with its neighbors, so the semantic information is captured accordingly. The same can be observed for “decision” and “appeal” as well.

“Evidence” had the closest neighbors “claim”, “medical”, “record”, etc. Our corpus has sentences such as “the veteran has received notice of the evidence necessary to substantiate

his claim” and similar ones tying claims with medical records which make up part of the evidence, so this behavior is expected as well. Other strings which also produced expected neighbors were “physician”, giving “doctor”, “clinician”, and “pain”, resulting in strings like “discomfort”, “aching”, etc.

5. Training & Optimizing Classifiers

TFIDF Featurization: A TFIDF Vectorizer is applied on the training data with document frequency threshold of 3 and unigrams. To build the feature vectors for model training, the normalized position of the sentence and the normalized number of tokens in the sentence were kept along with the TFIDF featurization to see the effect of altering either the TFIDF or the Word embeddings.

Word Embedding Featurization: The feature vector contained the average of the embedding vectors of the tokens as well as the two more to make a fair comparison.

Linear SVM Classifier, Logistic Regression, Radial and Polynomial Kernel SVM, Decision Tree Classifier and Random Forest models were trained on each featurization. The different hyperparameter settings for each non-linear classifier is given below. The random state was 0 for all models.

Radial kernel SVM: gamma = ‘scale’ / gamma = ‘auto’
Polynomial kernel SVM: degree = 3 / degree = 2
Decision Trees: max_depth = 12 / max_depth = 22 / min_samples_split = 10
Random Forests: max_depth = 20 / max_depth = None / n_estimators = 200

Both overfitting and underfitting behaviors were observed for some of these settings.
Overfitting: Random Forest with max_depth = None and Decision Tree with max_depth = 22 and min_samples_split = 10.
Underfitting: Radial Kernel SVM with gamma = auto, Decision Tree with max_depth = 12

6. Test Set Evaluation

Considering the dev accuracy, the embedding featurization gave slightly better results for almost all models except overfitting cases. Top three models for TFIDF were Linear SVC (82), Logistic Regression (83) and Polynomial Kernel SVM (degree =2, 82).
For embeddings, Linear SVC (84), Radial Kernel SVM (gamma = scale, 84) and Polynomial Kernel SVM (degree = 2, 83) were best performing.

Best TFIDF Featurization Model: While observing the best models, Logistic Regression had the highest accuracy. However, after observing the F1 score for individual classes, the class “LegalPolicy” had 0 score in both Logistic Regression and Polynomial Kernel SVM, but Linear SVC had at least 0.31 score even with a support of 4. Also, comparing the macro and weighted average for Precision, Recall and F1, **Linear SVC** stood out more than the other two and was considered the best model thereafter.

Classification Report:

TRAIN:

	precision	recall	f1-score	support
CaseFooter	0.94	0.98	0.96	85
CaseHeader	0.98	0.96	0.97	83
CaseIssue	0.94	1.00	0.97	78
Citation	0.99	1.00	0.99	1057
ConclusionOfLaw	0.89	0.92	0.91	163
Evidence	0.90	0.96	0.93	1926
EvidenceBased/Intermediate Finding	0.85	0.84	0.84	673
EvidenceBasedReasoning	0.88	0.68	0.77	552
Header	0.99	0.99	0.99	726
LegalPolicy	0.88	0.60	0.72	98
LegalRule	0.91	0.93	0.92	844
PolicyBasedReasoning	0.93	0.67	0.78	21
Procedure	0.94	0.93	0.94	822
RemandInstructions	0.93	0.95	0.94	336
accuracy			0.92	7464
macro avg	0.92	0.89	0.90	7464
weighted avg	0.92	0.92	0.92	7464

DEV:

	precision	recall	f1-score	support
CaseFooter	1.00	0.91	0.95	11
CaseHeader	1.00	1.00	1.00	9
CaseIssue	0.90	1.00	0.95	9
Citation	1.00	0.95	0.97	81
ConclusionOfLaw	0.73	0.55	0.63	20
Evidence	0.85	0.93	0.89	370
EvidenceBased/Intermediate Finding	0.43	0.31	0.36	67
EvidenceBasedReasoning	0.16	0.12	0.14	42
Header	0.99	1.00	0.99	75
LegalPolicy	0.22	0.50	0.31	4
LegalRule	0.82	0.73	0.77	73
PolicyBasedReasoning	0.00	0.00	0.00	3
Procedure	0.86	0.88	0.87	95
RemandInstructions	0.78	0.89	0.83	44
accuracy			0.82	903
macro avg	0.70	0.70	0.69	903
weighted avg	0.80	0.82	0.81	903

TEST:

	precision	recall	f1-score	support
CaseFooter	1.00	1.00	1.00	10
CaseHeader	1.00	1.00	1.00	9
CaseIssue	0.73	1.00	0.84	8
Citation	0.95	0.99	0.97	117
ConclusionOfLaw	0.85	0.81	0.83	21
Evidence	0.78	0.87	0.83	229
EvidenceBased/Intermediate Finding	0.51	0.55	0.53	58
EvidenceBasedReasoning	0.41	0.29	0.34	69
Header	0.97	0.99	0.98	76
LegalPolicy	0.33	0.11	0.17	9
LegalRule	0.79	0.76	0.77	111
PolicyBasedReasoning	0.00	0.00	0.00	4
Procedure	0.85	0.83	0.84	108
RemandInstructions	0.86	0.82	0.84	44
accuracy			0.80	873
macro avg	0.72	0.72	0.71	873
weighted avg	0.79	0.80	0.79	873

Best Word Embedding Featurization Model: Similarly, the accuracies and class level F1 scores were taken into account for the embedding featurization as well. Radial Kernel SVM was chosen to be the best model with the classification report given below.

TRAIN:

	precision	recall	f1-score	support
CaseFooter	0.95	0.95	0.95	85
CaseHeader	0.97	0.94	0.96	83
CaseIssue	0.91	0.96	0.94	78
Citation	0.97	0.99	0.98	1057
ConclusionOfLaw	0.82	0.74	0.77	163
Evidence	0.76	0.93	0.84	1926
EvidenceBased/Intermediate Finding	0.58	0.55	0.56	673
EvidenceBasedReasoning	0.51	0.23	0.31	552
Header	0.99	0.99	0.99	726
LegalPolicy	0.76	0.19	0.31	98
LegalRule	0.78	0.86	0.82	844
PolicyBasedReasoning	0.00	0.00	0.00	21
Procedure	0.90	0.88	0.89	822
RemandInstructions	0.87	0.70	0.77	336
accuracy			0.82	7464
macro avg	0.77	0.71	0.72	7464
weighted avg	0.81	0.82	0.80	7464

DEV:

	precision	recall	f1-score	support
CaseFooter	1.00	0.91	0.95	11
CaseHeader	1.00	1.00	1.00	9
CaseIssue	1.00	1.00	1.00	9
Citation	1.00	0.98	0.99	81
ConclusionOfLaw	0.78	0.70	0.74	20
Evidence	0.84	0.96	0.90	370
EvidenceBased/Intermediate Finding	0.41	0.27	0.32	67
EvidenceBasedReasoning	0.21	0.12	0.15	42
Header	1.00	1.00	1.00	75
LegalPolicy	1.00	0.25	0.40	4
LegalRule	0.82	0.81	0.81	73
PolicyBasedReasoning	0.00	0.00	0.00	3
Procedure	0.90	0.91	0.90	95
RemandInstructions	0.88	0.82	0.85	44
accuracy			0.84	903
macro avg	0.77	0.69	0.71	903
weighted avg	0.81	0.84	0.82	903

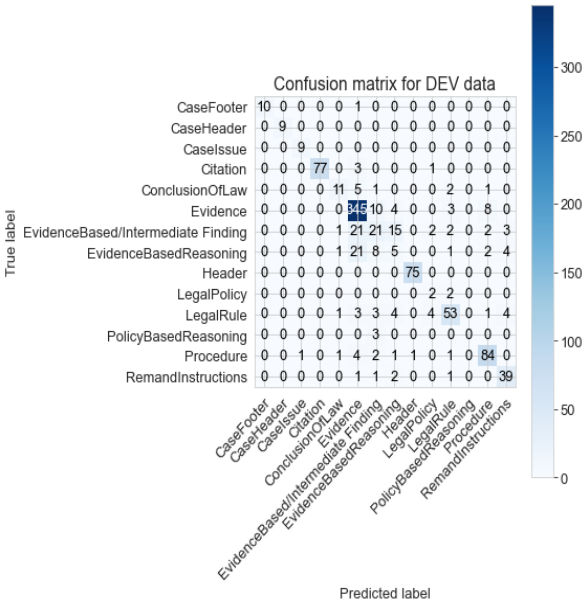
TEST:

	precision	recall	f1-score	support
CaseFooter	1.00	1.00	1.00	10
CaseHeader	1.00	1.00	1.00	9
CaseIssue	0.73	1.00	0.84	8
Citation	0.93	0.97	0.95	117
ConclusionOfLaw	0.79	0.71	0.75	21
Evidence	0.75	0.94	0.84	229
EvidenceBased/Intermediate Finding	0.46	0.52	0.49	58
EvidenceBasedReasoning	0.63	0.25	0.35	69
Header	0.97	0.97	0.97	76
LegalPolicy	0.50	0.11	0.18	9
LegalRule	0.78	0.74	0.76	111
PolicyBasedReasoning	0.00	0.00	0.00	4
Procedure	0.92	0.85	0.88	108
RemandInstructions	0.85	0.77	0.81	44
accuracy			0.80	873
macro avg	0.74	0.70	0.70	873
weighted avg	0.80	0.80	0.79	873

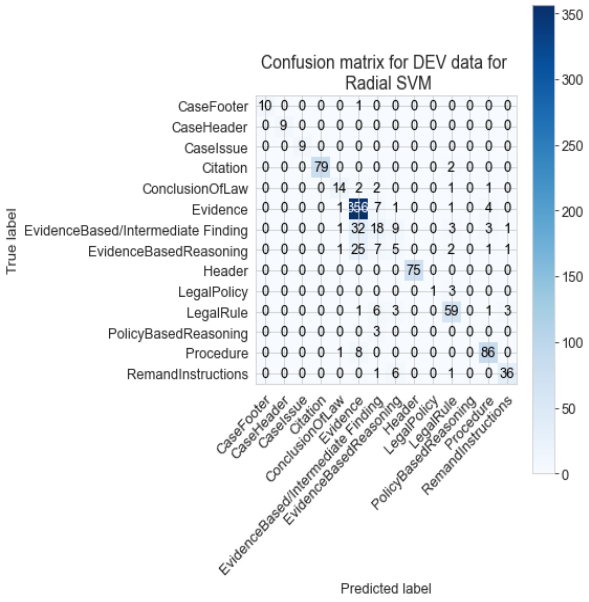
Observing the performance on **dev** set, Radial Kernel SVM had equal or better F1 scores than Linear SVC for all classes. Interestingly, the situation is completely flipped when looking at the training set, as Linear SVC had much higher scores. The results were not one-sided for the test set, as the TFIDF featurization gave better scores for some while the Word embedding featurization for others if not equal, showing absence of overfitting. However, if we strictly consider the dev set, we can see that Radial Kernel SVM generalizes better. “ConclusionOfLaw” had the highest difference in score in favor of Radial SVM. The worst classes for both classifiers were “EvidenceBased/Intermediate Finding”, “EvidenceBasedReasoning” and “PolicyBasedReasoning”.

7. Error Analysis

Confusion Matrix for TFIDF model



Confusion Matrix for Embedding model



Based on the reasoning given, as well as evident from the confusion matrices, the **Radial Kernel SVM** with word embedding featurization was selected to be the best model. The three most difficult types were “EvidenceBased/Intermediate Finding”, “EvidenceBasedReasoning” and “PolicyBasedReasoning”. Some false positives for these types were observed.

EvidenceBased/Intermediate Finding: The supermajority of false positives arose from confusion between findings and reasonings. This is somewhat expected and to some level is acceptable as the confusion was common even within annotators. Linguistically complex sentences seemed to be hard to classify, such as, “The only question before the Board is whether the Veteran's current condition is related to his exposure to herbicides during service.”. There seemed to be misclassifications in some cases where there shouldn’t have been any confusion. e.g., “New and material evidence sufficient to reopen the previously denied claim of service connection for a skin disorder has not been received, the appeal is denied.” A lack of trigger words such as “finds” as well as the linguistically diverse nature of the corpus, e.g., two different sentences starting with “The Board acknowledges…” are annotated as both “Reasoning” and “Finding” in different occasions, might be the reason for

this level of misclassification. Annotation errors were present as well because “Thus, any deficiency in VA's compliance is deemed to be harmless error…” was annotated as LegalRule.

EvidenceBasedReasoning: Similar to the previous type, the majority of conflicts occurred with findings. Sentences like “As the March 1979 audiogram indicates at least some level of hearing loss at service separation, and in the absence of any competent negative medical opinion, the Board finds the…” were misclassified, indicating the model could not identify the importance of the keyword “find”. Difficulty in capturing context was also evident, as “As a layperson, the veteran is not competent to render a medical opinion in this regard.” which belongs to a LegalRule was also misclassified.

PolicyBasedReasoning: This type had the lowest support in the training set, while the second lowest was four times higher, leading to the poor results. No classifier was correctly able to identify it in the dev set. However, the type with second lowest support (CaseHeader) had really good score, indicating the location of sentences is just as important as evident from the literature survey.

ConclusionOfLaw: Some annotation mistakes were noticed where the following was not annotated as conclusion, such as “Accordingly, this case is REMANDED for the following actions:” and “Therefore, service connection for post-traumatic stress disorder is granted.” which indicated the classifier was able to identify sentences where the decision is denoted to an acceptable degree.

RemandInstructions: This type was mostly confused with LegalRule which can be explained by their linguistic and semantic similarity. A lack of context captured by the classifier could be the cause of this misclassification although some presence of keywords such as in the sentence, “While on Remand, a request should be made for those records.” which is annotated as LegalRule could also be attributed.

LegalPolicy: A low support and semantic similarity with LegalRule could be the main reasons for misclassifications. e.g., “The Veterans Claims Assistance Act of 2000 (VCAA) describes VA's duty to notify and assist claimants in substantiating a claim for VA benefits.” A wrong annotation was also noticed as “The Board has given consideration to the Veterans Claims Assistance Act of 2000 (VCAA).” Is annotated as Evidence.

8. Discussion

Considering the objective of automating sentence annotations for BVA decisions, few conclusions can be drawn regarding the tasks involved. The complex nature of legal texts demands a specialized sentence segmenter such as LUIMA which outperforms vanilla SpaCy even with some exceptions added. However, it was evident that even this segmenter could be improved upon to perfectly identify boundaries. Word embeddings are quite beneficial in capturing the representation of vocabulary in the corpus which was done by training on unlabeled texts. While it might have been a better idea to train the embedding model for more epochs given the small dataset, the nearest neighbors showed the model captured the semantic relationship to a justifiable degree. Manual exceptions had to be added for tokenization as well, which left room for error and proves how important domain knowledge is.

For classification, Linear SVC performed best with TFIDF featurization and Radial SVM best with word embedding featurization. Comparing the two, Radial SVM seemed to be the winner by a small margin.
As seen from the literature survey, the type system design is mostly problem specific, so a system that satisfyingly covers grammatical patterns, especially interleaving of types, seems optimal. While annotation is tough, Westermann et al [3]. suggested lateral annotations which can also be used. Sentence embeddings can also be used to take things a step further, allowing better capturing of context. While embedding model performed slightly better, more features can be added such as cue phrases, combination of TFIDF, POS tags, etc. As noted by Hachey et al [1]., SVMs had the best performance which is also the case in this project. Deep learning approaches e.g., by Bhattacharya et al [2]., have also proven to perform really well compared to baselines and they required no hand-crafted features. However, that requires sufficient amount of data with good support for all types which was not the case for this project and thus was not explored. It is still recommended provided a large dataset is managed.

The course gave valuable insight to how domain specific Natural Language Processing can be challenging and interesting. Previous knowledge of NLP should be recommended so more time can be given on workshops on different type of legal texts and understanding the domain better.

9. Code Instructions

The folder contains the embedding model, classifier and luima files needed to run the **Setup_and_Analyze** notebook which contains the required **analyze** function. The notebook has three cells to be run sequentially, with one **cell** covering the task of the setup function described. The analyze function takes a string and returns result as instructed. The other notebook includes all the instructed tasks and cleaned code with the outputs cleared. It can be run if the required files such as unlabeled files are included in the correct directories. A requirements file is also included.

References

1. Hachey, B., Grover, C.: *Sentence classification experiments for legal text summarisation*. In: Proceedings of the 17th Annual Conference on Legal Knowledge and Information Systems (Jurix). (2004)
2. P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, and A. Wyner, *"Identification of rhetorical roles of sentences in indian legal judgments,"* in Proc. International Conference on Legal Knowledge and Information Systems (JURIX). (2019)
3. Hannes Westermann, Jaromír Šavelka, Vern R Walker, Kevin D Ashley, and Karim Benyekhlef. *Sentence Embeddings and High-Speed Similarity Search for Fast Computer Assisted Annotation of Legal Documents*. In JURIX 2020, Vol. 334. IOS Press, 164. (2020)