DOCUMENTATION

The goal is to classify given legal documents into 41 pre-defined classes. We have 999 documents in total, of which 899 of them have been labeled into their respective classes and the remaining 100 documents have to be labeled.

Data Reading

The first task would be to iterate through the Fixed Judgements folder and read the files in them. In doing so, they will be appended into a new data frame called df1, where the first column will hold the name of the text document under the column header ‘Judgements’ and the data in the file under the column header ‘keywords’. The initial data frame with the first 5 rows looks as follows:

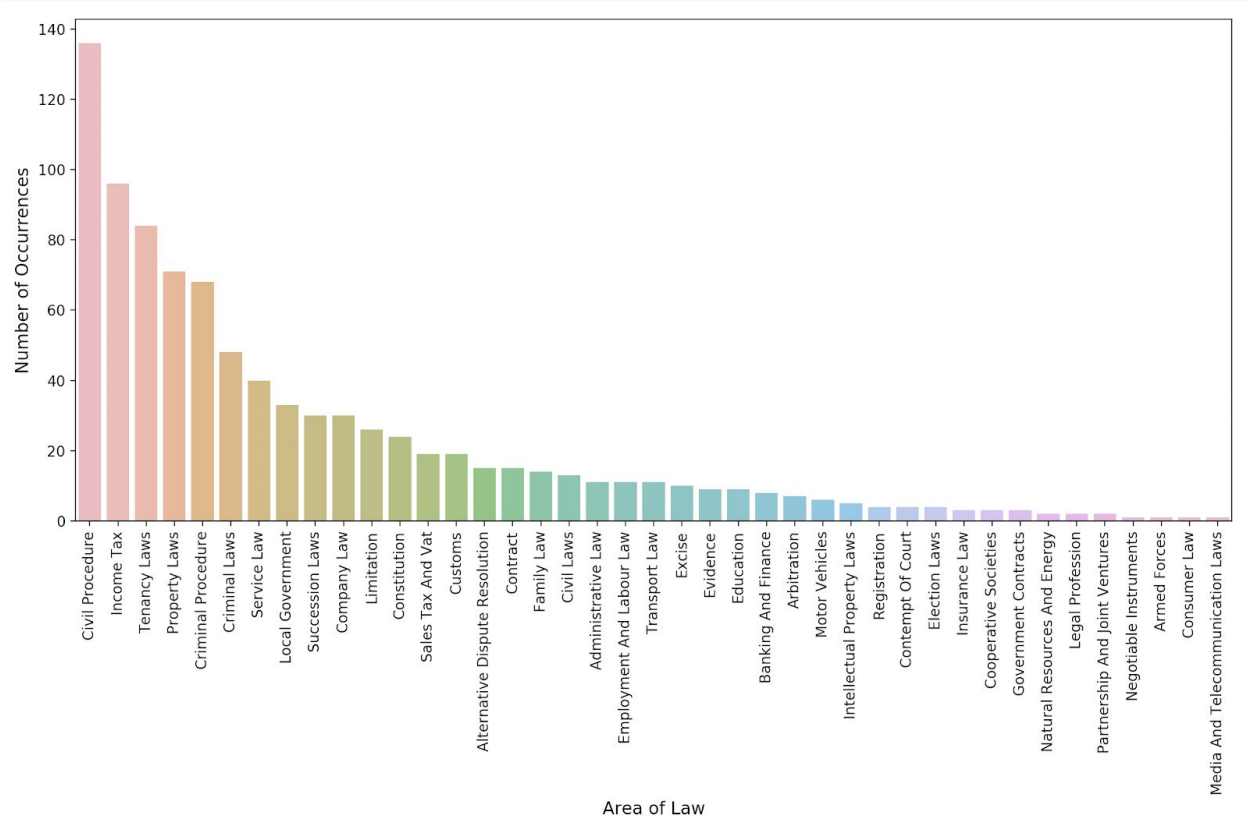
	Judgements	keywords
0	LNIND_1951_CAL_50	b'Parties\nAmulya Chandra Roy Versus Pashupati...
1	LNIND_1953_CAL_180	b"Parties\nMohunt Satyanarayan Giri Versus Nan...
2	LNIND_1951_CAL_93	b'\nParties\nNalinakhya Bysack Versus Shyam Su...
3	LNIND_1993_DEL_165	b'Parties\nM. L. Khurana Versus H. C. Chopra\n...
4	LNIND_1976_CAL_149	b'Parties\nPatai Sheikh Versus State of West B...

The next thing to do would be to map the generated data frame with its labels, which has been documented in the given Interview_Mapping.csv file. This file has been merged with the existing data frame on the column ‘Judgements’, and the column name ‘Area.of.Law’ which holds the labels of the documents has been renamed into column name ‘LawType’. The resulting data frame with the first 5 rows looks as below:

	Judgements	keywords	LawType
0	LNIND_1951_CAL_50	b'Parties\nAmulya Chandra Roy Versus Pashupati...	Tenancy Laws
1	LNIND_1953_CAL_180	b"Parties\nMohunt Satyanarayan Giri Versus Nan...	Contract
2	LNIND_1951_CAL_93	b'\nParties\nNalinakhya Bysack Versus Shyam Su...	Civil Procedure
3	LNIND_1993_DEL_165	b'Parties\nM. L. Khurana Versus H. C. Chopra\n...	Tenancy Laws
4	LNIND_1976_CAL_149	b'Parties\nPatai Sheikh Versus State of West B...	To be Tested

Data Exploration

Now, we would like to see the distribution of the dataset over the labels. This can be done by plotting the counts of each of the labels against the label names. The resulting bar plot looks as follows:



Data Preprocessing

Now, the 'keywords' column has to be pre-processed by converting text to lowercase and stripping punctuation and symbols from words. If we compare it with the previous console output pasted above, we can see that the column keywords has been modified. The resulting data frame is shown below:

	Judgements	keywords	LawType
0	LNIND_1951_CAL_50	parties\namulya chandra roy versus pashupati n...	Tenancy Laws
1	LNIND_1953_CAL_180	parties\nmohunt satyanarayan giri versus nanda...	Contract
2	LNIND_1951_CAL_93	parties\nnalinakhya bysack versus shyam sundar...	Civil Procedure
3	LNIND_1993_DEL_165	parties\nm. l. khurana versus h. c. chopra\nhi...	Tenancy Laws
4	LNIND_1976_CAL_149	parties\npatai sheikh versus state of west ben...	To be Tested

As we can see, all the words have been converted to lowercase and have been stripped of symbols.

The next step is to tokenize the ‘keywords’ column using the NLTK tokenizer. This would tag every keywords attribute with its Area of Law. An example of the keywords tagged as Criminal Law is pasted below. This will be our training feature.

```
TaggedDocument(['parties', 'mohammad', 'molla', 'versus', 'state', 'of', 'west', 'bengal', 'high', 'court', 'of', 'judicature', 'at', 'calcutta', 'judges', 'the', 'honourable', 'mr.', 'justice', 'g.n', 'ray', 'appeal', 'no', '--', '--', '--', '--', 'doj', '03.06.1980', 'advocates', 'appeared', 'for', 'the', 'appearing', 'parties', 'u.p', 'mukherji', 'm.p', 'mukherji', 'advocates', 'judgment', 'in', 'this', 'rule', 'memo', 'dated', '18th', 'august', '1978', 'issued', 'by', 'the', 'junior', 'land', 'reforms', 'officer', 'jaynagar', 'ii', '24-parganas', 'to', 'the', 'officer-in-charge', 'of', 'jaynagar', 'police', 'station', 'is', 'under', 'challenge', 'it', 'appears', 'that', 'the', 'respondent', 'no.5', 'rousan', 'ali', 'mondal', 'made', 'complaint', 'before', 'the', 'junior', 'land', 'reforms', 'officer', 'that', 'the', 'petitioner', 'mohammad', 'molla', 'was', 'creating', 'disturbances', 'in', 'the', 'peaceful', 'cultivation', 'of', 'the', 'said', 'land', 'and/or', 'was', 'attempting', 'to', 'trespass', 'upon', 'the', 'said', 'land', 'on', 'the', 'basis', 'of', 'such', 'complaint', 'made', 'to', 'the', 'junior', 'land', 'reforms', 'officer', 'the', 'junior', 'land', 'reforms', 'officer', 'caused', 'an', 'enquiry', 'and', 'it', 'appears', 'from', 'the', 'impugned', 'order', 'that', 'decision', 'of', 'the', 'block', 'level', 'land', 'reforms', 'advisory', 'committee', 'dated', '16th', 'august', '1978', 'was', 'also', 'made', 'available', 'to', 'the', 'said', 'junior', 'land', 'reforms', 'officer', 'and', 'on', 'that', 'basis', 'he', 'requested', 'the', 'officer-in-charge', 'jaynagar', 'police', 'station', 'to', 'give', 'necessary', 'police', 'protection', 'to', 'the', 'respondent', 'no', 'rousan', 'ali', 'mondal', 'for', 'his', 'peaceful', 'cultivation', 'of', 'the', 'disputed', 'land', 'mr.', 'mukherji', 'learned', 'counsel', 'for', 'the', 'petitioner', 'contend', 'that', 'simply', 'because', 'an', 'allegation', 'was', 'made', 'before', 'the', 'junior', 'land', 'reforms', 'officer', 'that', 'somebody', 'was', 'trespassing', 'upon', 'the', 'land', 'of', 'the', 'said', 'complainant', 'the', 'junior', 'land', 'reforms', 'officer', 'had', 'no', 'business', 'to', 'request', 'the', 'police', 'authorities', 'to', 'give', 'protection', 'to', 'the', 'party', 'concerned', 'and', 'such', 'eercise', 'of', 'power', 'is', 'not', 'authorized', 'under', 'any', 'of', 'the', 'provision', 'of', 'the', 'land', 'reforms', 'act', 'mr.', 'mukherji', 'has', 'pointed', 'out', 'that', 'if', 'dispute', 'as', 'to', 'barge', 'cultivation', 'is', 'raised', 'the', 'bhagchas', 'officer', 'is', 'required', 'to', 'decide', 'the', 'said', 'dispute', 'under', 'the', 'provisions', 'of', 's.', '18', 'of', 'the', 'land', 'reforms', 'act', 'mr.', 'mukherji', 'has', 'also', 'pointed', 'out', 'that', 'for', 'the', 'alleged', 'criminal', 'action', 'namely', 'trespass', 'and/or', 'interference', 'with', 'peaceful', 'possession', 'of', 'the', 'complainant', 'there', 'are', 'other', 'provisions', 'of', 'law', 'and', 'there', 'are', 'different', 'statutory', 'authorities', 'who', 'can', 'take', 'appropriate', 'action', 'against', 'such', 'wrong-doer', 'the', 'junior', 'land', 'reforms', 'officer', 'has', 'overstepped', 'his', 'limit', 'in', 'this', 'case', 'and', 'has', 'purported', 'to', 'assume', 'jurisdiction', 'not', 'warranted', 'under', 'the', 'provisions', 'of', 'land', 'reforms', 'act', 'the', 'learned', 'counsel', 'for', 'the', 'state', 'however', 'has', 'produced', 'report', 'from', 'which', 'it', 'appears', 'that', 'the', 'junior', 'land', 'reforms', 'officer', 'caused', 'an', 'enquiry', 'and', 'on', 'the', 'basis', 'of', 'the', 'enquiry', 'the', 'said', 'memo', 'was', 'issued', 'by', 'him', 'it', 'is', 'immaterial', 'whether', 'the', 'junior', 'land', 'reforms', 'officer', 'himself', 'caused', 'an', 'enquiry', 'or', 'he', 'got', 'report', 'from', 'other', 'sources', 'the', 'point', 'for', 'determination', 'in', 'this', 'rule', 'is', 'whether', 'on', 'the', 'basis', 'of', 'complaint', 'of', 'trespass', 'or', 'other', 'criminal', 'activities', 'the', 'junior', 'land', 'reforms', 'officer', 'had', 'any', 'authority', 'to', 'request', 'the', 'officer-in-charge', 'of', 'the', 'police', 'station', 'to', 'give', 'protection', 'to', 'particular', 'party', 'making', 'complaint', 'before', 'him', 'in', 'my', 'view', 'mr.', 'mukherji', 'is', 'correct', 'in', 'his', 'submission', 'that', 'the', 'junior', 'land', 'reforms', 'officer', 'was', 'not', 'clothed', 'with', 'any', 'such', 'power', 'under', 'the', 'provisions', 'of', 'the', 'land', 'reforms', 'act', 'he', 'clearly', 'overstepped', 'his', 'limit', 'by', 'sending', 'the', 'request', 'to', 'the', 'police', 'officer', 'on', 'the', 'basis', 'of', 'complaint', 'made', 'by', 'the', 'respondent', 'no.5', 'on', 'that', 'score', 'alone', 'the', 'rule', 'must', 'succeed', 'the', 'impugned', 'memo', 'is', 'therefore', 'quashed', 'the', 'rule', 'is', 'made', 'absolute', 'there', 'will', 'be', 'no', 'order', 'as', 'costs', 'however', 'make', 'it', 'clear', 'that', 'this', 'court', 'has', 'not', 'epressed', 'any', 'opinion', 'as', 'to', 'whether', 'the', 'petitioner', 'is', 'bargadar', 'or', 'not', 'and', 'if', 'proper', 'petition', 'is', 'presented', 'by', 'the', 'petitioner', 'claiming', 'his', 'right', 'as', 'bargadar', 'before', 'the', 'appropriate', 'forum', 'such', 'appropriate', 'forum', 'will', 'obviously', 'decide', 'the', 'dispute', 'raised', 'by', 'the', 'petitioner', 'as', 'to', 'barga', 'cultivation', 'it', 'is', 'also', 'made', 'clear', 'that', 'as', 'it', 'is', 'not', 'necessary', 'for', 'this', 'court', 'to', 'decide', 'whether', 'the', 'petitioner', 'is', 'bargadar', 'or', 'not', 'for', 'the', 'purpose', 'of', 'disposal', 'of', 'this', 'rule', 'this', 'court', 'has', 'not', 'decided', 'the', 'other', 'contentions', 'raised', 'by', 'the', 'parties', 'rule', 'discharged'], ['Criminal Laws'])
```

Set up Doc2Vec Training and Evaluation Models

In Doc2Vec, the 2 algorithms are Distributed Memory (DM) and Distributed Bag of Words (DBOW). The DBOW is analogous to the Word2Vec’s Skip-Gram model. The paragraph vectors are obtained by training a neural network on the task of predicting a probability distribution of words in a paragraph given a randomly-sampled word from the paragraph.

The parameters set for the model are:

1. dm = 0 : to set to DBOW model (dm = 1 for DM model).
2. vector_size = 300: 300 dimensional features.
3. min_count = 2: removing all words below frequency 2.
4. negative = 5: specifies how many ‘noise words’ are to be drawn.
5. hs = 0: negative sampling will be used.
6. workers = cores: use cores number of threads for the model.

Using these parameters to tune the model, we build the vocabulary for this model. Training a Doc2Vec is done by the Gensim package, and we train for 30 epochs. We then build the final vector features for the classifier and using these feature vectors, we train the logistic regression classifier.

Now, we do the same thing for the Distributed Memory model, and train for 30 epochs. Again we build the vector features for the logistic regression classifier. DM acts as a memory that remembers what is missing from the current context - or as the topic of the paragraph. While the

word vectors represent the concept of a word, the document vector intends to represent the concept of a document.

Model Pairing

According to Gensim Doc2Vec tutorial, combining a paragraph vector from DBOW and DM improves performance. Hence we will pair the above models together for evaluation.

First, we delete the temporary training data to free up the RAM. Then we concatenate the two models and build feature vectors. Finally, we train the Logistic Regression model using these feature vectors.

Results

The first 100 rows of the Interview_Mapping.csv file have been updated according to the predictions generated from the above model. They can be viewed in the Mitta_Sneha_results.csv file. The script for running this model to obtain results can be accessed by using the Mitta_Sneha_script.py file.