**Text Analytics on Law Cases**

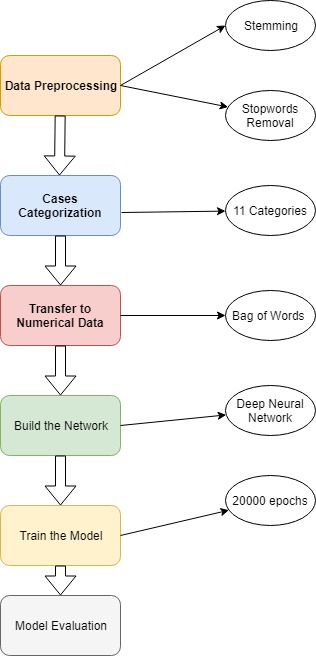
Minghan Wang

06.25.2018

**Introduction:**

The documentation of each law cases could be great amount of text data. If one is able to put them into good use, the data could provide great amount of information. In each law case, the result of it, whether it’s reversed or affirmed, is always related to the content of the case. The whole research consists of several text analytics methods, word count, clustering and classification. The goal of this project is to find out the characteristics of affirmed and reverse law cases and use machine learning method to predict the result of cases beforehand.

**Overall Methodology:**



**Data Preprocessing:**

**Stop-words Removal and Stemming**

The raw data was first tokenized, each case was tokenized into a list of words. And then the stop-words were removed while also stemmed.



**Figure 1** Result of Case 1 being Stemmed and Stop-words Removed(Output too large, only part of the result was put)

1. **Word Count**:

To get a better understanding on the text data, NLTK word frequency methods were used to find out the most commonly appeared words for law cases that are affirmed and reversed.

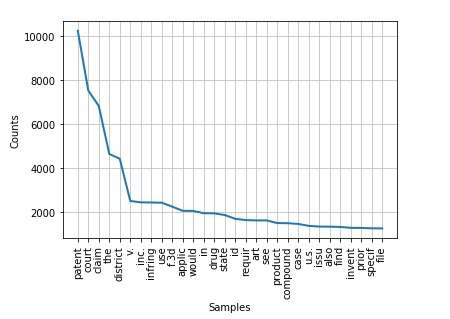
|  |  |
| --- | --- |
| **Word** | **Frequency** |
| Patent | 10256 |
| Court | 7529 |
| claim | 6823 |
| District | 4418 |
| drug | 1926 |
| state | 1850 |
| use | 2413 |

**Table 1 Commonly appeared Words for Affirmed Cases**

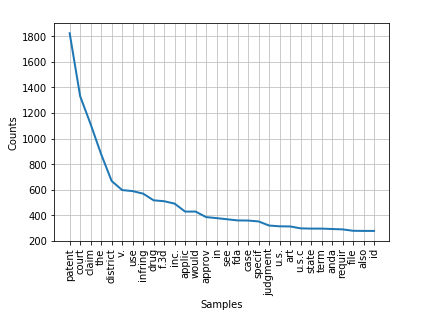
|  |  |
| --- | --- |
| **Word** | **Frequency** |
| Patent | 1823 |
| Claim | 1112 |
| Approv | 1003 |
| Process | 997 |
| Obvious | 984 |
| Infring | 570 |
| Drug | 518 |

**Table 2 Commonly appeared Words for Reversed Cases**

For both affirmed and reversed cases, the word ‘patent’ has the highest number of appearance while affirmed cases have other popular words like ‘court’, ‘state’ and ‘use’ and reversed cases have words such as ‘approv’, ‘infring’ and ‘obvious’.



**Figure 1.1 Line Chart of Word Frequency of Affirmed Cases**



**Figure 1.2 Line Chart of Word Frequency of Affirmed Cases**

1. **Clustering**

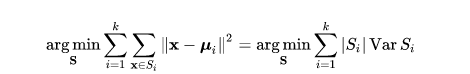
After conducting word frequency analysis, clustering was used on the data set for further understanding.

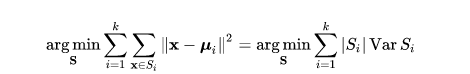
* 1. **K-means**

*k*-means clustering is a method of [vector quantization](https://en.wikipedia.org/wiki/Vector_quantization), originally from [signal processing](https://en.wikipedia.org/wiki/Signal_processing), that is popular for [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) in [data mining](https://en.wikipedia.org/wiki/Data_mining). *k*-means clustering aims to [partition](https://en.wikipedia.org/wiki/Partition_of_a_set) *n* observations into *k* clusters in which each observation belongs to the [cluster](https://en.wikipedia.org/wiki/Cluster_(statistics)) with the nearest [mean](https://en.wikipedia.org/wiki/Mean), serving as a prototype of the cluster. This results in a partitioning of the data space into [Voronoi cells](https://en.wikipedia.org/wiki/Voronoi_cell).

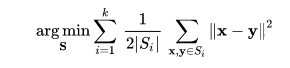
The problem is computationally difficult ([NP-hard](https://en.wikipedia.org/wiki/NP-hardness)); however, there are efficient [heuristic algorithms](https://en.wikipedia.org/wiki/Heuristic_algorithm) that are commonly employed and converge quickly to a [local optimum](https://en.wikipedia.org/wiki/Local_optimum). These are usually similar to the [expectation-maximization algorithm](https://en.wikipedia.org/wiki/Expectation-maximization_algorithm) for [mixtures](https://en.wikipedia.org/wiki/Mixture_model) of [Gaussian distributions](https://en.wikipedia.org/wiki/Gaussian_distribution) via an iterative refinement approach employed by both *k-means* and *Gaussian mixture modeling*. Additionally, they both use cluster centers to model the data; however, *k*-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

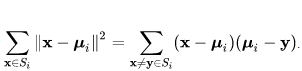
The algorithm has a loose relationship to the [*k*-nearest neighbor classifier](https://en.wikipedia.org/wiki/K-nearest_neighbor), a popular [machine learning](https://en.wikipedia.org/wiki/Machine_learning) technique for classification that is often confused with *k*-means due to the *k* in the name. One can apply the 1-nearest neighbor classifier on the cluster centers obtained by *k*-means to classify new data into the existing clusters. This is known as [nearest centroid classifier](https://en.wikipedia.org/wiki/Nearest_centroid_classifier) or [Rocchio algorithm](https://en.wikipedia.org/wiki/Rocchio_algorithm" \o ").

Given a set of observations (**x**1, **x**2, …, **x***n*), where each observation is a *d*-dimensional real vector, *k*-means clustering aims to partition the *n* observations into *k* (≤ *n*) sets **S** = {*S*1, *S*2, …, *Sk*} so as to minimize the within-cluster sum of squares (WCSS) (i.e. [variance](https://en.wikipedia.org/wiki/Variance)). Formally, the objective is to find:{\displaystyle {\underset {\mathbf {S} }{\operatorname {arg\,min} }}\sum \_{i=1}^{k}\sum \_{\mathbf {x} \in S\_{i}}\left\|\mathbf {x} –{\boldsymbol {\mu }}\_{i}\right\|^{2}={\underset {\mathbf {S} }{\operatorname {arg\,min} }}\sum \_{i=1}^{k}|S\_{i}|\operatorname {Var} S\_{i}}



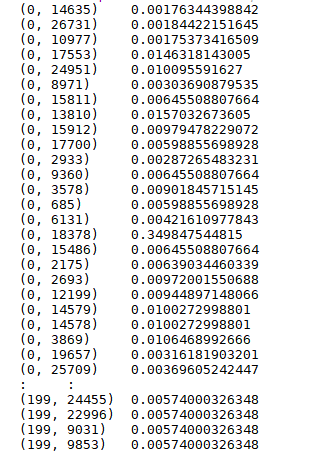
where ***μ****i* is the mean of points in *Si*. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

{\displaystyle {\underset {\mathbf {S} }{\operatorname {arg\,min} }}\sum \_{i=1}^{k}\,{\frac {1}{2|S\_{i}|}}\,\sum \_{\mathbf {x} ,\mathbf {y} \in S\_{i}}\left\|\mathbf {x} -\mathbf {y} \right\|^{2}}

The equivalence can be deduced from identity  {\displaystyle \sum \_{\mathbf {x} \in S\_{i}}\left\|\mathbf {x} -{\boldsymbol {\mu }}\_{i}\right\|^{2}=\sum \_{\mathbf {x} \neq \mathbf {y} \in S\_{i}}(\mathbf {x} -{\boldsymbol {\mu }}\_{i})({\boldsymbol {\mu }}\_{i}-\mathbf {y} )}. Because the total variance is constant, this is also equivalent to maximizing the sum of squared deviations between points in *different* clusters (between-cluster sum of squares, BCSS),[[1]](https://en.wikipedia.org/wiki/K-means_clustering#cite_note-:1-1) which follows easily from the [law of total variance](https://en.wikipedia.org/wiki/Law_of_total_variance).

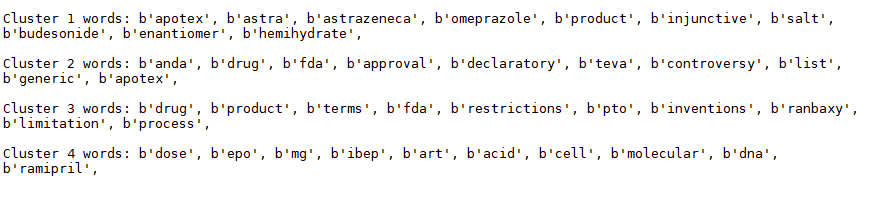
* 1. **TI-IDF Matrix**

First we generate the TF-IDF matrix. By definition, tf-idf, short for term frequency-inverse document frequency is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.



* 1. **K-means**

Figure 2.1 Shows the result of K-means clustering while the number of clusters is 4. The model groups all the law cases into 4 different clusters and output the most commonly appeared words in each cluster.

****

**Figure 2.1**

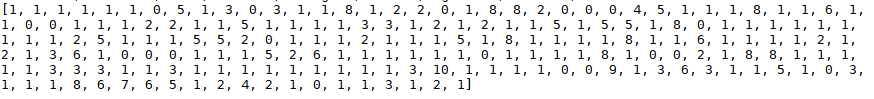
**3. Classification Using Deep Neural Network**

**3.1 Case Categorization:**

Each law case document has recorded the results of that case. There are in totally 11 outcomes for the given cases.

|  |  |
| --- | --- |
| Case Result | Category Number |
| Reversed | 0 |
| Affirmed | 1 |
| Affirmed in part, reversed in part, and remanded. | 2 |
| Reversed and remanded | 3 |
| Affirmed in part, reversed in part, and vacated. | 4 |
| Vacated and remanded | 5 |
| Affirmed in part and reversed in part. | 6 |
| Appeal dismissed; Affirmed | 7 |
| Affirmed in part, vacated in part, and remanded. | 8 |
| Affirmed in part and modified in part. | 9 |
| Motion denied | 10 |

Above table shows the results of different cases provided and the corresponding category number that was used in the project. For example cases that were affirmed were categorized into group 1 while cases that were reversed were in group 0.

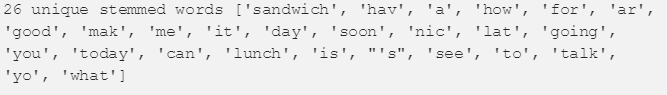


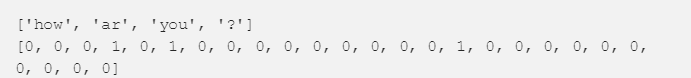
**Figure 3.1**

Figure 3.1 shows an array of length 200 which equals to the 200 cases provided. Each number in the array represents the results of the corresponding case. First index is 1 shows that case 1 is an affirmed case while case 8 is vacated and remanded as the eighth index is 5.

**3.2 Bag of Words Algorithm(Transfer Text Data to Numerical Data):**

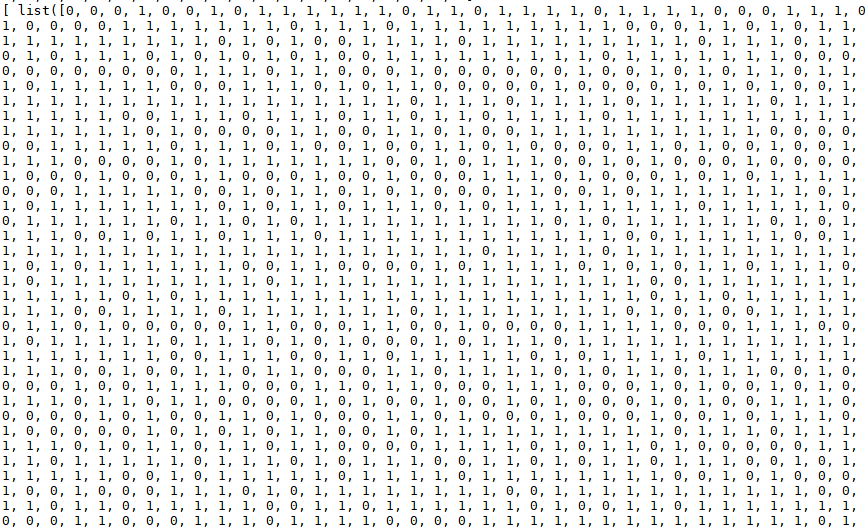
Because neural network model only takes in numerical data, the text data must be transferred into numerical data using some algorithms. The algorithm used in this project is the bag of words approach. The bag of word approach is a simple algorithm that transfers the text data into a binary array which only contains 0 and 1. The approach is to create an array contains the unique words in all the 200 law cases which in other words, a bag of words. And then for each case, create another array contains of 0 and 1s, where the indices are 1 when the corresponding words are in the bag of words and otherwise 0.





**Figure 3.2**

Figure 3.2 shows a simple example of how bag of words approach works. For example, the unique words in all documents are shown as above with size 26. Then “How are you ?” would be transferred into the above shown array. The word “how” are in the forth index in the first array, so the forth index in the second array is 1.

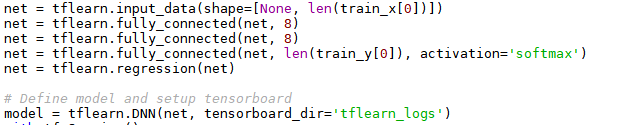




**Figure 3.3**

Figure 3.3 above shows an example of a binary representation of the case 1 using the bag of word approach. The second array shown in the figure with a length of 11 is the binary representation of the category this case is in. As show, case 1 is in group 1 which is affirmed.

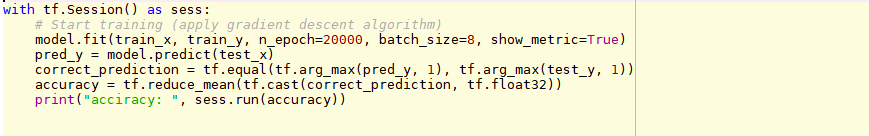
**3.3 Building the Neural Net:**



The network is built using the tensorflow and tflearn package. The neural net contains three fully connected layers where fully connected means each neurons of the current layer is connected to all the neurons in the previous layer. The activation function used was ‘softmax’ which would convert the numerical data into probabilities. And the output layer is set as a regression which will perform a regression to the input.

**3.4 Training the Neural Net:**

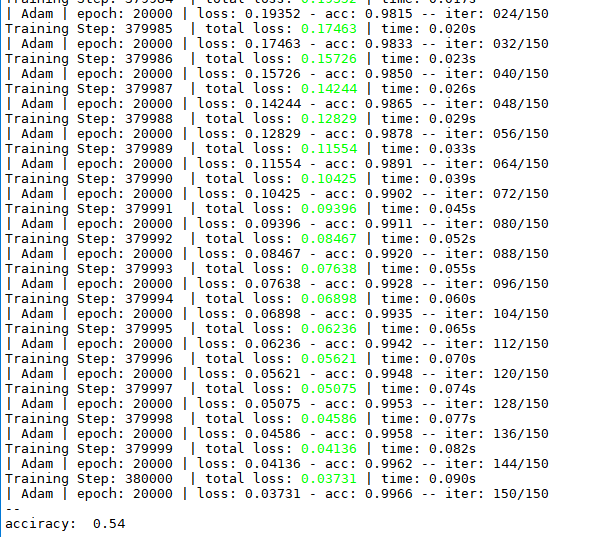
Before training the model, the processed numerical dataset is splitted into two set, a training set with 150 cases which were passed into the model as input and a testing set with 50 cases which were used to evaluate the accuracy of the model.



**Figure 3.4**

The model was trained on 20000 epochs. The number of epoch to be trained is adjustable.

**3.5 Model Result:**



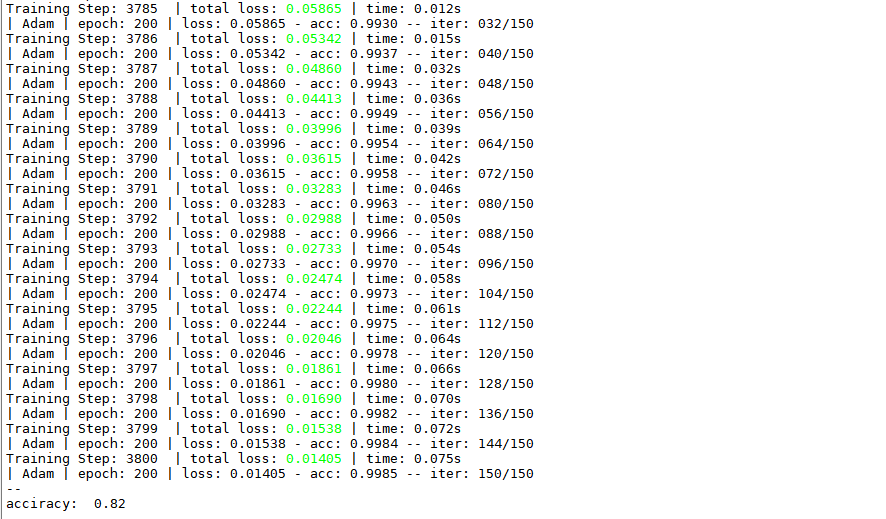
**Figure 3.5**

The accuracy of the neural network is 0.54. It is much higher than randomly guessing from the 11 outcomes which only has a 1/11 probability of getting right.

**3.6 Reducing Number of Outcomes to 4:**

|  |  |
| --- | --- |
| Case Result | Category Number |
| Reversed | 0 |
| Affirmed | 1 |
| Vacated and remanded | 2 |
| Motion denied | 3 |

In order to improve the model accuracy, the number of outcomes was reduced from 11 to 4. And the accuracy has improved to 82% for the neural network.



**References:**

1. Bag-of-words model, <https://en.wikipedia.org/wiki/Bag-of-words_model>
2. Text Classification using Neural Networks, <https://machinelearnings.co/text-classification-using-neural-networks-f5cd7b8765c6>
3. K-means Clustering, https://en.wikipedia.org/wiki/K-means\_clustering