6/16/2019

# Clause Search

Legal Document clause search

Aggarwal, Rohit
IIT BOMBAY

# Problem Statement

Finding the clauses that tells us power exercised by a Director of a company from various legal documents. After that summarize it to a smaller report for the user.

# Final Approach

The problem is one of unsupervised learning. Following are the steps followed by me:

1. Converting text to image
2. Converting image to text
3. Splitting text into clauses
4. Separating clauses containing Director word
5. Performing clustering on the clauses
6. Separating the clauses with different labels
7. Summarizing the clauses

# Problems Faced and Solution

**P: Problem** **S: Solution**

**P:** First problem was reading text from pdf files as every pdf file has different encoding and some of them were not even pdf but scanned documents of actual legal documents. These types of documents can't be read through conventional methods

**S:** Instead of using traditional methods I used OCR (Optical Character Recognition) for the problem faced. In this method every page in the pdf is converted to image and then OCR detects the text which can be returned in a text file. Then this text file can be read and parsed according to the need

**P:** Finding a method to cluster the unstructured data and find meaningful insights

**S:** Nothing is known about the documents. They are unstructured and no label is present with them. Thus, this problem statement belongs to unsupervised learning category. For this purpose I first segmented the text into clauses and then separated the clauses containing the word 'Director'. Then among various clustering algorithms present chose 'K-means' clustering to separate the clauses. After that analyze the texts returned according to the label tagged with it.

**P:** Summarizing the result also poses a problem

**S:** Currently used in-build module genism summarization which uses TextRank for the process. There are various other methods available which follows different algorithms. Though, Most of these algorithms uses TextRank.

# Software Required

Following are the external libraries used and software installed:

- Tesseract-ocr (to be installed separately and path needs to be set)
- pyTesseract (can be pip installed)
- Poppler (to be installed separately and path needs to be set)
- Pdf2image(can be pip installed)
- Pillow(can be pip installed)
- Scikit-Learn(can be pip installed)
- gensim_sum_ext(can be pip installed)

# Future Enhancements

Many enhancements are possible but needs time for implementation and testing

- Finding better clustering method which maybe more suitable for this purpose
- Feeding these text to Neural Networks and performing clustering in the middle layers for better performance
- Parallelizing the process to improve the speed
- Optimizing the algorithm.
- Various other NLP libraries are available like 'lexnlp'/nltk which are used mostly for analysis on more fine level like words/sentences.
- Better summarization of the result can be done using RNN (Recurrent Neural Network).g

# Important Links:

- https://www.geeksforgeeks.org/python-reading-contents-of-pdf-using-ocr-optical-character-recognition/
- https://www.nltk.org/book/
- https://medium.com/@MSalnikov/text-clustering-with-k-means-and-tf-idf-f099bcf95183
- https://contraxsuite.com/lexnlp-features/
- https://www.bogotobogo.com/python/NLTK/tf_idf_with_scikit-learn_NLTK.php