

Legal Text Classification using NLP

Submitted in partial fulfillment of the requirements
of the degree of

T. E. Computer Engineering

By

Jini Vedikattil	Roll No. 29	PID 182123
Vishakha Mistry	Roll No. 32	PID 182072
Semil Shah	Roll No. 35	PID 192262

Guide:

Mrs. Varsha Shrivastava
Assistant Professor



Department of Computer Engineering
St. Francis Institute of Technology
(Engineering College)

University of Mumbai
2020-2021

CERTIFICATE

This is to certify that the project entitled **“Legal Text Classification using NLP”** is a bonafide work of **“Jini Vedikattil” (Roll No. 29)**, **“Vishakha Mistry” (Roll No. 32) & “Semil Shah” (Roll No.: 35)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of T.E. in Computer Engineering

(Name and sign)

Mrs. Varsha Shrivastava
Guide

(Name and sign)

Mrs. Kavita Sonawane
Head of Department

Project Report Approval for T.E.

This project report entitled *Legal text Classification using NLP* by *Jini Vedikattil, Vishakha Mistry & Semil Shah* is approved for the degree of *T.E. in Computer Engineering*.

Examiners

1.-----

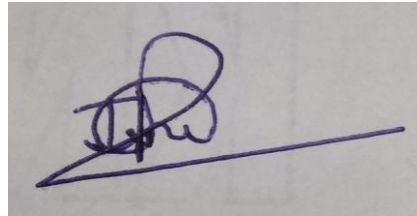
2.-----

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



(Signature)

Jini Veditakki

Roll No. 29

Date:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

A handwritten signature in blue ink, appearing to read 'Vishakha Mistry', is shown on a white background.

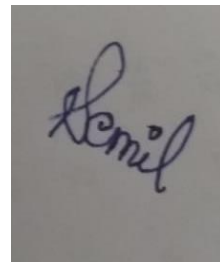
(Signature)

Vishakha Mistry Roll No. 32

Date:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



(Signature)

Semil Shah Roll No. 35

Date:

Abstract

Legal documents consist of long and difficult words and thus categorizing it into different legal domain becomes difficult. Understanding of such meticulous documents require expert knowledge and time. A legal expert would need to read hundreds or thousands of documents in order to place opinions into subject categories, whereas an automatic system could do this with little or no human effort. There are tons of old legal data, case rulings in various jurisdictions which are in the records that should be referred before a new verdict is passed. Thus, there is a need for a system that could be used to catalogue the data properly which could be used for future reference. Our goal is to determine the best method for applying automated document classification to legal texts using NLP techniques with the hopes of facilitating legal experts in their classification of court documents.

Keywords: Legal domain, Natural Language Processing, Text Classification, Machine learning

Contents

Chapter		Contents	Page No.
1		INTRODUCTION	1
	1.1	Description	1
	1.2	Problem Formulation	1
	1.3	Motivation	2
	1.4	Proposed Solution	2
	1.5	Scope of the project	3
2		REVIEW OF LITERATURE	4
3		SYSTEM ANALYSIS	6
	3.1	Functional Requirements	6
	3.2	Non Functional Requirements	6
	3.3	Performance Requirements	7
	3.4	Specific Requirements	8
	3.5	Use-Case Diagrams and description	9
4		ANALYSIS MODELING	10
	4.1	Functional Modeling-DFD	10
5		DESIGN	11
	5.1	Architectural Design	11
	5.2	User Interface Design	12
6		IMPLEMENTATION	13
	6.1	Algorithms / Methods Used	13
	6.2	Working of the project	14
7		CONCLUSIONS	18
8		REFERENCES	19
9		ACKNOWLEDGEMENTS	20

List of Figures

Fig. No.	Figure Caption	Page No.
3.5.1	Use case Diagram	9
4.1.1	DFD	10
5.1.1	Work Flow Diagram	11
5.2.1	User Interface	12
6.1.1	K Means clustering	13
6.2.1	SSE plot a range of clusters	15
6.2.2	Data Visualization	16
6.2.3	Clusters with top keywords	17

List of Abbreviations

Sr. No.	Abbreviation	Expanded form
1	NLP	Natural Language Processing
2	TFIDF	Term Frequency-Inverse Document Frequency
3	PCA	Principal Component Analysis
4	TSNE	t-Distributed Stochastic Neighbor Embedding
5	SSE	Sum of Squared Errors
6	CNN	Convolution Neural Networks
7	DFD	Data Flow Diagram

Chapter 1

Introduction

Every legal case falls into one or more areas of law (“legal areas”). These areas are lawyers’ shorthand for the subset of legal principles and rules governing the case. Thus lawyers often triage a new case by asking if it falls within tort, contract, or other legal areas. Answering this allows unresolved cases to be funneled to the right experts and for resolved precedents to be efficiently retrieved. Legal database providers routinely provide area-based search functionality; courts often publish judgments labelled by legal area. The law therefore yields pockets of expert labelled text.

1. Description

This project explores the feasibility of utilizing NLP techniques for the classification of legal opinions. Legal opinions and texts consist of long running sentences (or long spans of text) that do not conform to standard English linguistic or grammar patterns. NLP techniques and algorithms developed for processing standard linguistic sentences and documents often produce incorrect results while processing legal documents. Precedent is an important guiding principle in the legal domain. This work is part of our larger effort to automate analysis of legal opinions, which holds much promise towards developing systems that provide equal access to law.

2. Problem Formulation

Legal areas generally refer to a subset of related legal principles and rules governing certain dispute types. There is no universal set of legal areas. The reasoning behind particular legal opinions and rulings are often cited in other legal cases to further the legal theory put forward by attorneys. The bedrock of the legal system are the precedents upon which disposition of legal disputes are based. Every legal opinion and ruling needs to be classified and cataloged so that they can be accessed during searches. Current NLP based approaches towards classification of legal documents are often guided by the annotations by human experts. Our goal is to determine the best method for

applying automated document classification to legal texts with the hopes of facilitating legal experts in their classification of court documents.

3. Motivation

Classification of legal opinions and rulings is a very time consuming and tedious manual task. Human experts create predefined categories. Societal change may create new areas of law like data protection. Cases may fall into more than one legal area but never none. Legal text consists of long contiguous sentences which are difficult to understand, let alone categorize them into a particular area. Understanding of such meticulous documents require expert knowledge and time. Due to limitation in such resources, it is not feasible to use such important resources for frivolous tasks. A legal expert would need to read hundreds or thousands of documents in order to place opinions into subject categories, whereas an automatic system could do this with little or no human effort. There are tons of old legal data, case rulings in various jurisdictions which are in the records that should be referred before a new verdict is passed. Thus, there is a need for a system that could be used to catalogue the data properly which could be used for future reference. Also, it would become easier for a layman to understand the actual meaning of legal text without the need of an expert consultation.

4. Proposed Solution

A system that classifies legal texts by area would be useful for enriching older, typically unlabelled judgments with metadata for more efficient search and retrieval. The system can also suggest areas for further inquiry by predicting which areas a new text falls within. Despite its potential, this problem, which we refer to and later define as “legal area classification”, remains relatively unexplored. One explanation is the relative scarcity of labelled documents in the law (typically in the low thousands), at least by deep learning standards. Legal texts are typically longer than the customer reviews, tweets, and other documents typical in NLP research. Against this backdrop, this paper uses a novel dataset of Press release done by the government of United States comparative study the performance of various text classification approaches for legal area classification.

5. Future Scope

This project has a potential to grow on a larger scale. Automated systems may allow lawyers to build cases more quickly, such as aiding overworked public defenders and provide access to the law to a greater number of people (such as people reading up on the law to defend themselves).

Chapter 2

Review of Literature

Papers closest to ours are those that likewise examine legal area classification. In this paper, they found the best method for automated legal document classification is the SCC system that uses a CNN (72.4% accuracy for 15 general categories and 31.9% accuracy for the 279 more specific categories). On the other hand, the GRU architecture shows promising results compared to our tuned CNN (nearly as high for the 15 category task). A tuned GRU-based network can potentially complete the task with higher accuracy. The SCC system uses word embeddings from a general domain (Google News). It is possible that embeddings from the legal domain would improve results. Accordingly, we plan to compile a much larger corpus of US legal opinions from appellate and local courts in order to generate domain-specific word embeddings for their model. They conduct experiments using these embeddings instead of the Google News embeddings. [1]

Goncalves and Quaresma (2005) used bag-of-words (“BOW”) features learned using TF-IDF to train linear support vector machines (“linSVMs”) to classify decisions of the Portuguese Attorney General’s Office into 10 legal areas. Boella et al. (2012) used TF-IDF features enriched by a semi-automatically linked legal ontology and linSVMs to classify Italian legislation into 15 civil law areas. Sulea et al. (2017) classified French Supreme Court judgments into 8 civil law areas, again using BOW features learned using Latent Semantic Analysis (“LSA”) (Deerwester et al., 1990) and linSVMs. On legal text classification more generally, Aletras et al. (2016); Liu and Chen (2017); Sulea et al. (2017) used BOW features extracted from judgments and linSVMs for predicting case outcomes. Talley and O’Kane (2012) use BOW features and a linSVM to classify contract clauses. NLP has also been used for legal information extraction. Venkatesh (2013) used Latent Dirichlet Allocation (Blei et al., 2003) (“LDA”) to cluster Indian court judgments. Falakmasir and Ashley (2017) used vector space models to extract legal factors motivating case outcomes from American trade secret misappropriation judgments. There is also growing scholarship on legal text analysis. Typically, topic models are used to extract N-gram clusters from legal corpora, such as Constitutions, statutes, and Parliamentary records, then assessed for legal significance (Young, 2013; Carter et al., 2016). More recently, Ash and Chen (2019) used

document embeddings trained on United States Supreme Court judgments to encode and study spatial and temporal patterns across federal judges and appellate courts. We contribute to this literature by (1) benchmarking new text classification techniques against legal area classification, and (2) more deeply exploring how document scarcity and length affect performance. Beyond BOW features and linSVMs, we use word embeddings and newly developed language models. Our novel label set comprises 31 legal areas relevant to Singapore’s common law system. Judgments of the Singapore Supreme Court have thus far not been exploited. We also draw an important but overlooked distinction between cases and judgments. [2]

Chapter 3

System Analysis

3.1 Functional Requirements:

1. USER INTERFACE:

The user interface will be an application. The user will be given a text area which he wants to classify.

2. PROPER FORECASTING:

The system has to properly classify the text that the user has given as input into the specific category.

3. DATABASE:

Dataset contains large number of legal text passage for the system to classify the category accurately

3.2 Non-Functional Requirements:

1. Platform Independent:

The application would be platform-independent if all the requirements are installed in the device.

2. Performance:

The application should have better accuracy and should provide the information in less time

3. Capacity:

The capacity of the storage should be high so that a large amount of data can be stored in order to train the model.

3.3 Performance Requirements

The system response should be less than 5 seconds.

The system must process the number of transactions based on the following calculation method:

1. Safety Requirements

There are no specific safety requirements associated with the proposed system. The website executes on well-known and commonly used hardware which does not cause any safety hazards

2. Security Requirements

Updates shall only be made by authorized developers. The administrator of the system is the only one responsible for the change of all the system data.

3. Software Quality Attributes

Reliability:

The website should provide reliability to the user that it will run stably with all features mentioned above available and executing perfectly.

Resources:

The system should be designed in such a way that the requests of the user can be fulfilled with the minimum number of resource utilization, thus improving speed.

3.4 Specific. Requirements

3.4.1 Hardware Requirements:

- CPU --Intel Core i5 9300H
- Hard Disk Space -- 1TB
- Memory -- 2666Hz 8gb DDR4 ram
- Other Devices – Laptop

3.4.2 Software Requirements:

- Front End --Flask
- Back End -- Python
- Languages -- Python
- Operating System -- WINDOWS 10

3.5 Use case diagram

Following figure (Fig 3.5.1) illustrates the Use case Diagram



Fig 3.5.1: Use Case Diagram

Chapter 4

Analysis Modeling

4.1 DFD

The DFD shown in the following figure (Fig 4.1.1) represents the flow of the system. The unclassified data given by the user is fed to the model which processes it and produces the result in the form of the class/category that text falls into.

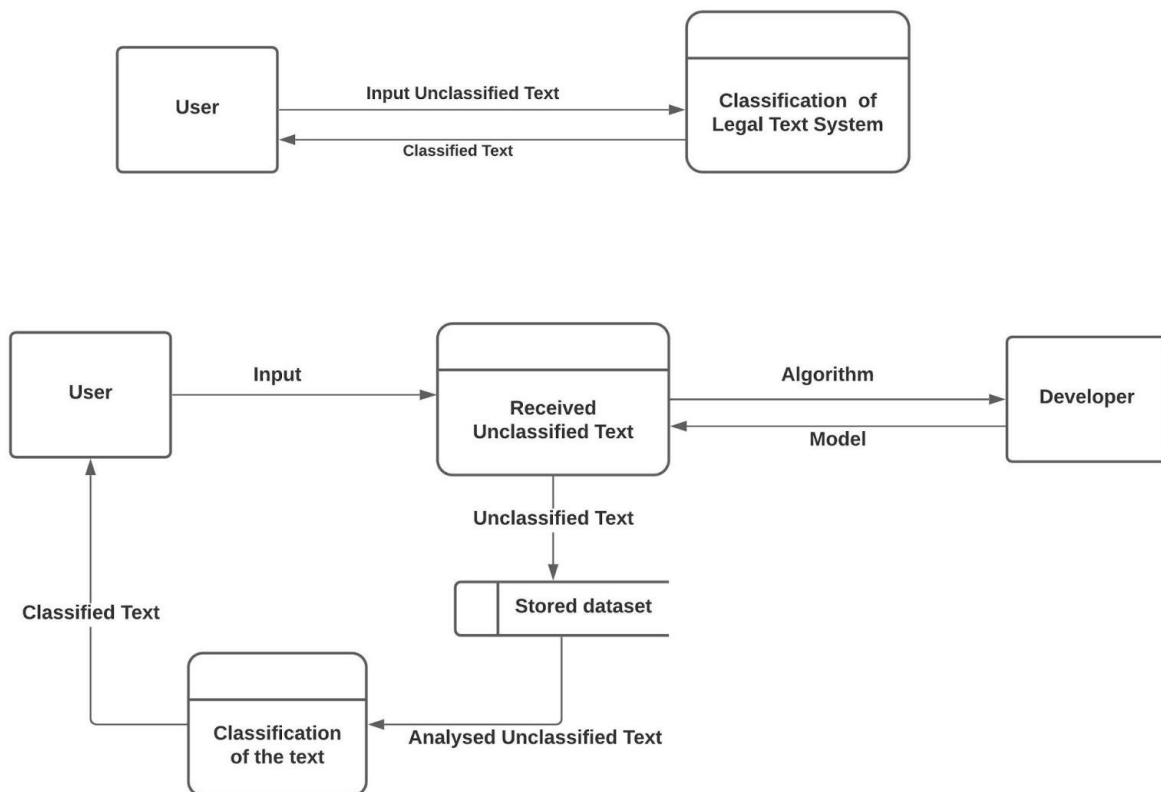


Fig 4.1.1: DFD

Chapter 5

Design

5.1 Architectural Design

- We first trained the clustering model by feeding them the data set to train.
- Then the text is processed and transformed and given to the model.
- The transformed text then gets processed and we get the class it falls into.

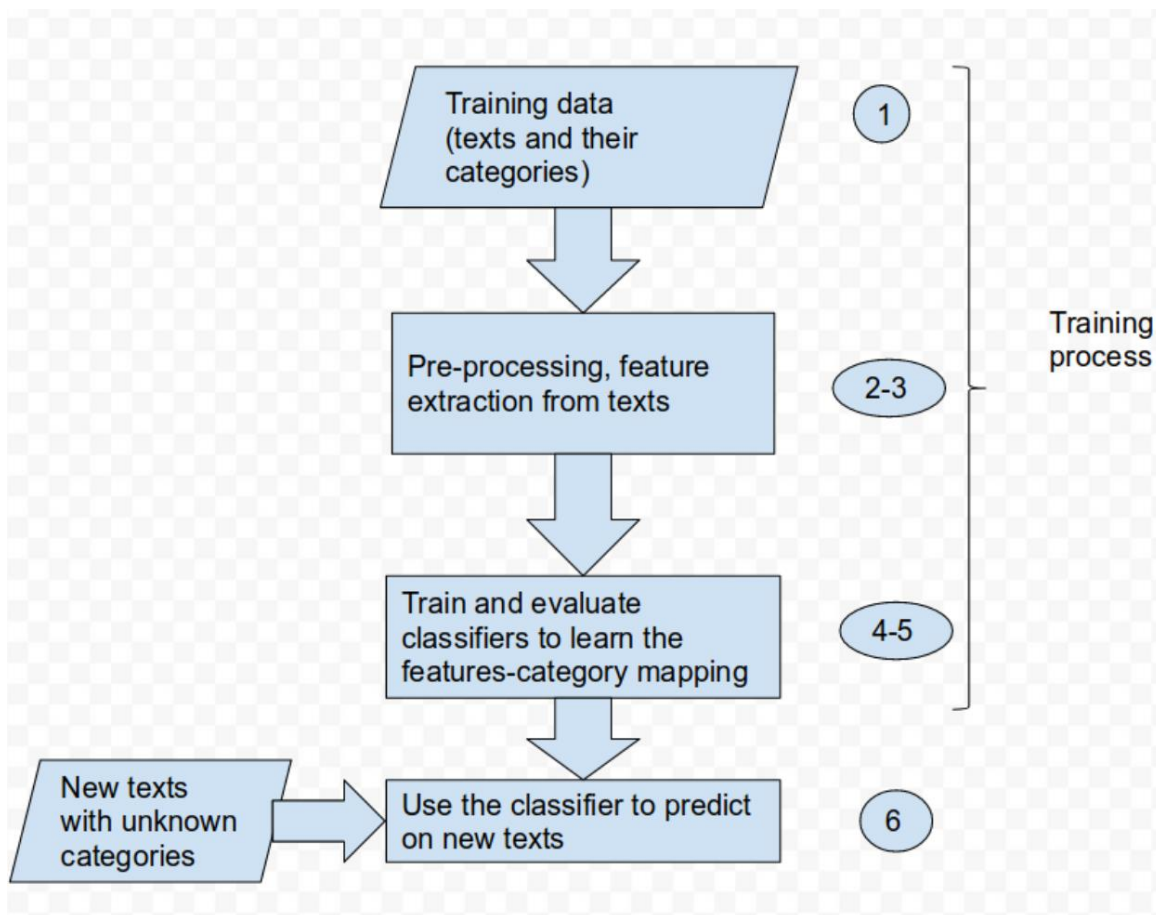
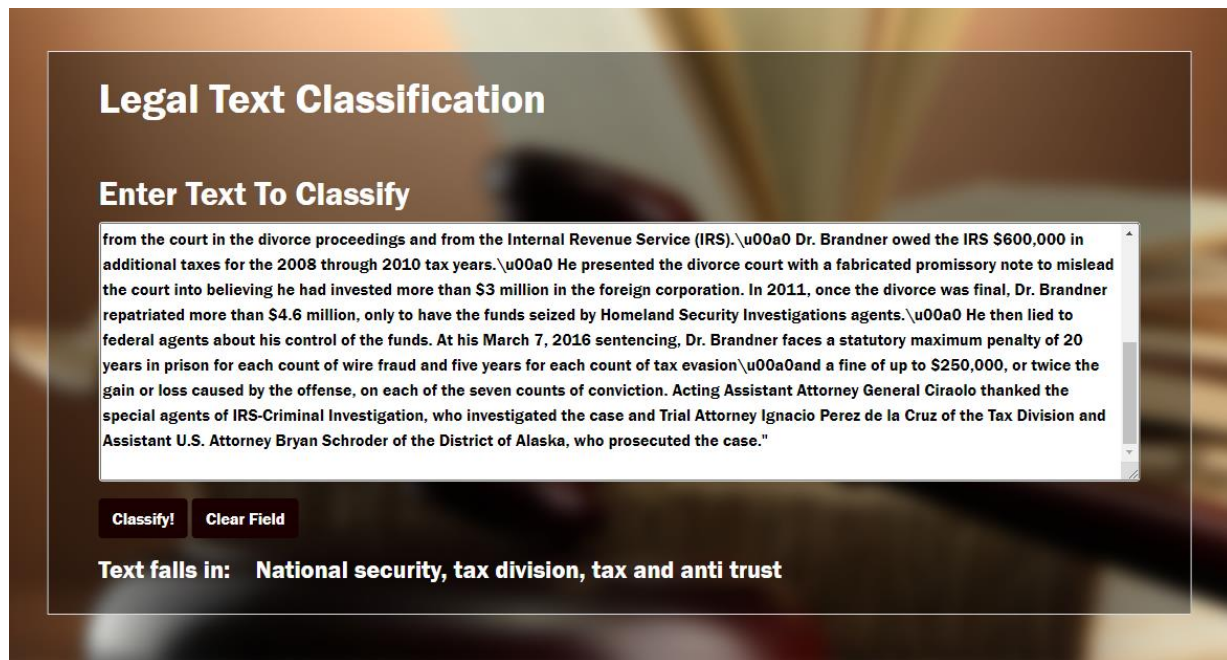


Fig 5.1.1: Work flow of System

5.2 User Interface

The User Interface for the model of text classification was implemented as a website application. HTML, CSS and Flask were used to make the entire application. Here, the user inputs the legal text paragraph in the space provided and after hitting the classify button, he is given output of the laws and divisions related to the text. The image representation below (Fig 5.2.1) shows the text given as an input and its corresponding class(es).



The screenshot shows a web application titled "Legal Text Classification". Below the title is a section labeled "Enter Text To Classify". Inside this section is a text input field containing a paragraph about Dr. Brandner's legal case. Below the input field are two buttons: "Classify!" and "Clear Field". Below the buttons, the output is displayed as "Text falls in: National security, tax division, tax and anti trust".

Legal Text Classification

Enter Text To Classify

from the court in the divorce proceedings and from the Internal Revenue Service (IRS). Dr. Brandner owed the IRS \$600,000 in additional taxes for the 2008 through 2010 tax years. He presented the divorce court with a fabricated promissory note to mislead the court into believing he had invested more than \$3 million in the foreign corporation. In 2011, once the divorce was final, Dr. Brandner repatriated more than \$4.6 million, only to have the funds seized by Homeland Security Investigations agents. He then lied to federal agents about his control of the funds. At his March 7, 2016 sentencing, Dr. Brandner faces a statutory maximum penalty of 20 years in prison for each count of wire fraud and five years for each count of tax evasion and a fine of up to \$250,000, or twice the gain or loss caused by the offense, on each of the seven counts of conviction. Acting Assistant Attorney General Ciruolo thanked the special agents of IRS-Criminal Investigation, who investigated the case and Trial Attorney Ignacio Perez de la Cruz of the Tax Division and Assistant U.S. Attorney Bryan Schroder of the District of Alaska, who prosecuted the case."

Classify! **Clear Field**

Text falls in: National security, tax division, tax and anti trust

Fig 5.2.1: User Interface Design

Chapter 6

Implementation

6.1 Algorithms / Methods Used:

We used a **Clustering Algorithm- K Means** from scikit learn library to classify the legal text which the user has given input.

To process the learning data, the K-means algorithm starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.

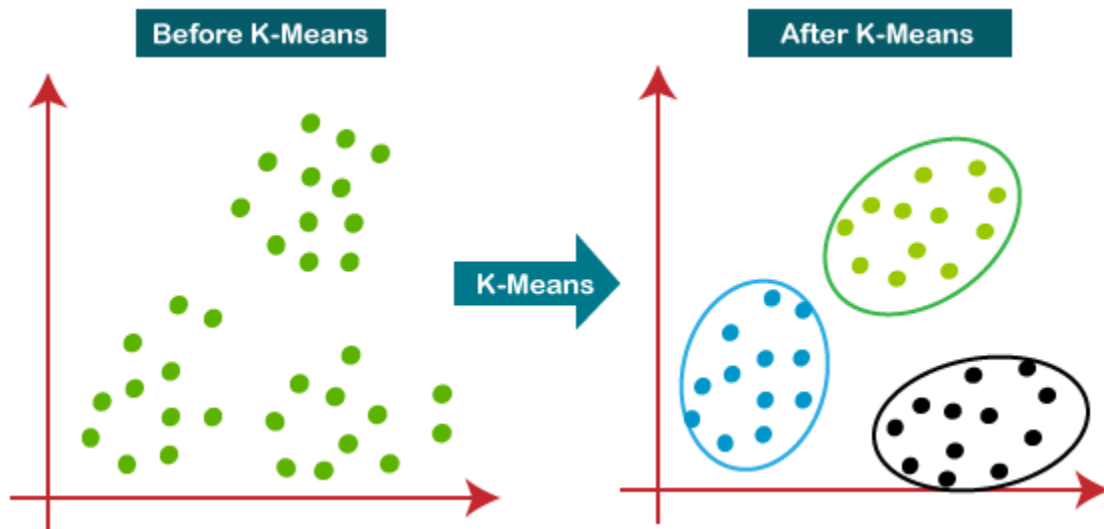


Fig 6.1.1: K Means Clustering

6.2 Working of the project

In this project, we have used TFIDF vectorizer and MiniBatchKmeans to perform some simple document clustering. Importing the dataset is done using Pandas. The source file is a newline delimited JSON file.

Extracting keywords

Here we have used the TfidfVectorizer since the IDF score will pull out unique words that can be used in clustering.

Finding Optimal Clusters

Clustering is an unsupervised operation, and KMeans requires that we specify the number of clusters. One simple approach is to plot the SSE for a range of cluster sizes. We look for the "elbow" where the SSE begins to level off. Unfortunately the regular Kmeans implementation is too slow so we went with MiniBatchKMeans. MiniBatchKMeans introduced some noise so we raised the batch and init sizes higher. The elbow point obtained was at 14 (Fig: 6.2.1) so we chose 14 clusters.

Program code: Function to find optimal clusters

```
def find_optimal_clusters(data, max_k):
    iters = range(2, max_k+1, 2)
    sse = []
    for k in iters:
        sse.append(MiniBatchKMeans(n_clusters=k, init_size=1024, batch_size=2048,
random_state=20).fit(data).inertia_)
        print('Fit { } clusters'.format(k))

    f, ax = plt.subplots(1, 1)
    ax.plot(iters, sse, marker='o')
    ax.set_xlabel('Cluster Centers')
```



```
ax.set_xticks(iters)
ax.set_xticklabels(iters)
ax.set_ylabel('SSE')
ax.set_title('SSE by Cluster Center Plot')
```

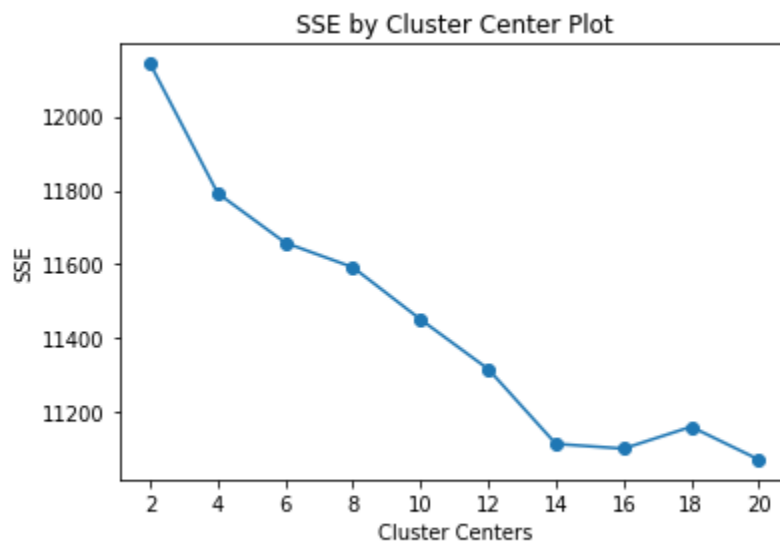


Fig 6.2.1: SSE plot for a range of clusters

Plotting Clusters

Here we plot the clusters generated by our KMeans operation. One plot uses PCA which is better at capturing global structure of the data. The other uses TSNE which is better at capturing relations between neighbors. In order to speed up the process with TSNE, we sample from 3,000 documents and perform a PCA 50 dimension reduction on the data first. Next we show a scatterplot further sampling the sample down to 300 points.

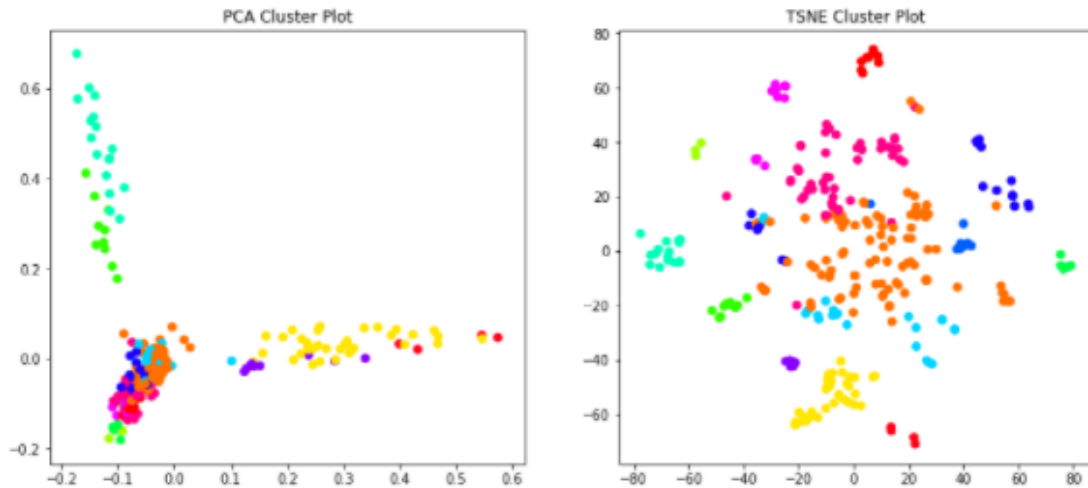


Fig 6.2.2: Data Visualization

Top Keywords

Lastly, we'll cycle through the clusters and print out the top keywords based on their TFIDF score to see if we can spot any trends. We'll do this by computing an average value across all dimensions in Pandas, grouped by the cluster label. Using numpy, finding the top words is simply sorting the average values for each row, and taking the top N.

You can see that we have a pretty good result. Topics including exploitation of children, tax fraud, civil rights, and environmental issues can be inferred from the top keywords.

Program code: Function to find optimal clusters

```
def get_top_keywords(data, clusters, labels, n_terms):

    df = pd.DataFrame(data.todense()).groupby(clusters).mean()

    for i,r in df.iterrows():

        print('\nCluster {}'.format(i))

        cluster_list.append(f"Cluster {i}")
```

```

print(', '.join([labels[t] for t in np.argsort(r)[-n_terms:])))

actual_list.append([labels[t] for t in np.argsort(r)[-n_terms:]])

Cluster 0
images,sexual,children,ceos,safe,project,exploitation,childhood,pornography,child

Cluster 1
fbi,states,office,united,fraud,indictment,department,criminal,district,attorney

Cluster 2
revenue,division,refunds,taxes,attorney,false,income,returns,irs,tax

Cluster 3
hotline,provision,status,800,citizenship,immigration,ina,employment,discrimination,osc

Cluster 4
patients,medicaid,fraud,services,false,settlement,medicare,claims,care,health

Cluster 5
department,civil,lawsuit,rights,act,disabilities,hud,discrimination,fair,housing

Cluster 6
beneficiaries,home,strike,oig,services,care,hhs,fraud,health,medicare

Cluster 7
division,foreclosure,mortgage,task,rigging,bid,auctions,fraud,financial,antitrust

Cluster 8
county,police,ms,texas,abt,aka,murder,racketeering,members,gang

Cluster 9
environment,act,pollution,oil,clean,air,water,environmental,epa,settlement

Cluster 10
banks,undeclared,offshore,ubs,irs,account,tax,swiss,bank,accounts

Cluster 11
act,department,514,agreement,access,election,rights,disabilities,voting,ada

Cluster 12
division,victim,police,officers,law,attorney,justice,department,civil,rights

Cluster 13
prepared,preparing,return,irs,income,injunction,complaint,customers,returns,tax

```

Fig 6.2.3: Cluster with top keywords

Using these top keywords obtained, any input data is classified after extracting the keywords from the text and comparing them to that obtained by the model.

Chapter 7

Conclusion

In this paper we investigated the application of text classification methods to the legal domain using the cases and rulings from the Department of Justice's press releases . We showed that a system based on clustering algorithm (K Means) can obtain high scores in predicting the law area and the ruling of a case, given the case description, and the time span of cases and rulings.

This paper comparatively benchmarked traditional topic models against more recent, sophisticated, and computationally intensive techniques on the legal area classification task. Our results also suggest that more work can be done to adapt state-of-the-art NLP models for the legal domain. Legal text Classification is an automated document classification system which is used to analyse and categorize the legal documents. It is easier for a layman to understand the actual meaning of legal text without the need of an expert consultation. The work presented in this paper confirms that text classification techniques can indeed be used to provide a valuable assistive technology base as support for law professionals in obtaining guidance and orientation from large corpora of previous court rulings. In future work, we would like to investigate the extent to which a more accurate draft form can be induced from the court's case description.

References

- [1] Samir Undavia, Adam Meyers, John E. Ortega, “A Comparative Study of Classifying Legal Documents with Neural Networks”, Proceedings of the Federated Conference on Computer Science and Information Systems pp. 515–522, 2018
- [2] Jerrold Tsin Howe SOH, How Khang LIM, Ian Ernst CHAI , “Legal topic classification: A comparative study of text classifiers on Singapore Supreme Court judgments” Proceedings of the Natural Legal Language Processing Workshop 2019, pages 67–77 ,June 7, 2019
- [3] Dhriya V, “VLegal Text Classification in to Practice Areas Using Machine Learning ,” International Journal of Computer Science and Engineering Communications, Volume.5, Issue.3 (2017): Page.1581-1586
- [4] Wan, L., Papageorgiou, G., Seddon, M., & Bernardoni, M. (2019) “Long-length Legal Document Classification”
- [5] R. Bartolini, A. Lenci, S. Montemagni, V. Pirrelli, and C. Soria. “Automatic classification and analysis of provisions in italian legal texts: a case study” in Proceedings of the Second International Workshop on Regulatory Ontologies, 2004.

Acknowledgements

We feel immense pleasure in submitting this report on “**Legal Text Classification**”. While submitting this report we avail this opportunity to express our gratitude towards the mentor who has guided and helped us in completing this task successfully. We owe a deep gratitude to our guide **Mrs. Varsha Shrivastava** who proved to be mere to guide us. Apart from bringing to us what can be joy for creation, every time she acted promptly to correct our mistakes. The successful completion of this project was possible by her guidance and co-operation only.

Sincere thanks to group members.