ISM6930 - Data Science Programming

Clustering of Legal Summary Reports

by

Rushikesh Maheshwari, Vijaya Singh, and Priyanka Sariya

Muma College of Business

University of South Florida, Tampa, FL

(rmaheshwari@mail.usf.edu, vijayasingh@mail.usf.edu, psariya@mail.usf.edu)

***Abstract**: In this report we present our approach towards using unsupervised clustering techniques to cluster legal-summary documents. In legal cases, the arguments and case points are often constructed referring to past cases. This is a very common task that paralegals carry out today by manually browsing through thousands of reports to find relevant legal case reports so as to cite from them. Our goal is to provide with segmented documents by types of cases so that the paralegals has one less task in their hand. The project includes collating legal summary from 3890 documents, cleaning and preprocessing the text, converting text into vector form to use it with different clustering algorithms and coming up with clustered documents.*

1. Introduction

The legal cases across many countries demands robust documentation at every stage of the proceeding of the case and hence has an abundance of textual data. Thus, owing to the large amount of legal knowledge contained in written form, the legal sector is in need of faster information retrieval. In a process known as stare decisis, past decisions can have a binding effect on future decisions, especially in countries with common law systems such as Australia, the UK and the USA. As a result, judges need to learn past cases to be reliable, and attorneys use them to "only" find arguments for their case in their application of law. Court decisions or cases can be instructive in introducing a new principle or rule, updating or interpreting an established principle or rule, or addressing an issue on which the law is unclear.

In this report we outline our approach towards providing segments of similar documents using text mining and clustering techniques.

The next section deals with related work done in document clustering. After this is the summary in Section 3 of our annotated legal corpus. We present our approach to the text vectorization and clustering methods in Section 4. Our first findings on legal document clustering are covered in Section 5 below followed by corrective actions to have some useful insights. Our achievements so far are discussed in the final section 6 that also outlines future research.

2. Related work

There has been an ample amount of research work done in the area of text documents segregation in last decade given the increase in text-data and easier information retrieval. Text documents has many features like words, sentences, semantic, sentiments, hidden topics etc. and all were explored for clustering.

Hierarchical text document clustering algorithms are often used to organize news articles/blogs (text available via online stream) however popular *Cobweb* and *Classit* algorithms in text documents [1]. To address the text document clustering, the frequent co-occurring words (item sets) derived from association mining are used as features for clustering documents and to derive the topic tree for clusters [2]. Representation of text document as a hierarchical graph utilizing apriori paradigm leading to subgraphs which captures the frequent senses(contexts) and these are later used to generate sense-based document clusters [3].

The central theme in a legal document can be multi-topical, which has very specific and professional, domain-specific language, which addresses broad and wavy coverage of legal issues. Hence it has been a challenge to cluster such multi-topical legal documents. An attempt with classification-based recursive soft-clustering algorithm which also avails built-in topic segmentation have been proven to have clustered legal documents with efficiency close to those created by a domain expert [4][8]

In this project we tried to build the cluster of legal documents using TFIDF and Word2Vec text vectorizer techniques along with K-means clustering algorithm. We found that the clustering of

documents can be very useful to replace the conventional, often manual research and interpretation of the legal text corpora.

3. Data

AustLII (Australian Legal Information Institute) [6] provides free access to certified copies and summary reports of legal cases in Australian courts. From this source, we downloaded court case documents in XML format which included all cases from the year 2006, 2007, 2008 and 2009. Our dataset contains Australian Federal Court of Australia (FCA) legal cases. The reports had sentence wise summary, catchphrases and citations. We worked with sentences and catchphrases (key pointers in the summary documents listed above the sentences) separately**.**

3.1 Data Extraction

Downloaded court case documents were all XML files, which has many tags present in it. As stated above, we focused text of "sentences" and "catchphrases" tags. We used HTML parser and extracted only specified text we wished to perform text analytics on. Below is a snapshot of XML file:

```
<catchphrases>
<catchphrase "id=c0">application for leave to appeal</catchphrase>
<catchphrase "id=c1">authorisation of multiple infringements of copyright established</catchphrase>
<catchphrase "id=c2">prior sale of realty of one respondent to primary proceedings</catchphrase>
<catchphrase "id=c3">payment of substantial part of proceeds of sale to offshore company in purported repayment
<catchphrase "id=c4">absence of material establishing original making and purpose of loan</catchphrase>
<catchphrase "id=c5">mareva and ancillary orders made by primary judge</catchphrase>
<catchphrase "id=c6">affidavits disclosing assets sworn</catchphrase>
<catchphrase "id=c7">orders made requiring filing of further affidavits of disclosure and cross-examination of
</catchphrase>
<catchphrase "id=c8">no error in making further ancillary orders</catchphrase>
<catchphrase "id=c9">leave refused</catchphrase>
<catchphrase "id=c10">practice and procedure</catchphrase>
</catchphrases>
```

```
<sentences>
<sentence id="s0">
 Background to the current application

1 The applicants Sharman Networks Ltd ('Sharman Networks'), Sharman License Holdings Ltd ('Sharman License') and Ms
preservation orders made by Wilcox J on 22 March 2005 ('the Mareva orders').</sentence>
<sentence id="s1">When referring to the applicants generally, I will do so as 'the Sharman applicants'.</sentence>
<sentence id="s2">Each of the Sharman applicants was one of ten respondents to infringement of copyright proceedin
respect of the operation of what was described by the parties as the 'Kazaa system' ('the primary proceedings').</
<sentence id="s3">Wilcox J made orders ancillary to the Mareva orders on 22 March 2005 requiring each of the Sharma
of all of their assets, wherever situated, and to specify whether those assets were held by each applicant either
<sentence id="s4">2 Wilcox J delivered judgment on the complex issues of liability arising in the primary proceedi
Sharman License Holdings Ltd (2005) 220 ALR 1).</sentence>
<sentence id="s5">In the meantime, Ms Hemming had filed two disclosure affidavits pursuant to Wilcox J's orders of
unsuccessfully sought several stays on various grounds of that same order insofar as it applied to them (see Univer
FCA 406 per Hely J, delivered 8 April 2005; Universal Music Australia Pty Ltd v Sharman License Holdings Ltd [2005]
License Holdings Ltd v Universal Music Australia Pty Ltd [2005] FCA 505 per Moore J, delivered 28 April 2005).</se
<sentence id="s6">Disclosure affidavits were eventually sworn on behalf of Sharman License and Sharman Networks by
affidavits sworn also by Mr Gee on 16 June 2005.</sentence>
<sentence id="s7">Sharman License and Sharman Networks had also unsuccessfully sought an enlargement of time in wh
```

3.2 Data Cleaning

3.2.1 Removal of special characters: While going through the extracted text, we found that it had a lot of junk data which needed to be cleaned. There were special characters and unnecessary http link and also 'www' site name mentioned which we did not require. We got rid of it with the help of regex expression.

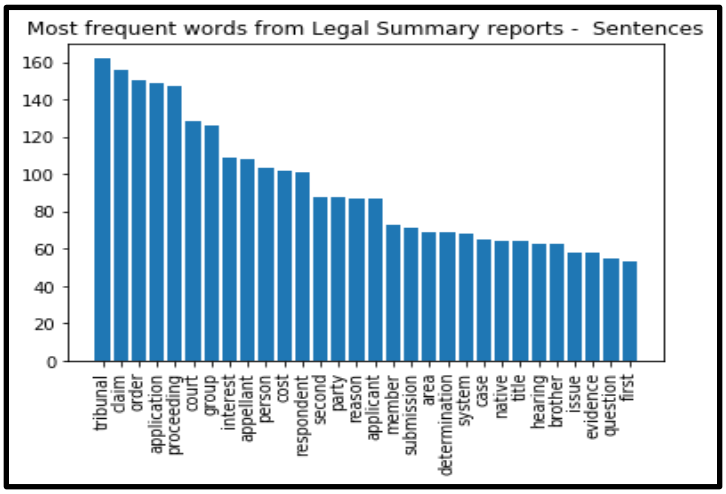
3.2.2 Removal of Stop words & Punctuation: We got rid of stop words and punctuation because stop words do not carry much of the relevance. Also, we got rid of the words whose word-length is less than 3. This is kind of heuristic approach we decided to have in our text analysis.

3.2.3 Removal of Abbreviations: In the text extracted, it contained many acronyms of different company names. In order to get rid of all the abbreviated words, we went to Australian government site and from that HTML page, collected all abbreviations and made a list of it. We excluded those acronyms from our text corpus which were present in the list [7]

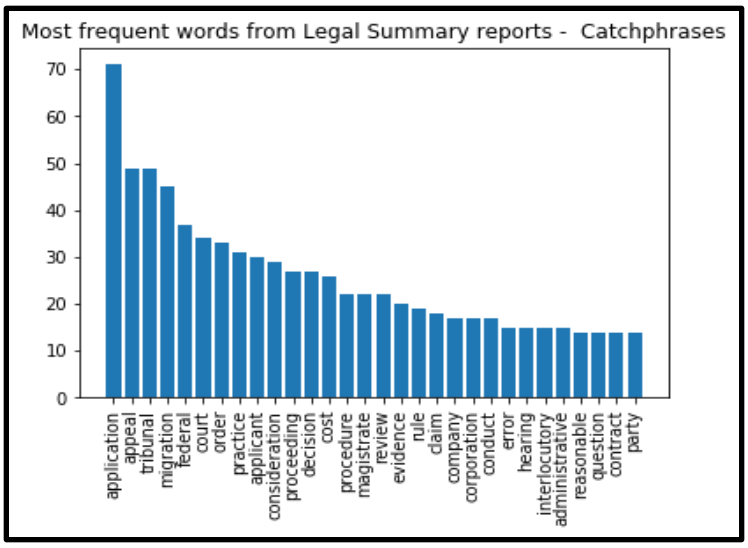
3.2.4 Filtering Noun and Verbs: For performing text analytics, we needed to come up with "bag of words" for each document, but not all types of words are relevant and they can be removed. Like adjectives, conjunctures positioned words can be easily eliminated. For our analysis part we only kept those words which are Nouns or Verbs.

3.2.5 Perform Lemmatization: To reduce the inflectional forms of each word to its root word, we performed lemmatization.

3.3 Exploratory Data Analysis



The clean text was broken into word tokens which resulted into 44,567,43 word. So, we computed a frequency of words and plotted top 30 most used words in the text corpus on a sample of 10 documents at random.



The similar word frequency was computed for catchphrases with 100 documents. The graph here illustrates the most common words appearing in catchphrases.

3.4 Data Pre-processing

3.4.1 Converting text into TF-IDF: As machine learning algorithm only understands numbers and that is why we needed to convert "cleaned text" into some form of vector numbers. This was done using TFIDF vectorizer. It captures the normalized frequency of a token appearing in a given document (TF) and log inverse of the frequency of appearance of a token across multiple documents (IDF). The multiplication of these two generates a TFIDF matrix where tokens are treated as columns while number of rows are equal to number of documents. We identified 64053 words across 3890 documents.

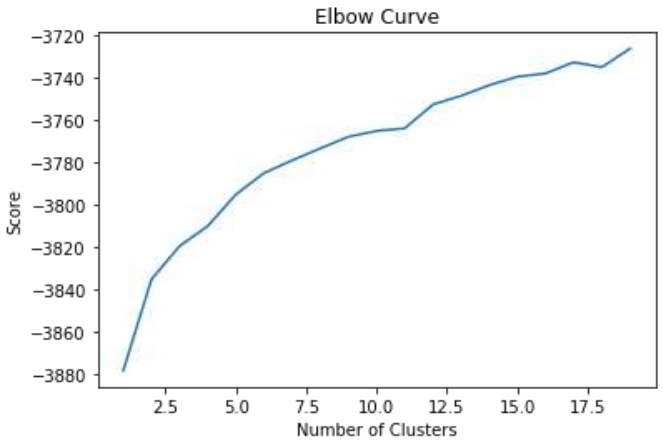
3.4.2 Converting text into Word2Vec: To capture the semantics, interdependencies, co-occurrences of text in different documents, Word2Vec was used for our further analysis. Word2Vec represents a word in a dense vector which capture the semantics. These vectors are used with clustering algorithm.

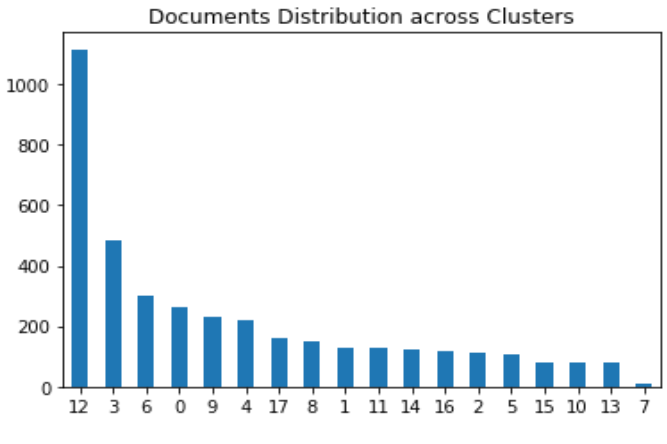
4. Document Clustering

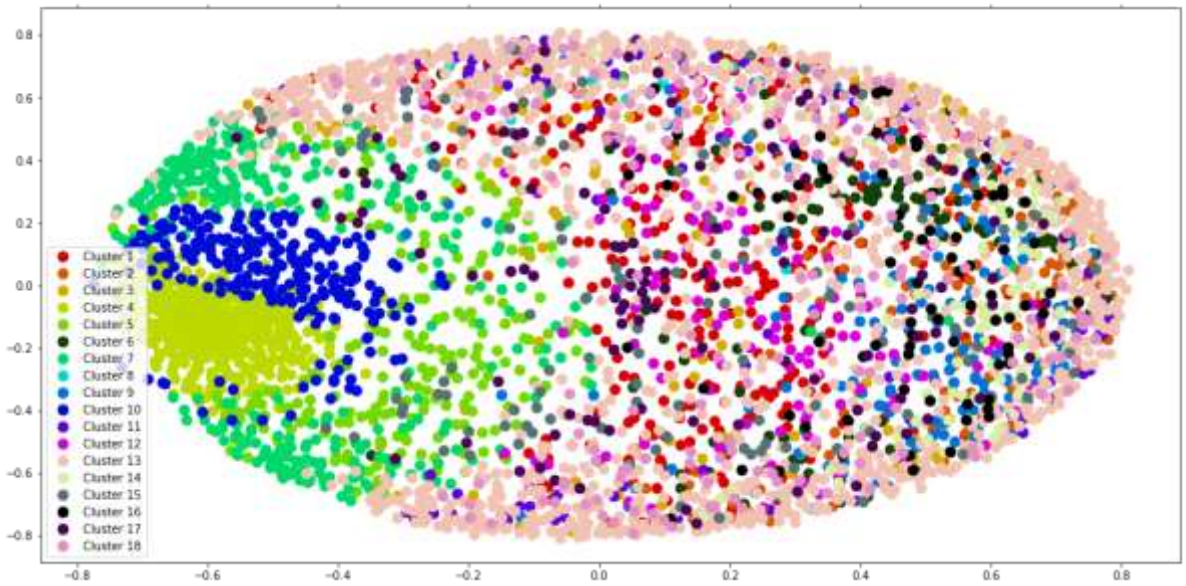
4.1 Clustering using Sentences with TFIDF

The TFIDF representation of word tokens generated with ‘sentence’ in summary reports were fed to K-means clustering algorithm. The number for initial clusters (K) is generally determined by using elbow curve method. Since the cases were taken from Federal Court of Australia (FCA), we researched the general types of legal cases addressed in FCA. The National Practice Areas (NPAs) mentioned on their website addresses such 18 categories of legal cases. Considering this as a ground truth and looking at the elbow curve, we chose the K-value (18) and ran the clustering [5]

Image of clusters with sentence along with table of cluster centers and words





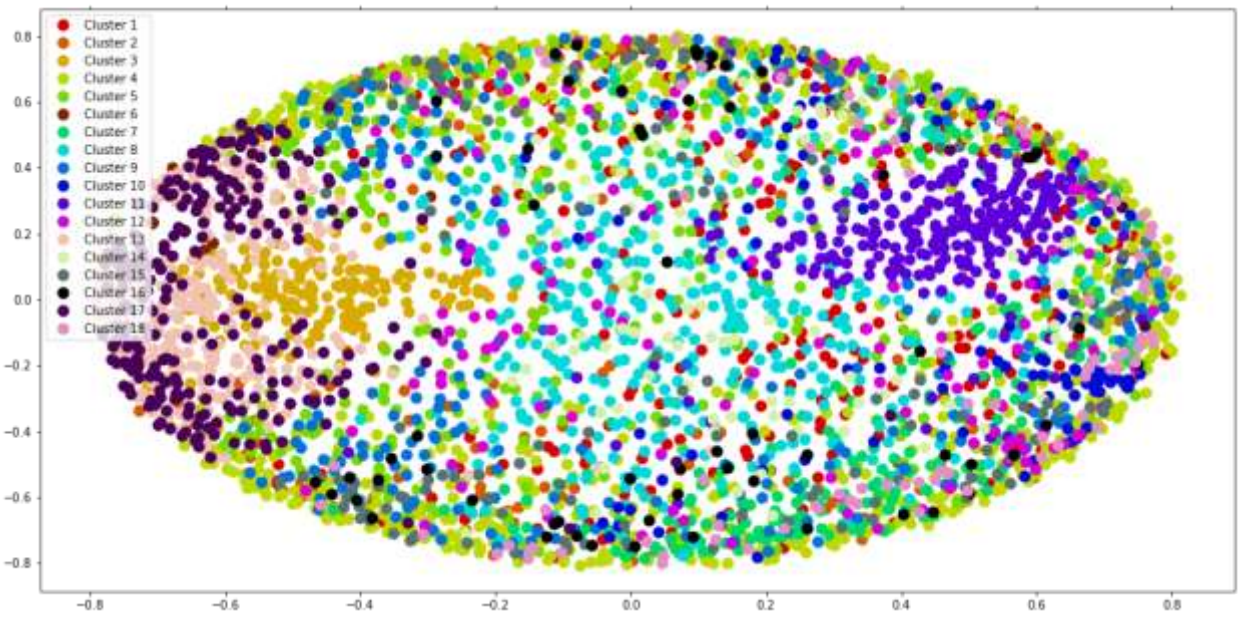


Cluster centroids mapped with top 10 words at centers from TFIDF

Cluster 1:	claim, motion, trial, statement, second, security, pleading, notice, cross, respondents,
Cluster 2:	scheme, share, shareholder, meeting, company, plaintiff, member, asic, security, director,
Cluster 3:	mark, extradition, trade, offence, warrant, good, person, custom, summons, criminal,
Cluster 4:	appellant, tribunal, magistrate, minister, appeal, federal, persecution, information, immigration, review,
Cluster 5:	appeal, magistrate, federal, appellant, tribunal, minister, extension, notice, rule, immigration,
Cluster 6:	native, title, area, land, group, claim, determination, traditional, people, right,
Cluster 7:	tribunal, minister, visa, appeal, person, injury, appellant, veteran, child, pension,
Cluster 8:	allphones, franchisees, franchise, agreement, mobile, document, chew, claim, trial, group,
Cluster 9:	liquidator, company, asic, carey, winding, receiver, westpoint, corporation, defendant, property,
Cluster 10:	tribunal, magistrate, appeal, minister, appellant, review, federal, immigration, visa, ground,
Cluster 11:	patent, invention, claim, product, infringement, specification, university, commissioner, document, patentee,
Cluster 12:	document, privilege, discovery, legal, advice, professional, production, affidavit, disclosure, communication,
Cluster 13:	claim, agreement, service, person, company, contract, conduct, product, affidavit, document,
Cluster 14:	creditor, administrator, company, doca, deed, meeting, administration, liquidator, winding, debt,
Cluster 15:	bankruptcy, bankrupt, trustee, debtor, creditor, property, petition, notice, debt, sequestration,
Cluster 16:	defendant, plaintiff, company, demand, claim, plaintiffs, affidavit, offer, winding, defendants,
Cluster 17:	income, commissioner, taxpayer, assessment, taxation, year, assessable, tribunal, objection, itaa,
Cluster 18:	employee, union, penalty, agreement, industrial, site, work, contravention, employer, workplace,

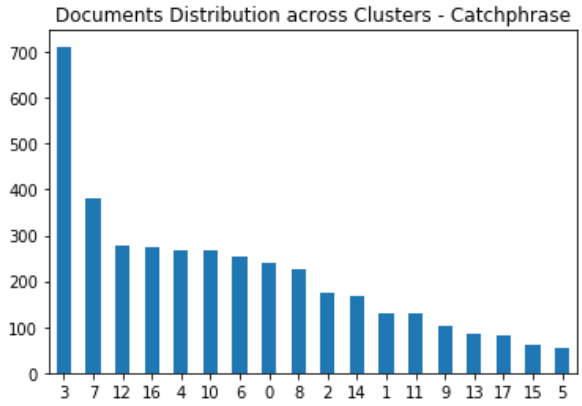
4.2 Clustering using Catchphrases with TFIDF

The catchphrases are the key points captured in the summary report. The image below shows the cluster of documents based on the text in catchphrases in each document.



Cluster centroids mapped with top 10 words at centers from TFIDF

Cluster 1	scheme, corporation, meeting, arrangement, company, creditor, convening, member, administration, approval,
Cluster 2	tribunal, migration, appellant, visa, review, refugee, protection, information, error, decision,
Cluster 3	claim, statement, pleading, practice, procedure, application, strike, motion, cause, action,
Cluster 4	native, title, determination, application, group, claim, claimant, consent, applicant, registration,
Cluster 5	income, taxation, assessment, taxpayer, commissioner, business, objection, assessable, benefit, capital,
Cluster 6	privilege, legal, professional, document, advice, waiver, communication, discovery, procedure, claim,
Cluster 7	court, procedure, proceeding, practice, order, discovery, federal, rule, application, document,
Cluster 8	point, principle, migration, question, appeal, magistrate, federal, decision, immigration, human,
Cluster 9	evidence, application, applicant, contract, patent, respondent, practice, procedure, agreement, court,
Cluster 10	penalty, industrial, relation, workplace, agreement, employee, contravention, union, breach, pecuniary,
Cluster 11	magistrate, federal, appeal, migration, decision, error, court, tribunal, application, jurisdictional,
Cluster 12	administrative, tribunal, appeal, decision, compensation, review, applicant, question, judicial, entitlement,
Cluster 13	trade, mark, practice, misleading, representation, conduct, deceptive, good, contract, damage,
Cluster 14	appeal, application, extension, time, file, migration, procedure, notice, practice, interlocutory,
Cluster 15	bankruptcy, bankrupt, notice, sequestration, order, creditor, petition, trustee, application, court,
Cluster 16	injunction, interlocutory, convenience, balance, relief, application, question, practice, restrain, patent,
Cluster 17	cost, order, security, indemnity, applicant, respondent, application, party, offer, proceeding,
Cluster 18	corporation, company, liquidator, winding, application, insolvency, order, creditor, liquidation, defendant,



4.3 Removing Insignificant tokens from the Text Corpus

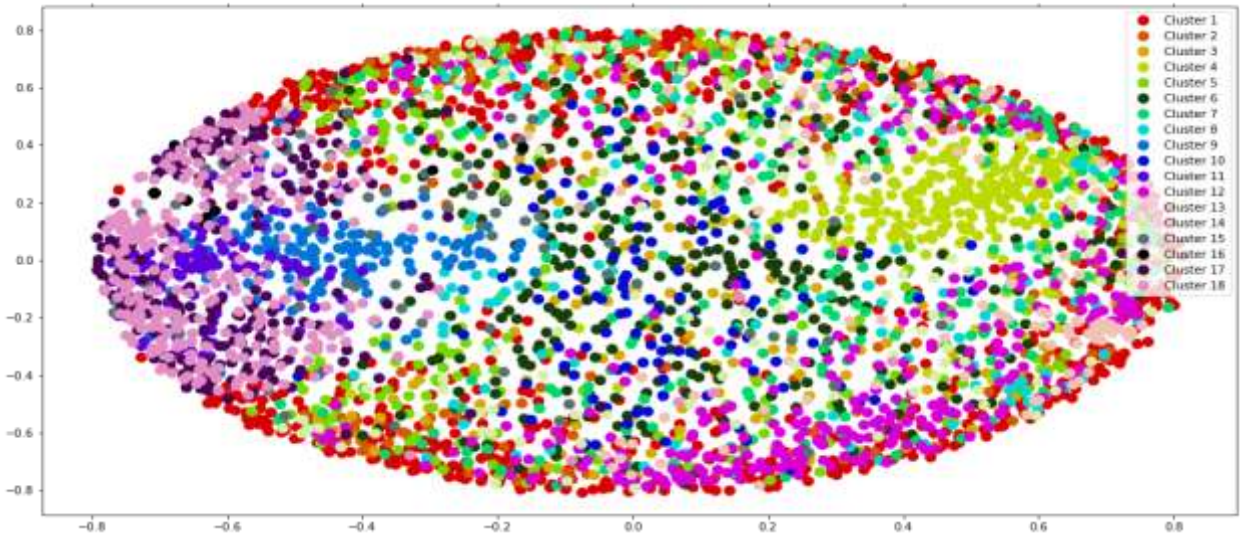
From tables above, we observed some tokens are not very significant to the respective cluster and are very commonly used in legal documents or general context which are present in text corpus. So, we removed such words from the text corpus retaining more significant tokens and ran another iteration of the clustering to see if the clusters converge better.

A list of such words is provided below.

'case','date','court','hearing','counsel','application','copy','time','matter','appellant','appeal','trial', 'review','tribunal','notice','fact','law','year','state','term','section','provision','considertaion','good'

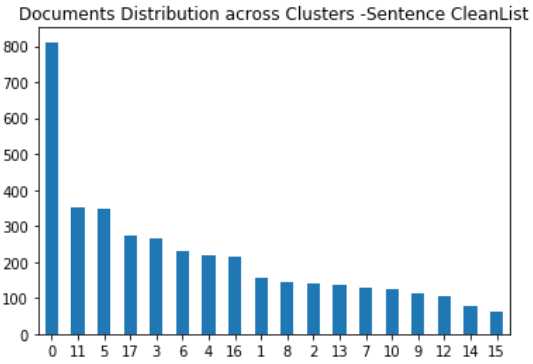
,'invention','conduct','plaintiff','url','title','group','rule','word','term','person','solicitor','honor', 'decision','party','reason','purpose','reference','issue','right','proceeding'

4.4 Clustering using Sentences with TFIDF (without insignificant words)



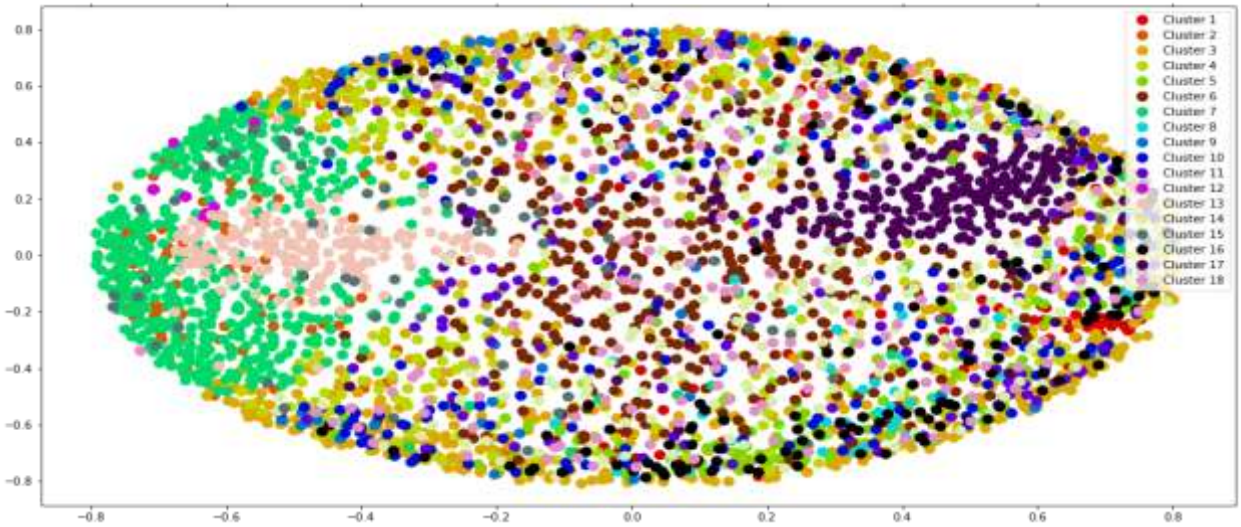
Cluster centroids mapped with top 10 words at centers from TFIDF

Cluster 1:	mark, trade, product, infringement, custom, registration, jemella, registrar, counterfeit, affidavit,
Cluster 2:	extradition, offence, tervonen, magistrate, surrender, bail, minister, rivera, warrant, finland,
Cluster 3:	company, liquidator, creditor, administrator, winding, asic, deed, debt, corporation, meeting,
Cluster 4:	appellant, magistrate, persecution, minister, information, tribunal, immigration, protection, fear, finding,
Cluster 5:	scheme, share, meeting, shareholder, company, member, asic, director, security, proxy,
Cluster 6:	document, privilege, discovery, advice, subpoena, production, professional, affidavit, disclosure, communication,
Cluster 7:	magistrate, minister, immigration, error, refugee, multicultural, migration, visa, protection, extension,
Cluster 8:	minister, visa, migration, immigration, magistrate, delegate, letter, appellant, error, multicultural,
Cluster 9:	native, area, land, determination, traditional, people, registrar, water, aboriginal, custom,
Cluster 10:	patent, infringement, product, specification, university, commissioner, patentee, cross, enantiomer, document,
Cluster 11:	bankruptcy, trustee, bankrupt, creditor, debtor, property, petition, sequestration, magistrate, debt,
Cluster 12:	defendant, carey, receiver, westpoint, asic, company, affidavit, demand, property, corporation,
Cluster 13:	employee, union, penalty, agreement, industrial, site, work, contravention, employer, workplace,
Cluster 14:	gong, falun, china, magistrate, practitioner, appellant, minister, information, protection, persecution,
Cluster 15:	income, commissioner, taxpayer, assessment, taxation, assessable, objection, itaa, penalty, deduction,
Cluster 16:	offer, croker, indemnity, calderbank, compromise, unreasonable, letter, imprudent, defendant, rejection,
Cluster 17:	service, injury, commission, medical, complaint, disability, employment, discrimination, pension, compensation,
Cluster 18:	motion, agreement, security, company, affidavit, contract, business, action, service, document,



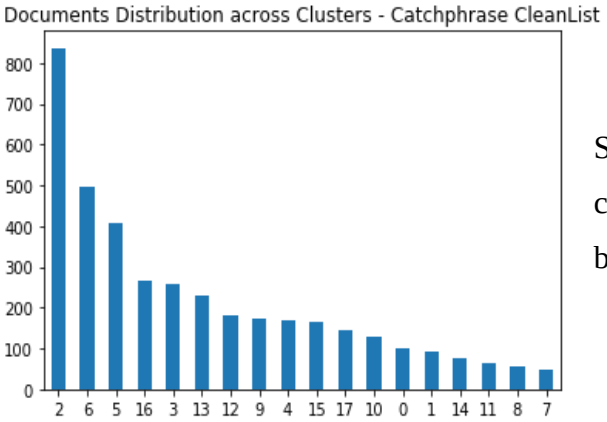
We noticed, after removing the insignificant terms, the clusters were sparser but we identified some new informative words filling in for the removed word which are more aligned towards legal terms.

4.5 Clustering using Catchphrases with TFIDF (without insignificant words)



Cluster centroids mapped with top 10 words at centers from TFIDF

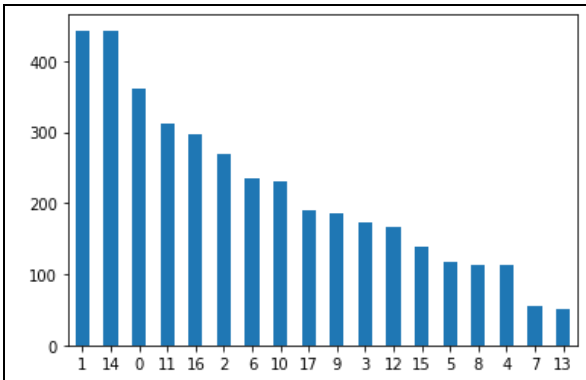
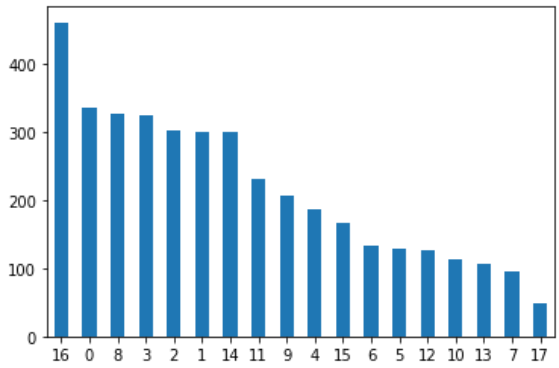
Cluster 1	income, taxation, assessment, taxpayer, commissioner, business, objection, benefit, scheme, assessable,
Cluster 2	industrial, penalty, relation, agreement, workplace, employee, union, breach, contravention, employment,
Cluster 3	magistrate, federal, migration, error, refugee, ground, interlocutory, jurisdictional, visa, leave,
Cluster 4	trade, practice, mark, misleading, representation, deceptive, contract, damage, respondent, contravention,
Cluster 5	bankruptcy, bankrupt, sequestration, order, creditor, petition, trustee, debtor, debt, magistrate,
Cluster 6	native, determination, claim, consent, claimant, applicant, registration, registrar, order, land,
Cluster 7	evidence, patent, applicant, respondent, injunction, practice, procedure, order, contract, claim,
Cluster 8	migration, visa, applicant, protection, minister, refugee, extension, procedural, ground, fairness,
Cluster 9	information, migration, refugee, visa, applicant, country, protection, failure, procedural, tribunal,
Cluster 10	consideration, order, corporation, procedure, practice, insolvency, federal, migration, pursuant, contention,
Cluster 11	corporation, company, liquidator, winding, scheme, creditor, order, arrangement, meeting, insolvency,
Cluster 12	error, migration, jurisdictional, refugee, protection, persecution, visa, finding, magistrate, failure,
Cluster 13	point, principle, migration, question, magistrate, federal, protection, visa, jurisdictional, error,
Cluster 14	security, cost, procedure, order, practice, applicant, respondent, discretion, asset, impecunious,
Cluster 15	administrative, compensation, applicant, entitlement, question, veteran, veterans, pension, injury, error,
Cluster 16	procedure, practice, order, discovery, federal, claim, interlocutory, motion, applicant, statement,
Cluster 17	cost, indemnity, order, applicant, respondent, offer, basis, successful, claim, award,
Cluster 18	privilege, legal, professional, document, advice, waiver, communication, discovery, claim, procedure



Similar observation was also captured in catchphrases. The insignificant terms were replaced by more unique terms.

4.6 Using Word2Vec matrix for clustering

In order to capture the semantics of our text corpuses, we tried using Word2Vev input matrix for clustering instead of using TFIDF. And Below is the distribution of documents across 18 clusters when we implemented k- means for “Sentences” corpus and “Catchphrases” corpus:

	
Kmeans word2Vec on Catchphrases corpus	**Kmeans word2Vec on Sentences corpus**

5. Evaluation of Results

Here we are performing two operations:

- Trying to fetch similar documents from documents corpus based on the keywords or text provided by a person (paralegal team member)

- Evaluate the accuracy of clustered form.

In order to implement first point we used, doc2Bow from “gensim”. To evaluate the similarity scored of documents so that it returns those documents which matches the most with the input example fed to the model.

In our code we gave example, where we are trying to fetch cases related to “cash liquification issues” and our model has returned the top 10 similar records with their respective similarity score.

For second point, evaluating accuracy of our clusters formed, we exported the results into an excel sheet which has the information of which document lies in which cluster group.

So ideally, one kind of documents should lie in one cluster only. Below is example of same “cash liquidity related issues” (and they all should lie in one cluster). Let’s see which model is giving us the best results:

file name	KMeans_Word2Vec_Sentences	KMeans_Word2Vec_catchphrase
06_1438.xml	0	0
06_1223.xml	8	0
06_1329.xml	8	6
06_314.xml	8	0
06_1222.xml	8	6
06_118.xml	8	6
06_887.xml	8	6
06_277.xml	8	6
06_555.xml	0	0
06_17.xml	8	6

From above we can evaluate accuracy of the clustering techniques implemented here.

When we used word2Vec k-means clustering over sentences corpus, it gave an error of 2 and assigned those 2 documents in different cluster.

And for word2Vec k-means clustering over catchphrases corpus, it gave an error of 4 and assigned those 4 documents in different cluster.

Kmeans clustering over Word2Vec of Sentences corpus is performing better and is able to distinguish our clusters more finely as compared to other models and also once keyword given as input to Doc2Bow model, it is able to fetch similar records from documents.

6. Future Work

We presented our approach to automatic legal text clustering in this paper, which clusters the legal documents into clusters based on how closely related the documents are. Automatic segregation of case law documents is very important in assisting common law legal research. Clustering large collections of documents remains a challenge, particularly in the legal field.

For future work, we can try below:

1. Soft cluster the documents, where a document can be assigned to multiple clusters. For every legal issue, identify the set of most important documents for that issue, and associate every document in our collections with one or more of these issues.

2. Topic Segmentation for the documents can be done by using topic modelling. With this we can build a hierarchical topical tree structure for set of documents.

3. This task was very time consuming and computationally heavy. Some of the code took hours (more than 13 hours) to get some results. Using Map-Reduce Framework to support distributed computing on very large data sets of clusters produces ability to tackle very large-scale document processing problems. This can also be considered in our future work resolve critical problems, including those of very large-scale document clustering and classification. We are looking into these subjects as another future research direction.

4. References

[1] Sahoo, Nachiketa, et al. "Incremental hierarchical clustering of text documents." Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, 2006

[2] Fung, Benjamin CM, Ke Wang, and Martin Ester. "Hierarchical document clustering using frequent itemsets." Proceedings of the 2003 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2003.

[3] Hossain, M. Shahriar, and Rafal A. Angryk. "Gdclust: A graph-based document clustering technique." Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007). IEEE, 2007

[4] Lu, Qiang, et al. "Legal document clustering with built-in topic segmentation." Proceedings of the 20th ACM international conference on Information and knowledge management. ACM, 2011.

[5] https://www.fedcourt.gov.au/about/national-court-framework/npas/all-npas

[6] Australasian Legal Information Institute, http://www.austlii.edu.au/

[7] Australian Acronyms and abbreviations https://www.pmc.gov.au/who-we-are/accountability-and-reporting/acronyms-and-abbreviations

[8] Q Lu, JG Conrad, K Al-Kofahi, W Keenan: Legal document clustering with built-in topic segmentation- Proceedings of the 20th ACM international conference on Information and knowledge management (2011)