DS5230 Unsupervised Machine Learning

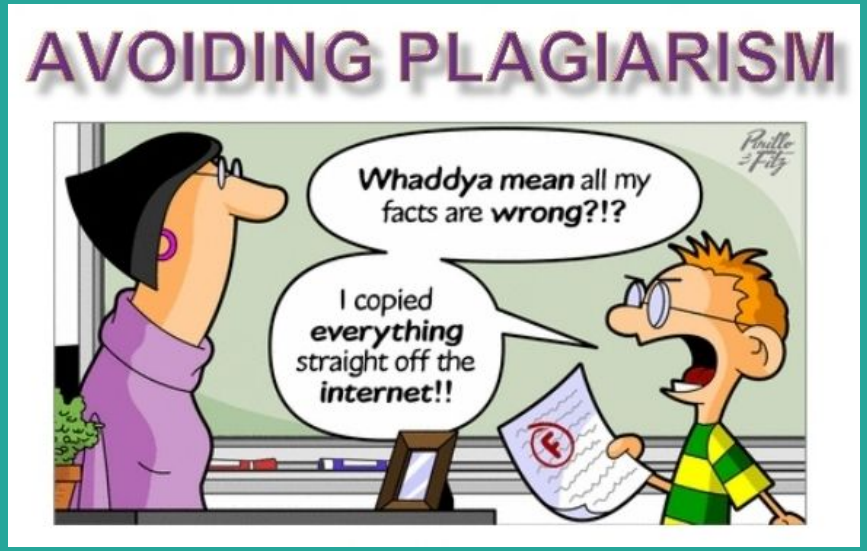
Plagiarism Detection

Team Members:
Aishwarya Vantipuli,
Spatika Krishnan,
Nandini Jampala

Plagiarism is the wrongful appropriation and stealing and publication of another author's language, thoughts, ideas, or expressions and the representation of them as one's own original work.

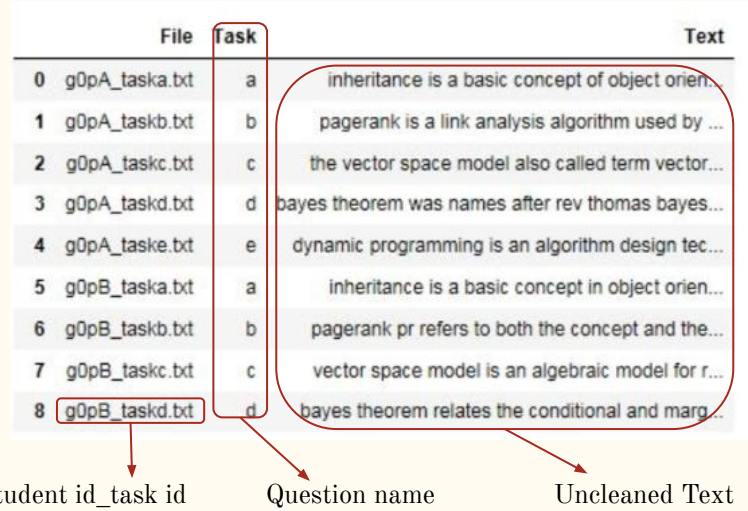
Business Application:

- Verifying Originality
- Tracking Content Misuse
- In-Depth Plagiarism Analysis



Dataset

Publicly available dataset
Corpus of Plagiarised Short Answers was chosen.



	File	Task	Text
0	g0pA_taska.txt	a	inheritance is a basic concept of object orien...
1	g0pA_taskb.txt	b	pagerank is a link analysis algorithm used by ...
2	g0pA_taskc.txt	c	the vector space model also called term vector...
3	g0pA_taskd.txt	d	bayes theorem was names after rev thomas bayes...
4	g0pA_taske.txt	e	dynamic programming is an algorithm design tec...
5	g0pB_taska.txt	a	inheritance is a basic concept in object orien...
6	g0pB_taskb.txt	b	pagerank pr refers to both the concept and the...
7	g0pB_taskc.txt	c	vector space model is an algebraic model for r...
8	g0pB_taskd.txt	d	bayes theorem relates the conditional and marg...

The corpus consists of answers to the following five short questions on a variety of topics in the Computer Science field:

a) What is inheritance in object-oriented programming?
b) Explain the PageRank algorithm that is used by the Google search engine.
c) Explain the Vector Space Model that is used for Information Retrieval
d) Explain Bayes Theorem from probability theory.
e) What is dynamic programming?

Each question has 19 students answers and 1 wikipedia answer

Dataset:

https://ir.shef.ac.uk/cloughie/resources/plagiarism_corpus.html

Goal of the Project:

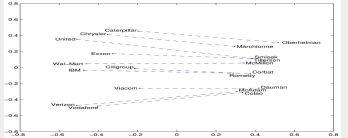
The goal of this project is to find the list of students with plagiarised work for from the student answers given as input.

For this we proposed two methods.

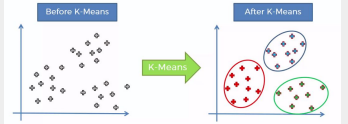
Method 1

1. Text Cleaning

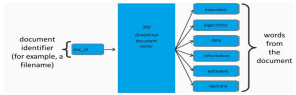
2. Glove Word Embeddings



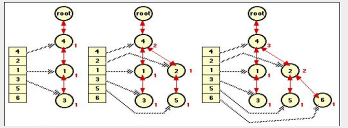
3. K-Means Word Clustering



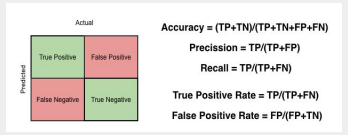
4. Document-level Vectorization



5. FP-Growth



6. Model Evaluation



Method 1

Text Pre-Processing:

- All punctuations and stop words are removed and sentences are tokenized using NLTK

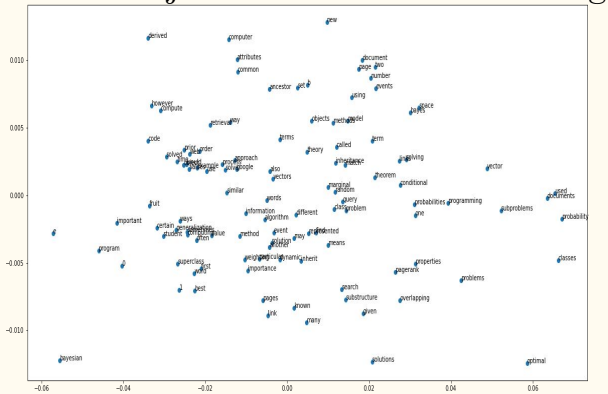
Glove Embeddings:

List of words in each sentence

```
['inheritance', 'basic', 'concept', 'objectoriented', 'programming', 'basic', 'idea', 'create', 'new', 'classes', 'add', 'extra', 'detail', 'exist
['done', 'allowing', 'new', 'classes', 'reuse', 'methods', 'variables', 'existing', 'classes', 'new', 'methods', 'classes', 'added', 'specialise',
['inheritance', 'models', '"iskindof"', 'relationship', 'entities', 'objects', 'example', 'postgraduates', 'undergraduates', 'kinds', 'student']
['break', 'problem', 'different', 'smaller', 'subproblems']
['pagerank', 'derived', 'theoretical', 'probability', 'value', 'logarithmic', 'scale', 'like', 'richter', 'scale']
['richard', 'bellman', 'originally', 'coined', 'term', '1940s', 'describe', 'method', 'solving', 'problems', 'one', 'needs', 'find', 'best', 'deci
```

GLOVE

PCA Projection of Word Embeddings



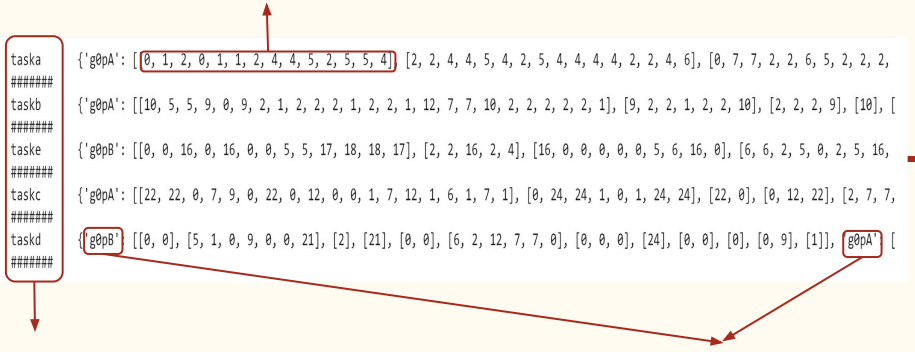
K-Means Clustering:

- Using the K-means algorithm, we cluster these word vectors into K clusters based on their contextual meaning. The output of this step is K word clusters.

Method 1

Document level Vectorization:

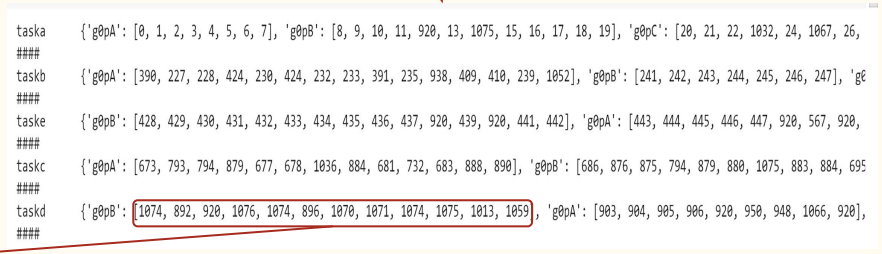
Sentence vector: each entry is a word representing cluster label it belongs to.



Types of questions in corpus.

Student id

Rule:
We form answer vector by assigning a unique number if two sentence vectors are different. If two sentence vectors are same we assign each of them the same number.



Document Vector:
each entry is sentence numbered according to the rule.

FP-GROWTH

Transactions	Itemset (Min sup > 3)
g0pB_taske	[686, 876, 875, 794, 879, 880]
g0pB_taskc	[241, 242, 243, 244, 245, 246, 247]
g0pA_taska	[0, 1, 2, 3, 4, 5, 6, 7]
g0pB_taska	[8, 9, 10, 11, 920, 13, 1075, 15, 16, 17, 18, 19]
g1pB_taskc	[489, 631, 632, 634, 635, 636, 1052]

100 Documents

Question	List of students with plagiarism
Task e	{'g1pB', 'orig', 'g3pB', 'g0pE', 'g4pB'}
Task a	{'g3pC', 'orig', 'g2pC', 'g0pE', 'g4pC'}
Task d	{'g2pA', 'g2pB', 'g0pB', 'orig', 'g3pA', 'g0pC', 'g1pA'}

Method-1

Model Evaluation:

- With Binary Classification, i.e, Heavily Plagiarised and Non Plagiarised the model yielded pretty good results.
- With Multi-Class Classification i.e, light, heavy, cut, non Plagiarised levels, Precision is high. This is mainly because of imbalanced classes

Binary Classification

Accuracy	79.03%
Precision	90%
Recall	53.55%
F1-Score	68.81%

Multi-Class Classification

Accuracy	61%
Precision	75%
Recall	45.5%
F1-Score	66.23%

Goal of the Project:

To find out the list of students who have plagiarised the tasks and to what extent.

Second Method consists of following steps...>

Method 2

1. Text Cleaning

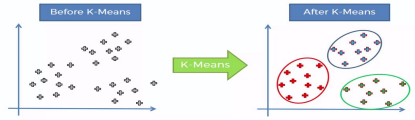
2. Tf-Idf Vectorization

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

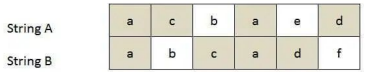
3. Document Similarity using cosine distance.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

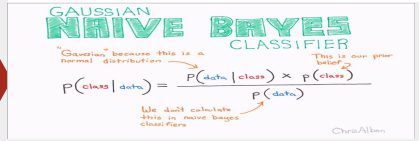
4. K-means clustering.



5. Containment and LCS scores

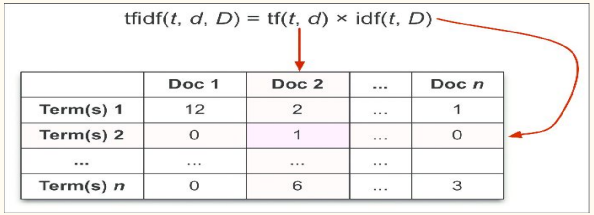


6. Model Evaluation



Tf-idf Vectorization

- First count word occurrences by document.
- Then apply the term frequency-inverse document frequency weighting.



- Words that occur frequently within a document but not frequently within the corpus receive a higher weighting as these words are assumed to contain more meaning in relation to the document

Tf-idf Output:

Vector form of g0pA_taska document

```
[[ 0.         0.         0.         0.         0.         0.         0.
   0.         0.         0.57501852 0.34113499 0.         0.         0.
   0.         0.         0.47152341 0.57501852 0.        ]
 [ 0.         0.         0.         0.40824829 0.40824829 0.
   0.40824829 0.         0.         0.         0.         0.         0.
   0.40824829 0.40824829 0.40824829 0.         0.         0.         0.        ]
 [ 0.         0.490779   0.490779   0.         0.         0.         0.
   0.         0.         0.         0.         0.         0.         0.
   0.         0.         0.490779   0.4024458  0.         0.3397724 ]
 [ 0.26396385 0.         0.         0.         0.         0.52792769
   0.         0.79189154 0.         0.         0.15659897 0.         0.
   0.         0.         0.         0.         0.         0.         0.        ]
 [ 0.         0.         0.         0.         0.         0.         0.
   0.         0.59430676 0.         0.35257792 0.59430676 0.         0.
   0.         0.         0.         0.         0.         0.41144595]
 [ 0.         0.         0.         0.         0.         0.         0.
   0.         0.         0.         0.26990725 0.         0.9099135
   0.         0.         0.         0.         0.
   0.31497221]]
```

Output of Cosine Similarity

	0	1	2	3	4	5	6	7	8	9	...	1
0	1.000000	1.000000	-0.001518	0.029564	0.029697	0.317527	0.000018	0.000018	0.000018	0.000018	...	-
1	1.000000	1.000000	-0.001518	0.029564	0.029697	0.317527	0.000018	0.000018	0.000018	0.000018	...	-
2	-0.001518	-0.001518	1.000000	0.040489	0.031902	0.042664	0.000094	0.000094	0.000094	0.000094	...	0
3	0.029564	0.029564	0.040489	1.000000	0.999642	0.081541	0.017496	0.017496	0.017496	0.017496	...	0
4	0.029697	0.029697	0.031902	0.999642	1.000000	0.077855	0.017598	0.017598	0.017598	0.017598	...	0
5	0.317527	0.317527	0.042664	0.081541	0.077855	1.000000	0.122909	0.122909	0.122909	0.122909	...	-
6	0.000018	0.000018	0.000094	0.017496	0.017598	0.122909	1.000000	1.000000	1.000000	1.000000	...	0
7	0.000018	0.000018	0.000094	0.017496	0.017598	0.122909	1.000000	1.000000	1.000000	1.000000	...	0
8	0.000018	0.000018	0.000094	0.017496	0.017598	0.122909	1.000000	1.000000	1.000000	1.000000	...	0
9	0.000018	0.000018	0.000094	0.017496	0.017598	0.122909	1.000000	1.000000	1.000000	1.000000	...	0

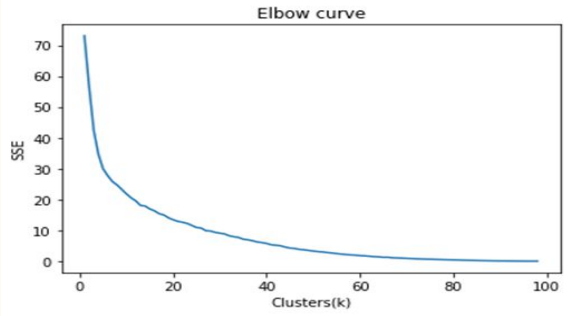
Cosine similarity:

- Cosine similarity is measured against the tf-idf matrix.It returns a 100*100 matrix that contains the distance measure of each document with every other document in our corpus.

K-means clustering

- Using the cosine similarity matrix, you can run a slew of clustering algorithms to better understand the hidden structure within the synopses.

Hyperparameter Selection Method for No.of Cluster (K)



Here the elbow curves at k=8 and hence our predetermined number of clusters is 8

Output of K-means clustering

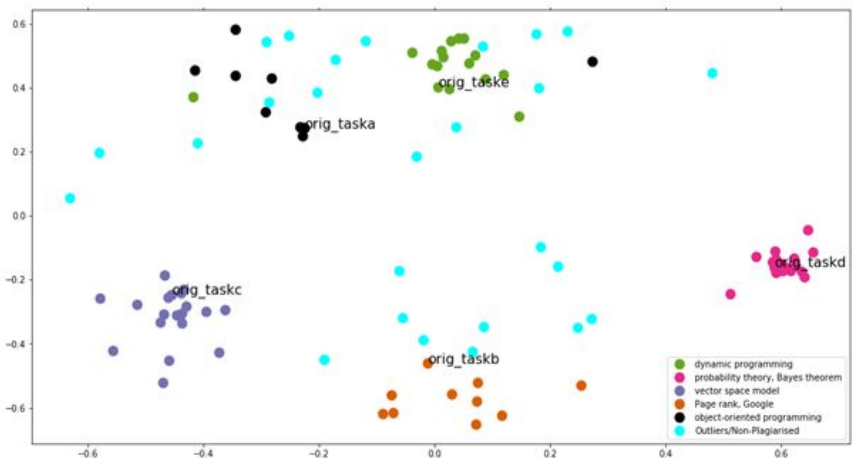
title	cluster
g0pA_taska	0
g0pA_taskb	4
g0pA_taskc	3
g0pA_taskd	1
g0pA_taske	6
g0pB_taska	6
g0pB_taskb	7
g0pB_taskc	3
g0pB_taskd	1
g0pB_taske	0
g0pC_taska	0
g0pC_taskb	5
g0pC_taskc	3
g0pC_taskd	1
g0pC_taske	0
g0pD_taska	0
g0pD_taskb	4
g0pD_taskc	3
g0pD_taskd	1
g0pD_taske	0
g0pE_taska	6
g0pE_taskb	4
g0pE_taskc	3
g0pE_taskd	1

Each Students answer is assigned a cluster label from 0-7.

The original(wikipedia answers) is also assigned a cluster.

orig_taska	6
orig_taskb	4
orig_taskc	3
orig_taskd	1
orig_taske	0

PCA to visualize document clusters

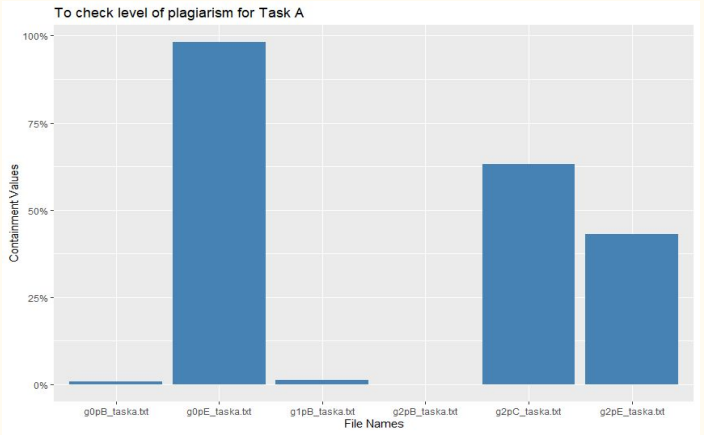


Now to classify the documents into the levels of plagiarism , we calculate the containment and LCS scores between the original document and the documents that share the same cluster as that.

Feature Extraction:

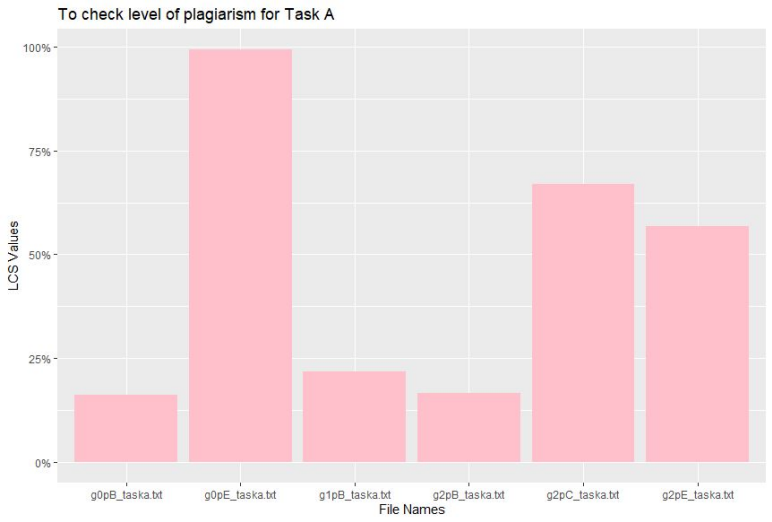
Containment:

For the specified student task, we compared it's n-gram count to the corresponding wikipedia n-gram count to calculate the containment values.



Least Common Subsequence:

The longest common subsequence is the longest string of words that are the same between the wikipedia task text and the student answer.



Classification Metrics :

Accuracy	92%
Precision	93%
Recall	92%
F1-Score	92%

Conclusion

- The accuracy of text models depends heavily on the preprocessing methods of the text data.
- In Method 1,using FP-Growth, we observed a high precision and accuracy scores of 90% and 79% respectively for Binary Classification.
- This behaviour fell as multi-level classification is implementation with a precision of 75%. This is mainly due to Class Imbalances.
- In Method 2, we observed a high precision as well as high accuracy scores of 93% and 92% respectively.
- We've experimented with k-means clustering and hierarchical clustering in both methods. K-means give better results as the other is not suitable for large datasets and Initial seeds have a strong impact on the final results thus giving us inaccurate cluster labels.