The following project compares two text mining clustering algorithms for grouping academic papers into different categories using the Python programming language.

1 Methodology

1.1 Baseline

In general terms, the following procedures were performed in this approach: preprocessing, stemming, tf-idf data transformation and K-means clustering.

1.1.1 Preprocessing

The simple preprocessing steps were as follows:

1) The abstract and title of each paper were merged.
2) The texts were converted to lowercase.
3) Punctuation was removed.
4) Chinese characters were deleted.
5) Words associated with url (words beginning with www, url, http, https) were removed.
6) Digits were deleted.

In addition, stopwords were eliminated using the Sklearn library. The python line is as follows

from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS

The final words obtained were stemmed using the *Snowball Stemmer* from the NLTK library. This stemmer is similar to the Porter Stemmer, which is an algorithm developed by Martin Porter in 1980 that provides a smooth stemming. Snowball stemming is more aggressive, however it presents the difficulties inherent in the stemming of over and under stemming resulting in meaningless words.

1.1.2 TF-IDF Data transformation

For this step two approaches were used, the basic form presented in the course slides, and the SKlearn library using the l2 norm.

For the former, the *CountVectorizer* function was used to obtain the frequencies of the words in each of the documents. In this function, a minimum threshold of two common words was established, i.e. the word was only counted if it was common in at least two documents. With these frequency data per document, the following equation was applied:

$$tfidf(w_j, d_i) = f(w_j|d_i) * log(\frac{n}{n_j}))$$

where w is the word d is the document,$f(w_j|d_i)$ is the frequency of the word j given document i and n_j is the number of documents were the word j appear. The frequency was normalized using the number of words in document i.

For the second option, the *TfidfVectorizer* function from Sklearn. package was used with the l2 norm parameter.

1.1.3 Clustering and evaluation

Simple K-means clustering from Sklearn package was used, using 5 as the number of clusters.

Additionally, to verify the performance of our clustering algorithm with the groundtruth classification, the normalized mutual info score was calculated using this formula:

$$NMI = \frac{I(C, D)}{\sqrt{H(C)H(D)}}$$

This formula is similar to the $normalized\_mutual\_info\_score$ function from the Sklearn library using the average method as geometric.

1.2 Proposed approach

As many of the steps are similar to the baseline approach, we will now focus on the changes that were made with respect to the previous method.

1.2.1 Preprocessing

In addition to the Snowball stemmer, a **Lemmatizer** was used, which in theory gives better results in word grouping. Furthermore, dimensionality reduction was used after using the TF-IDF transformation. **Principal Component Analysis (PCA)** from the SKLearn library was used with a number of components that would explain 90% of the data variance.

1.2.2 Clustering

Spectral Clustering was employed from the *sklearn.cluster* library and the gamma parameter, which represents the kernel coefficient, was modified.

2 Results

2.0.1 NMI

The following are the results of the experiments performed with each of the different types of clustering: K-means and Spectral.

	Spectral Clustering								K- means Clustering			
	Stemming				Lemmatizer				Stemming		Lemmatizer	
	γ=1		γ=0.01		γ=1		γ=0.01					
	Basic tf-idf	tf-idf Sklearn	Basic tf-idf	tf-idf Sklearn	Basic tf-idf	tf-idf Sklearn	Basic tf-idf	tf-idf Sklearn	Basic tf-idf	tf-idf Sklearn	Basic tf-idf	tf-idf Sklearn
NMI	0.632	0.6819	0.6288	0.7109	0.5934	0.6109	0.6303	**0.7123**	0.473	**0.726**	0.2973	0.6533

Figure 1: NMI results for each of the method combinations

From the figure above we can deduce that stemming gives a better performance for K-means clustering while lemmatizing gives a better result for Spectral Clustering. Also, the tf-idf representation provided by the Sklearn library gives consistently better results in all methods and configurations. As for Spectral Clustering, a gamma value of 0.01 provides better results than one of value 1. Finally, the best method under the NMI guidelines is K-means clustering using stemming and Sklearn's tf-idf representation. Although the difference with Spectral Clustering is not very significant.

2.0.2 Topics

To determine the topics of each classification we used the 10 most common words of each label of the best configuration in the k-means and Spectral Clustering algorithms. Here are the results

Spectral Clustering:
1: program, language, compiler, code, paper, based, time, application, memory, algorithm
2: quantum, security, data, key, scheme, proposed, cryptography, protocol, encryption, attack
3: robotic, control, model, using, task, robot, result, environment, approach, soft
4: database, data, query, information, paper, ontology, relational, model, sql
5: proposed, detection, model, learning, data, vision, using, image, method, algorithm

K-means Clustering:
1: image, method, use, proposed, detect, model, computer, learn, perform
2: result, perform, model, task, control, robot, use, learn, base
3: key, security, encryption, proposed, data, use, scheme, cryptography, protocol
4: database, data, relational, use, query, model, inform, approach, process.
5: program, computation, compile, code, base, language, optimum, implement, quantum

Interestingly, it is easier to determine the subject by spectral clustering and its words than by K-means. Maybe it is due to the use of lemmatizer instead of stemming. Based on the keywords obtained by these two clustering algorithms, the topics are: **programming languages, computer vision, relational databases, robotics, and security & cryptography**.

Execution instructions:

To run K-means clustering run the file kmeans.py , if there is error download NTLK and follow the instructions provided in the error message. To run the spectral clustering algorithm run the spectral.py file, if there is an error download NTLK and follow the instructions provided in the error message.