Universite Jean Monnet

Faculte des Sciences et Techniques

Data Mining Project Report

Recommendation System using Unsupervised Machine learning and Community Detection

Masters Machine Learning & Data Mining

Authors:
Gunasekara Jayani
Ezukwoke Kenneth
Ekpo Etienne
Orhan Solak

Professor:
Christine Largeron

January 2020

Contents

1 Introduction **2**

1.1 Objectives . . . 2

1.2 Related Works . . . 2

2 Text Mining **3**

2.1 Dataset . . . 3

2.2 Text Preprocessing . . . 4

3 Methodology **4**

3.1 Clustering with PCA . . . 4

3.1.1 Principal Component Analysis . . . 4

3.1.2 Kernel Principal Component Analysis (kernel-PCA) . . . 6

3.1.3 Partition based clustering . . . 9

3.1.4 KMEANS . . . 9

3.1.5 Kernel KMEANS . . . 9

3.1.6 Kernels . . . 11

3.2 Recommendation System using Community Detection . . . 11

3.2.1 Network Preprocessing & Visualization . . . 12

3.2.2 Network Analysis . . . 13

3.2.3 Community Detection . . . 13

3.2.4 Candidate Selection . . . 15

3.2.5 Candidate Ranking . . . 16

4 Evaluation and Result **16**

4.1 Evaluation for Clustering . . . 16

4.1.1 Evaluation metric . . . 17

4.1.2 T-SNE Visualization . . . 17

4.1.3 Result and Demo . . . 18

4.2 Evaluation for Graph Modeling . . . 21

4.2.1 Evaluation metric . . . 21

4.2.2 Result and Demo . . . 22

5 Conclusion **24**

1 Introduction

This paper presents an approach for a recommendation system to retrieve the similar research papers related to the same topic and content published on the famous conference EGC. In each year, the conference presents only limited number of selected research papers. In addition, the submissions and the authors are not clearly interacted, that is the authors do not know about the person who had been evaluated their papers. Therefore, a recommendation system can easily sort out the problem as it returns the appropriate answer within milliseconds. Today, the recommendation systems are widely used in various fields. This system allows users to discover the similar type of research papers according to their interest. The project source code can be found on the github link at the footer[1].

First, we pre-processed the data rigorously with various text mining techniques and detected the similarities between documents. Section 2.2 explains about the data pre-processing through tokenization, lemmatization and stemmatization. Then the communities detected using social mining and network analysing from the extracted content of the research papers. The pre-processed, text mined data are unsupervised as there is no classification labels already defined. Section 3 explained the network and extracting ranks out of the detected communities. We discussed the experiments and results in section 4 with the demo followed by the evaluation of graph modelling. Finally we discuss about the conclusions and perspectives.

1.1 Objectives

The objective of this research is to suggest the similar research papers to the users in a manner of sorted ranking list.

1.2 Related Works

Scienstein, A Research Paper Recommender System [1] is an enhanced version of existing recommendation system, explains about the similarity index and in text impact factor

[1]Complete source code for graph mining for recommender system github

which are novel approaches. By using the content based and collaborative filtering, this system purposes a hybrid recommendation system.

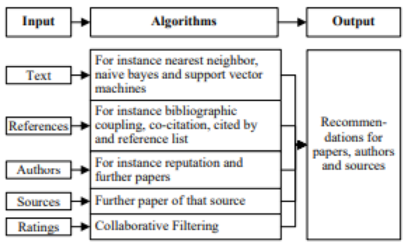

Input
Algorithms
Output
Text
References
Authors
Sources
Ratings
For instance nearest neighbor, naive bayes and support vector machines
For instance bibliographic coupling, co-citation, cited by and reference list
For instance reputation and further papers
Further paper of that source
Collaborative Filtering
Recommen-dations for papers, authors and sources

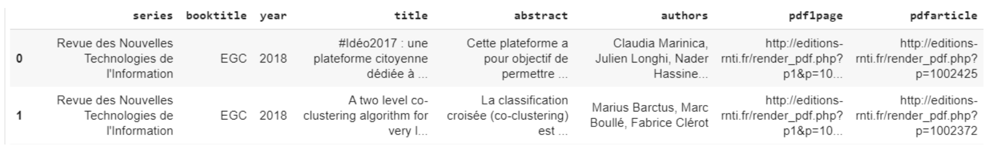
Figure 1: Hybrid recommender system [5]

Moreover, the paper reveals about citation analysis, implicit, explicit ratings, author analysis and source analysis. At the instances when there are more items than users, the researchers believe that collaborative filtering is not beneficial in terms of domain [4]. In text citation frequency analysis, text impact factor was determined in order to represent the number of citations that are linking to particular document in terms of overall citations [5].

2 Text Mining

2.1 Dataset

For the analysis we used the dataset export_articles_EGC_2004_2018.csv which contains only the published research papers by EGC in year 2004.There are 1269 samples with 8 features such as series, booktitle, year, title, abstract, authors, pdf1page, pdfarticle. The diverse of the dataset lay out into different categories and languages(French and English)



	series	booktitle	year	title	abstract	authors	pdf1page	pdfarticle
0	Revue des Nouvelles Technologies de l'Information	EGC	2018	#Idéo2017 : une plateforme citoyenne dédiée à ...	Cette plateforme a pour objectif de permettre ...	Claudia Marinica, Julien Longhi, Nader Hassine...	http://editions-rnti.fr/render_pdf.php?p1&p=10...	http://editions-rnti.fr/render_pdf.php?p=1002425
1	Revue des Nouvelles Technologies de l'Information	EGC	2018	A two level co-clustering algorithm for very l...	La classification croisée (co-clustering) est ...	Marius Barctus, Marc Boullé, Fabrice Clérot	http://editions-rnti.fr/render_pdf.php?p1&p=10...	http://editions-rnti.fr/render_pdf.php?p=1002372

Figure 2: Dataset

2.2 Text Preprocessing

Initially we preprocessed the natural language data into machine readable format. We used basic preprocessing steps such as stopword removal, removal of symbols, punctuation and signs inconsistent with meaningful language sentences followed by tokenization, lemmatization and stemmatization.

- Tokenization: this is the process of converting the words in a sentence into independent words or tokens. Here, each of the unique word was assigned a number and this stage followed after stopword removal.
- Lemmatization: is the process of reducing a word to its most simplest form and identifying the actual content. This may be converting a plural to its simplest noun.
- stemmatization: is the process of removing affixes.

3 Methodology

3.1 Clustering with PCA

Clustering is the task of gathering samples into groups of similar samples according to some predefined similarity and they find applications in data compression, data summurization for recommender systems, similarity grouping of web search result, stock indices and customer profiles. PCA (Principal Component Analysis) is a dimensionality reduction technique in machine learning and also used for feature selection . The method used in the paper focuses on non-linear dimensionality reduction for non-linear clustering. By introducing the concept of non-linearity in our model, we achieve better clustering result on the documents.

3.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised dimension reduction technique that depends on the orthogonal transformation of a higher dimensional data space to a lower dimensional subspace [6] . An idea originally proposed by Karl Pearson [7] as a statistical method to find lines or planes of best fit in the context of regression. In this paper we use PCA as a preprocessing procedure to reduce the dimension of the text before building a clustering model. PCA improves the time complexity of the clustering algorithm by reducing the dimension of the similarity matrix.

Classical PCA algorithm aims at finding a linear subspace of lower dimension than the original space (space of cosine similarity matrix). The PCA algorithm is described below in step-wise order. We use $\sum$ to denote the **covariance**, which is different from summation $\sum_{i=i}^{N}$ symbol.

We perform the classical PCA technique as follows. Given a dataset $x_i \in \mathbb{R}^D$. We proceed by centering the dataset around its mean value, we do this by taking the mean.

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

$$\mathbf{dx_i} = x_i - \hat{x}_i \tag{2}$$

We then calculate the covariance matrix thus

$$\sum = \frac{1}{N}\sum_{i=1}^{N} dx_i dx_i^T \tag{3}$$

After which we compute the eigenvectors $\mathbf{u_k}$ corresponding the individual eigenvalues λ_k using

$$\sum \mathbf{u_k} = \lambda \mathbf{u_k} \tag{4}$$

We then sort the eigenvectors according to the corresponding eigenvalues in decreasing order and return the transformed data based on the numbers of components k. So that

$$\hat{x} = \mathbf{dx}^T \mathbf{u_k} \tag{5}$$

We summarize the PCA procedure succinctly

Algorithm 1: PCA procedure

```
Input  : Similarity matrix $x \in \mathcal{X}$ where $x \in \mathbb{R}^D$
Output: $\hat{x} \in \mathbb{R}^k$ where $k \ll D$
1 begin
2     $\hat{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$;
3     $dx_i = x_i - \hat{x}$;
4     $\sum = \frac{1}{N}\sum_{i=1}^{N} dx_i dx_i^T$;
5     $\sum \mathbf{u}_k = \lambda_k \mathbf{u}_k$;
6     $\mathbf{u}_k = \arg\mathrm{sort}(\mathbf{u}_k)$;
7     $\hat{x} = \mathbf{u}_k \cdot dx_i$;
8 end
```

3.1.2 Kernel Principal Component Analysis (kernel-PCA)

The Kernel PCA seeks to address the limitation of the classical PCA algorithm by generalizing to non-linear dimensionality reduction [8]. By projecting the feature space x_i into a kernel space $\phi(x_i)$, the kernelized version is able to capture efficiently a subspace that reduces the dimension of the original feature space. We capitalize on this to effective learn a space where the data is independently and identically distributed.

To construct the solution of the kernel-PCA we begin like the classical approach, except in this case we assume the new feature space have **zero mean**, such that

$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i) = 0 \tag{6}$$

We then compute the covariance matrix as

$$\sum = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T \tag{7}$$

The eigenvalue and corresponding eigenvectors is given by

$$\sum \mathbf{u_i} = \lambda_k \mathbf{u_i} \quad \forall i = 1, \dots, k \tag{8}$$

We then substitute equation (7) into (8),

$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\{\phi(x_i)^T\mathbf{u_k}\} = \lambda_k \mathbf{u_k} \tag{9}$$

Hence,

$$\mathbf{u_k} = \sum_{i=1}^{N} a_{ki}\phi(x_i) \tag{10}$$

where

$$a_{ki} = \frac{1}{\lambda_k N}\phi(x_i)^T\mathbf{u_k} \tag{11}$$

Following **Mercer's** theory for kernels, we know that kernel κ takes the form

$$\kappa(\mathbf{x_i}, \mathbf{x_j}) = \phi(x_i)^T\phi(x_j) \tag{12}$$

Subsitute equation (10) into equation (9) and multiply both sides of equation by $\phi(x_l)$

$$\frac{1}{N}\sum_{i=1}^{N}\phi(x_l)\phi(x_i)\sum_{j=1}^{N}a_{kj}\phi(x_i)\phi(x_j) =$$
$$\lambda_k a_{ki}\phi(x_l)\phi(x_i)$$

Which we can re-writen as

$$\frac{1}{N}\kappa(x_l, x_i)\sum_{i=1}^{N}a_{kj}\kappa(x_i, x_j) = \lambda_k\sum_{i=1}^{N}\kappa(x_l, x_i) \tag{13}$$

$$\mathbf{K^2 a_k} = \lambda_k N\mathbf{a_k} \tag{14}$$

where

$$\mathbf{K} = \kappa(x_i, x_j) \tag{15}$$

$\mathbf{a_k}$ is the eigen vector which is an N-dimensional column vectors of a_{ki}. $\mathbf{a_k}$ can be solved from

$$\mathbf{K a_k} = \lambda_k N\mathbf{a_k} \tag{16}$$

The resulting kernel principal components transformation is

$$\hat{x} = \phi(\mathbf{x})^T\mathbf{u_k} = \sum_{i=1}^{N}a_{ki}\kappa(\mathbf{x}, \mathbf{x}_i) \tag{17}$$

Given an uncentered kernel matrix, we compute the zero mean of the kernel [8] thus

Let

$$\phi(x_i) = \phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j) \tag{18}$$

$$\hat{\mathbf{K}} = \left\| \phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j) \right\|_2 \tag{19}$$

$$= \left(\phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j)\right)^T \left(\phi(x_i) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j)\right)$$

After expansion we have that

$$= K_{ij} - \sum_{i=1}^{N}\phi(x_i)^T\phi(x_j)\frac{1}{N} - \frac{1}{N}\sum_{i=1}^{N}\phi(x_j)^T\phi(x_i) - \frac{1}{N}\sum_{i=1}^{N}\phi(x_j)\phi(x_j)$$

This can be rewritten in short as

$$\hat{\mathbf{K}} = \mathbf{K} - 1_{1/N}\mathbf{K} - \mathbf{K}1_{1/N} + 1_{1/N}\mathbf{K}1_{1/N} \tag{20}$$

Where $\mathbf{K} = \mathbf{K}_{ij}$. $\hat{\mathbf{K}}$ is called the Gram matrix (normalized kernel matrix).

We summarize the precudure for kernel-PCA as follows

Algorithm 2: Kernel PCA Algorithm

```
   Input  : $\phi(x) \in \kappa(x_, x_j)$ where $\phi(x) \in \mathbb{R}^D$. Let $\mathbf{K} = \kappa(x_i, x_j)$
   Output: $\hat{x} \in \mathbb{R}^k$ where $k \ll D$
1  begin
2      Select a kernel $\kappa$;
3      Construct Gram matrix $\hat{\mathbf{K}} = \mathbf{K} - 1_{1/N}\mathbf{K} - \mathbf{K}1_{1/N} + 1_{1/N}\mathbf{K}1_{1/N}$;
4      Solve eigen problem $\mathbf{K}\mathbf{u_k} = \lambda\mathbf{u_k}$;
5      Project data in new space $\hat{x} = \sum_{i=1}^{N}\mathbf{u_k}\kappa$;
6  end
```

The kernel PCA algorithm is expressed only in terms of dot products, this trick allows us to construct kernels only from training data $\{\mathbf{x_i}\}$ [9].

3.1.3 Partition based clustering

Clustering is the task of gathering samples into groups of similar samples according to some predefined similarity and they find applications in data compression, data summarization for recommender systems, similarity grouping of web search result, stock indices and customer profiles.

KMeans falls under the clustering category called **partition clustering**. This approach used a technique of splitting the datasets into subgroups of $k - clusters$ and iteratively tries to find the best $k - cluster$ that best explains the partition of a data.

3.1.4 KMEANS

The objective of traditional clustering methods is to partition training vectors by using similarity criteria applied on the basis of Euclidean metrics. More precisely, the Euclidean distance or inner product is used to measure the similarity between the training vectors in the original vector space, $\{x_i, i = 1, \ldots, N\}$. The objective function we try to minimize in KMeans is give by

$$argmin_{w_k} \left\{ \sum_{k=1} K \sum_{X_t \in C_k} \|x_t - \mu_k\|^2 \right\} \tag{21}$$

Where μ_k denotes the mean of k *th* cluster's centroid. In an optimal K-means solution, the centroid, say μ_k is associated with a training vector x_j that yields the minimum distance among all the centroids.

$$\mu_k = \frac{1}{C_k} \sum_{i \in C_k} x_i \tag{22}$$

Lloyd algorithm is one of the most well used classical kmeans algorithm [10].

3.1.5 Kernel KMEANS

Kernel Kmeans is a non-linear version of the classical kmeans algorithm. Several method have been proposed and used over the years like the spectral version [11]. The data x_i

Algorithm 3: Classical K-Means (Lloyd's) algorithm

```
   Input  : $X, \mu^r$ random center initialization
   Output: $\mu_k \in C_k$
1  begin
2      while not converged do
3          $C_1, \ldots, C_k \leftarrow \phi$;
4          for $i \in 1, \ldots, n$ do
5              $\arg\min \sum_{c=1}^{k} \sum_{x_i \in C_k} ||x_i - \mu_k||^2$
6          end
7          for $j \in 1, \ldots, k$ do
8              $\mu_j \leftarrow \frac{1}{C_k} \sum_{i \in C_j} x_i$
9          end
10     end
11 end
```

is projected into a higher dimensional subspace $\phi(x_i)$ where we are only concerned in the dot products of the feature vectors. The distance in the new space becomes

$$\|\phi(x_i) - \phi(\mu_j)\|^2 \tag{23}$$

Where

$$\phi(\mu_k) = \frac{1}{C_k} \sum_{i \in C_k} \phi(x_i) \tag{24}$$

By expanding the above equation we have

$$\kappa(x_i, x_i) + \kappa(\mu_j, \mu_j) - 2\kappa(x_i, \mu_j) \tag{25}$$

Algorithm 4: Kernel K-Means clustering algorithm

```
   Input  : $\kappa$
   Output: $C_k$
1  begin
2      while not converged do
3          Compute the distance of each point in $\phi(x)$ from center $\mu$
            $\arg\min \sum_{c=1}^{k} \sum_{x_i \in C_k} ||\phi(x_i) - \phi(\mu_j)||^2$
4      end
5  end
```

3.1.6 Kernels

The kernels used in this project are chosen appropriately to accurately enhance the clustering result. They are listed below:

- **Linear kernel**

$$\kappa(x_i, x_j) = \mathbf{x}_i \mathbf{x}_j^T \tag{26}$$

- **Sigmoid kernel**

$$\kappa(x_i, x_j) = \tanh(\gamma \mathbf{x}_i \mathbf{x}_j^T + c) \tag{27}$$

  where $c \geq 0$ and $\gamma = \frac{1}{2\sigma^2}$.

- **Polynomial kernel**

$$\kappa(x_i, x_j) = (\mathbf{x}_i \mathbf{x}_j^T + c)^d \tag{28}$$

  where $c \geq 0$ and d is the degree of the polynomial usually greater than 2.

- **Cosine kernel**

$$\kappa(x_i, x_j) = \frac{\mathbf{x}_i \mathbf{x}_j^T}{||\mathbf{x}_i||||\mathbf{x}_j||} \tag{29}$$

3.2 Recommendation System using Community Detection

In this section, we aim at implementing an algorithm that would efficiently recommend new articles to a user $\boldsymbol{X}$ having that $\boldsymbol{X}$ already read a set of articles or journals from the given document collection or dataset.

Based on the similarity metrics acquired after the text processing step in the section discussed above, we modelled our dataset as a graph in which each nodes represent a document uniquely identified by its ID while the resultant pairwise similarity measures represent the weighted edges that link any two nodes/documents in the given collection. In this sense, two(2) documents A and B with cosine similarity 0.4 in our dataset will be depicted as two(2) nodes (ID_A, ID_B) in our Graph linked each other by an edge of weight 0.4. It will be also considered that A and B are not linked if the weight or Similarity measure is zero(0). All other associated information about the document such as booktitle, year, authors and many more are set as attributes or properties of the node.

Using Python Graph mining framework "**networkx**"(https://pypi.org/project/networkx/) and python community detection library "**community**" (https://pypi.org/project/community/), we built our graph structure, visualized and analysed the network before detecting and evaluating all its underlying communities. Then based upon the articles already read by the end user X, we select the best N candidates out of those communities and ranked them adequately in few seconds. All those sub-tasks will be further explained in the sections to come.

Also, it is worth noting that most implementations were conducted on **GoogleColab Cloud Servers** (https://colab.research.google.com/) where we were freely provided with 12 GB RAM of processing power for this project. This environment tremendously eased collaboration on the same notebook as well as experimentation of several Community Detection Algorithms on our graph.

3.2.1 Network Preprocessing & Visualization

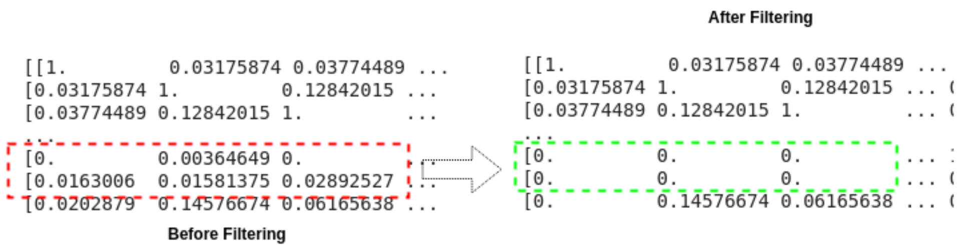
Upon building our graph, we realized that it was approximately fully connected based on the initial assumption that consisted in using the similarity metrics as the edges weights. To induce some sparsity to the graph we added a filter with a threshold (T) set to 0.03 such that all weights lesser than T are set to 0.

$$weight_{after} = [weight_{before} < T?0 : weight_{before}]$$

We tuned the value of the threshold (T) with respect to the density of the overall graph (≈ 0.6).

The screenshot below shows the upper left section of the adjacency matrix before and after applying of the filter.

Once the filter applied, we visualized the Graph using both Networkx and Matplotlib



```
[[1.         0.03175874 0.03774489 ...
 [0.03175874 1.         0.12842015 ...
 [0.03774489 0.12842015 1.         ...
 ...
 [0.         0.00364649 0.         ...
 [0.0163006  0.01581375 0.02892527 ...
 [0.0202879  0.14576674 0.06165638 ...
```

```
[[1.         0.03175874 0.03774489 ...
 [0.03175874 1.         0.12842015 ...
 [0.03774489 0.12842015 1.         ...
 ...
 [0.         0.         0.         ...
 [0.         0.         0.         ...
 [0.         0.14576674 0.06165638 ...
```

Figure 3: Section Adjacency matrix before and after Filtering)

Libraries as shown in the figure below

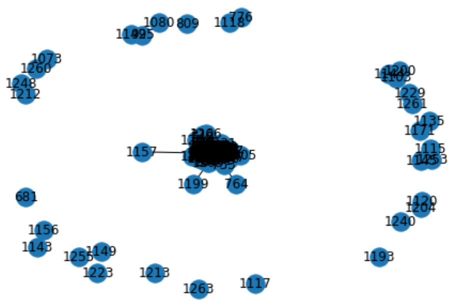


Figure 4:   Graph Visualization

3.2.2 Network Analysis

The above Graph G is composed of 1269 nodes interconnected by 490 158 edges with a Graph Density of 0.61. As discussed in the previous section, the threshold has been tuned to maintain the level of density for the overall network. Also, while the max Degree is 1194 , the average/Mean degree is 773

3.2.3 Community Detection

Here we explored main(5) main community detection algorithms on our Graph *G*.

- **Greedy approach** (Modularity Maximization)

  This algorithm is based on the Clauset-Newman-Moore greedy modularity maximization.It begins with each node in its own community and joins the pair of communities that most increases modularity until no such pair exists [15] [16].

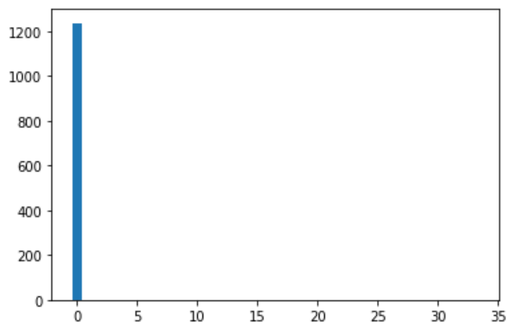


Figure 5: Bar Chart of distribution of number of nodes per clusters (Greedy)

Upon applying this algorithm on our Graph we observed 34 communities as shown in the graph above. While the majority of the nodes were clustered in only (1)one community, 33 others remained on their own communities.

- **Louvain Algorithm**: (Modularity Maximization)

  This community detection algorithm seeks to maximize the cluster modularity in an agglomerative manner. In the first instance, each node is assigned to its own network, then iteratively it attempts to join any two(2) communities such that the modularity is maximized.

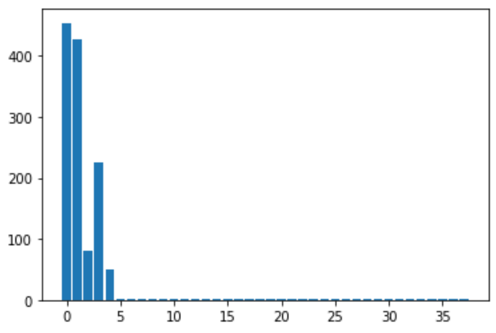


Figure 6: Bar Chart of distribution of number of nodes per clusters (Louvain)

Upon applying this algorithm on our Graph we observed 38 communities as shown in the graph above. While five communities seems to have relatively enough nodes, we recorded 33 communities with one node each.

- **Clique Based Approach** This algorithm find overlapping communities within the graph based on the assumption that a community is a complete subgraph of k nodes[13]. Though quite relevant in finding overlapping communities, this algorithm is computationally expensive while evaluating the completeness of k-graph. While applying this algorithm on our Graph G, we run shot of **12GB RAM** virtual machine.

- **Girvan-Newman algorithm**

  This algorithm detects community within Graph using edge betweeness centrality measure. [13]

Algorithm 5: Girvan-Newman algorithm

```
1. Compute betweenness for all edges in the network
2. Remove the edge with highest betweenness
3. Go to step 1 until no edges left
```

Upon applying our implementation of Edge betweeness (wrapper on Community Library) on the graph, we encountered several sessions timeout on our virtual Environment. This informed us on the time complexity constraint or limitation of this algorithm as the number of nodes become significant.

Due to the various time and space constraints related with the explored community Detection algorithms, we selected the Louvain Algorithm for our recommendation task. It recorded 5 clusters of 247 nodes on average with 33 isolated communities (one node per community).

3.2.4 Candidate Selection

Having detected the various community and taking into account the articles read by the User, we implemented a candidate selection algorithm that seeks to retrieve N number of candidates from the various communities according to a computed distribution.

..

Algorithm 6: Candidate Selection algorithm

```
1 Retrieve the community assigned to each article read by the user.
2 Compute the distribution for each community
3 Randomly select N amount of candidate in each community per the computed
    distribution
4 Return N Candidate
```

3.2.5 Candidate Ranking

Once the best N candidates have been selected from the various communities, we combine the list of articles read with the those N candidate and create a subgraph H. We then compute a pairwise edge betweeness score for every node. This score is then used for ranking the candidates before

Algorithm 7: Candidate Ranking algorithm

```
1 Build a subgraph H including all articles read by user and best candidate
    returned.
2 Compute edge betweeness centrality score for each node in H.
3 Sort candidates according to their node betweeness score
4 Return N ordered Candidates
```

..

4 Evaluation and Result

4.1 Evaluation for Clustering

Evaluation clustering result is unlike the supervised approach where we compare classification prediction to original class. In Clustering however, the use measures such as the **Silhoutte score** and **Elbow** method to get the optimal value of k to cluster the documents. In contrast to Graph mining where **Modularity** is used to measure the accuracy of communities belonging to perculiar cliques, Clustering uses an iterative method to find a k value that gives the minimum sum of squared error.

$$\arg\min \|x - \hat{x}\|_2^2 \tag{30}$$

4.1.1 Evaluation metric

In this paper we used the **Elbow method** to find the optimal k-value required to accurately cluster the data into their respective classes.

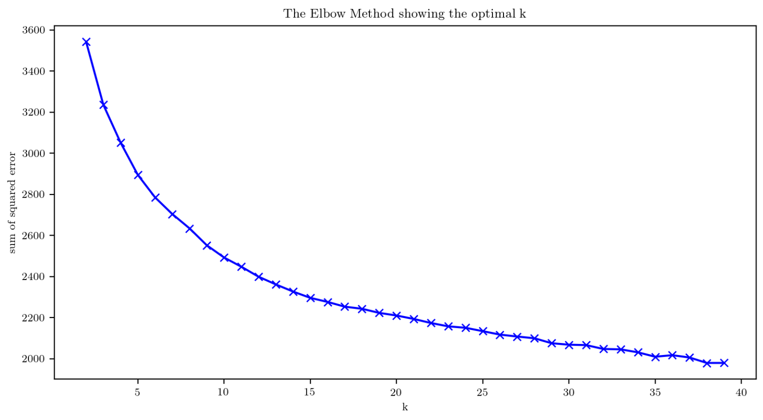


Figure 7: Elbow method for finding optimal k

We then choose the k value corresponding to the point where sum of squared error is reasonably unchanged. This point is called the **Elbow** point. We achieve a near optimum value of between 33 and 37 for the k-value as seen in Table 9.

4.1.2 T-SNE Visualization

T-Distributed Stochastic Neighbor Embedding (T-SNE) is a machine learning technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets [17, 18]. In this paper we used this technique to visulalize our data in a web application. Using T-SNE is advantageous since its stochastically align highly related data points into similar cluster.

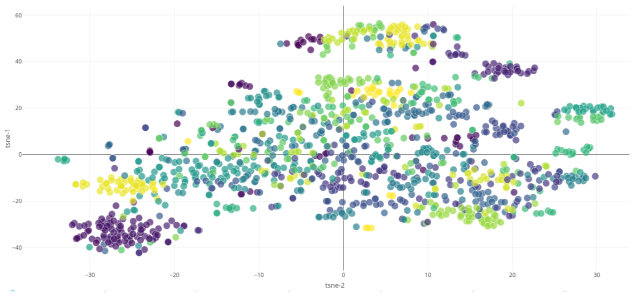
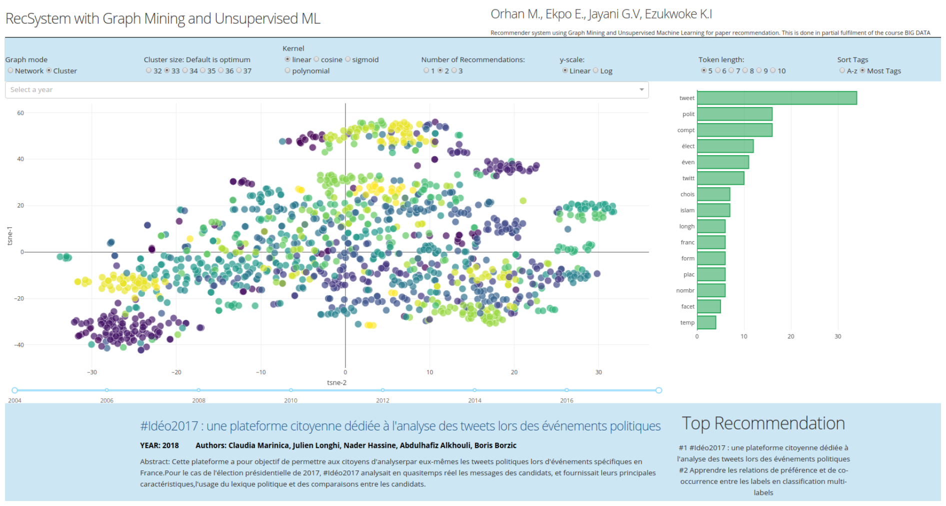


Figure 8: T-SNE Visualization using 33 components

4.1.3 Result and Demo

A demonstration of the project is possible on a free platform online [2]. It might be challenging loading the graph module on the web application, this is essentially because of the time complexity required to render the graph network. Also because we are using a free hosting application. [3]. Choosing different cluster value alters the rendering of the t-SNE distribution. The same applies for the kernels.



RecSystem with Graph Mining and Unsupervised ML

Orhan M., Ekpo E., Jayani G.V, Ezukwoke K.I

Recommender system using Graph Mining and Unsupervised Machine Learning for paper recommendation. This is done in partial fulfilment of the course BIG DATA

Graph mode: Network [✓ Cluster]

Cluster size: Default is optimum: 32 [✓ 33] 34 35 36 37

Kernel: [✓ linear] cosine sigmoid polynomial

Number of Recommendations: 1 [✓ 2] 3

y-scale: [✓ Linear] Log

Token length: [✓ 5] 6 7 8 9 10

Sort Tags: A-z [✓ Most Tags]

#Idéo2017 : une plateforme citoyenne dédiée à l'analyse des tweets lors des événements politiques

YEAR: 2018 **Authors: Claudia Marinica, Julien Longhi, Nader Hassine, Abdulhafiz Alkhouli, Boris Borzic**

Abstract: Cette plateforme a pour objectif de permettre aux citoyens d'analyserpar eux-mêmes les tweets politiques lors d'événements spécifiques en France.Pour le cas de l'élection présidentielle de 2017, #Idéo2017 analysait en quasitemps réel les messages des candidats, et fournissait leurs principales caractéristiques,l'usage du lexique politique et des comparaisons entre les candidats.

Top Recommendation

#1 #Idéo2017 : une plateforme citoyenne dédiée à l'analyse des tweets lors des événements politiques
#2 Apprendre les relations de préférence et de co-occurrence entre les labels en classification multi-labels

Figure 9: Web Application GUI

Component of the webapp

- **Graph Mode**: The default is set to Cluster. By selecting cluster we only want to see a visualization of the clustering result. Selecting Network outputs the result of

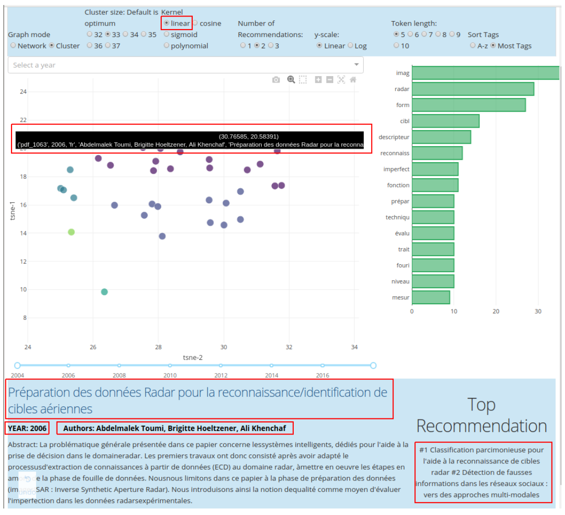
[2]Web application for Unsupervised Clutering with PCA and Network graph Clustering with PCA
[3]Graph Mining Notebook on github Graph Mining

graph minning. Due to server limitations however, the visualization of the graph network crashes the server when loaded.

- **Cluster size**: This component of the application dynamically changes the clustering result. it is the input of both PCA and KMeans algorithm.
- **Kernel**: The kernel component updates the both kernel PCA and kernel KMeans dynamically.
- **number of Recommendation**: This component is responsible for the number of recommendation a user desires to see on hovering on a data point.
- **y-scale**: is the scale of the y-axis to either linear (default) form or logarithmic form.
- **Token Length**: This component controls the size of the token visualized by the bar chart.
- **Year Selector**: A drop down box directly above the t-SNE chart. Used to select years of interest.
- **Main Chart- Scatter plot**: t-SNE chart for clustering and fruchtermanreingold layout for graph network.
- **Bar Chart**: This component displays the top 15 most frequent word.
- **Year Slider**: Component controls the period of year of interest.
- **Footer**: The Footer Contains
  - Title of the datapoint on hover
  - Year: Year of publication
  - Author Name
  - Abstract: Abstract of the paper
  - Top Recommendation: Top recommendations

We observe that the optimal k of 33 is best for recommendation of a research paper. We also observe that the linear kernel is the best for accurately clustering the papers into their respective class. We observe the authenticity of recommendations from Figure

10, 11, 12, 13. Using the linear kernel, we observe that the recommendation for pdf 1063 is more relateable for the linear kernel and cosine kernel than it is for sigmoid and polynomial kernel. 3


Graph mode: Network [✓ Cluster]
Cluster size: Default is optimum: 32 [✓ 33] 34 35 36 37
Kernel: [✓ linear] cosine sigmoid polynomial
Number of Recommendations: 1 [✓ 2] 3
y-scale: [✓ Linear] Log
Token length: [✓ 5] 6 7 8 9 10
Sort Tags: A-z [✓ Most Tags]

Select a year

(30.76585, 20.58391)
('pdf_1063', 2006, 'fr', 'Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf', 'Préparation des données Radar pour la reconna

Préparation des données Radar pour la reconnaissance/identification de cibles aériennes

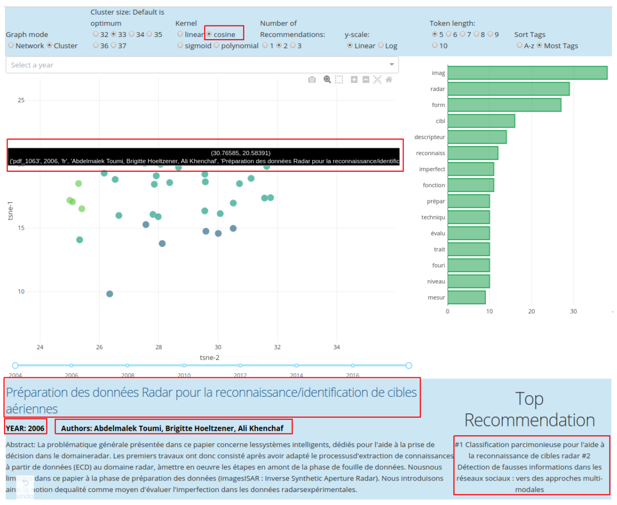
YEAR: 2006 **Authors: Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf**

Abstract: La problématique générale présentée dans ce papier concerne lessystèmes intelligents, dédiés pour l'aide à la prise de décision dans le domaineradar. Les premiers travaux ont donc consisté après avoir adapté le processusd'extraction de connaissances à partir de données (ECD) au domaine radar, àmettre en oeuvre les étapes en am[illegible] e la phase de fouille de données. Nousnous limitons dans ce papier à la phase de préparation des données (im[illegible] SAR : Inverse Synthetic Aperture Radar). Nous introduisons ainsi la notion dequalité comme moyen d'évaluer l'imperfection dans les données radarsexpérimentales.

Top Recommendation

#1 Classification parcimonieuse pour l'aide à la reconnaissance de cibles radar #2 Détection de fausses informations dans les réseaux sociaux : vers des approches multi-modales

Figure 10: Pdf index 1063 with recommendation using k value of 33 and linear kernel


Graph mode: Network [✓ Cluster]
Cluster size: Default is optimum: 32 [✓ 33] 34 35 36 37
Kernel: linear [✓ cosine] sigmoid polynomial
Number of Recommendations: 1 [✓ 2] 3
y-scale: [✓ Linear] Log
Token length: [✓ 5] 6 7 8 9 10
Sort Tags: A-z [✓ Most Tags]

Select a year

(30.76585, 20.58391)
('pdf_1063', 2006, 'fr', 'Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf', 'Préparation des données Radar pour la reconnaissance/identific

Préparation des données Radar pour la reconnaissance/identification de cibles aériennes

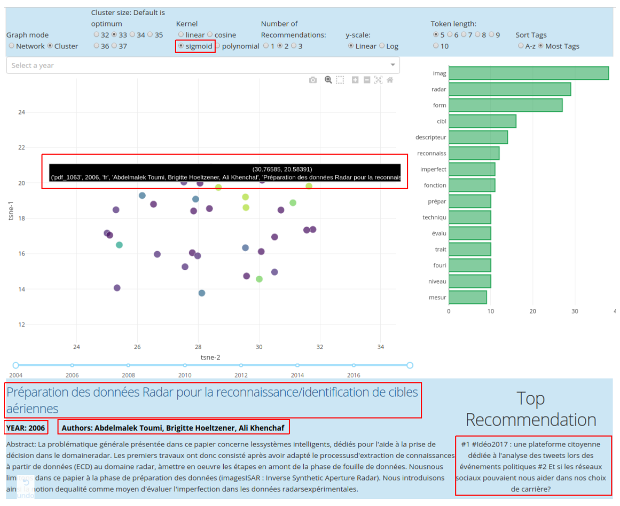
YEAR: 2006 **Authors: Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf**

Abstract: La problématique générale présentée dans ce papier concerne lessystèmes intelligents, dédiés pour l'aide à la prise de décision dans le domaineradar. Les premiers travaux ont donc consisté après avoir adapté le processusd'extraction de connaissances à partir de données (ECD) au domaine radar, àmettre en oeuvre les étapes en amont de la phase de fouille de données. Nousnous lim[illegible] dans ce papier à la phase de préparation des données (imagesISAR : Inverse Synthetic Aperture Radar). Nous introduisons ain[illegible] otion dequalité comme moyen d'évaluer l'imperfection dans les données radarsexpérimentales.

Top Recommendation

#1 Classification parcimonieuse pour l'aide à la reconnaissance de cibles radar #2 Détection de fausses informations dans les réseaux sociaux : vers des approches multi-modales

Figure 11: Pdf index 1063 with recommendation using k value of 33 and cosine kernel


Graph mode: Network [✓ Cluster]

Cluster size: Default is optimum: 32 [✓ 33] 34 35 36 37

Kernel: linear cosine [✓ sigmoid] polynomial

Number of Recommendations: 1 [✓ 2] 3

y-scale: [✓ Linear] Log

Token length: [✓ 5] 6 7 8 9 10

Sort Tags: A-z [✓ Most Tags]

Select a year

(30.76585, 20.58391)
('pdf_1063', 2006, 'fr', 'Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf', 'Préparation des données Radar pour la reconnais

Préparation des données Radar pour la reconnaissance/identification de cibles aériennes

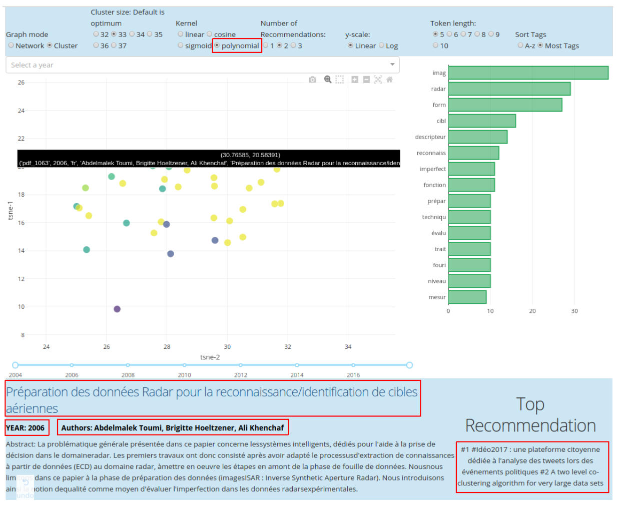
YEAR: 2006 **Authors: Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf**

Abstract: La problématique générale présentée dans ce papier concerne lessystèmes intelligents, dédiés pour l'aide à la prise de décision dans le domaineradar. Les premiers travaux ont donc consisté après avoir adapté le processusd'extraction de connaissances à partir de données (ECD) au domaine radar, àmettre en oeuvre les étapes en amont de la phase de fouille de données. Nousnous lim dans ce papier à la phase de préparation des données (imagesISAR : Inverse Synthetic Aperture Radar). Nous introduisons ain notion dequalité comme moyen d'évaluer l'imperfection dans les données radarsexpérimentales.

Top Recommendation

#1 #Idéo2017 : une plateforme citoyenne dédiée à l'analyse des tweets lors des événements politiques #2 Et si les réseaux sociaux pouvaient nous aider dans nos choix de carrière?

Figure 12: Pdf index 1063 with recommendation using k value of 33 and sigmoid kernel


Graph mode: Network [✓ Cluster]

Cluster size: Default is optimum: 32 [✓ 33] 34 35 36 37

Kernel: linear cosine sigmoid [✓ polynomial]

Number of Recommendations: 1 [✓ 2] 3

y-scale: [✓ Linear] Log

Token length: [✓ 5] 6 7 8 9 10

Sort Tags: A-z [✓ Most Tags]

Select a year

(30.76585, 20.58391)
('pdf_1063', 2006, 'fr', 'Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf', 'Préparation des données Radar pour la reconnaissance/iden

Préparation des données Radar pour la reconnaissance/identification de cibles aériennes

YEAR: 2006 **Authors: Abdelmalek Toumi, Brigitte Hoeltzener, Ali Khenchaf**

Abstract: La problématique générale présentée dans ce papier concerne lessystèmes intelligents, dédiés pour l'aide à la prise de décision dans le domaineradar. Les premiers travaux ont donc consisté après avoir adapté le processusd'extraction de connaissances à partir de données (ECD) au domaine radar, àmettre en oeuvre les étapes en amont de la phase de fouille de données. Nousnous lim dans ce papier à la phase de préparation des données (imagesISAR : Inverse Synthetic Aperture Radar). Nous introduisons ain notion dequalité comme moyen d'évaluer l'imperfection dans les données radarsexpérimentales.

Top Recommendation

#1 #Idéo2017 : une plateforme citoyenne dédiée à l'analyse des tweets lors des événements politiques #2 A two level co-clustering algorithm for very large data sets

Figure 13: Pdf index 1063 with recommendation using k value of 33 and polynomial kernel

4.2 Evaluation for Graph Modeling

4.2.1 Evaluation metric

In order to evaluate the quatlity of our communities, we used the modularity measure which value ranges from -1 to 1. It evaluates the density of the edges within the same communities with respect to different communities [12]. Modularity measure is be defined as:

$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i * k_j}{2m} \delta(c_i, c_j), \quad (31)$$

Where

- A_ij represents the edge weight between nodes i and j ;
- k_i and k_j and are the sum of the weights of the edges attached to nodes i and j respectively;
- m is the sum of all of the edge weights in the graph;
- c_i and $c_j$and are the communities of the nodes;
- δ is Kronecker delta function ($\delta_{x,y}$ if $x = y$, 0 otherwise).

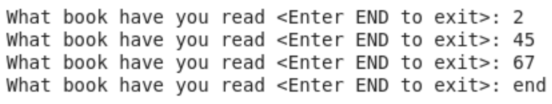
We recorded a modularity score of **0.128** for the Louvain Approach. (Algorithm used for our community detected task)

4.2.2 Result and Demo

We implemented a console interface for the proposed recommendation system. As input, the user will provide the Ids of the articles he/she has already read, then we will provide as output the recommended articles.

Step1:

Repetitively input the ID of the articles that you read and then enter **END** once done as shown below



```
What book have you read <Enter END to exit>: 2
What book have you read <Enter END to exit>: 45
What book have you read <Enter END to exit>: 67
What book have you read <Enter END to exit>: end
```

Figure 14: Screenshot Demo Input

Step 2 :

You will be provided with the subgraph H of the best candidate selected using community detection

```
You have read 3 Articles : [ [2, 67, 45] ]

Recommendation
---------------
[2, 67, 45, 879, 778, 1190, 547, 316, 268, 324, 814, 878,
```

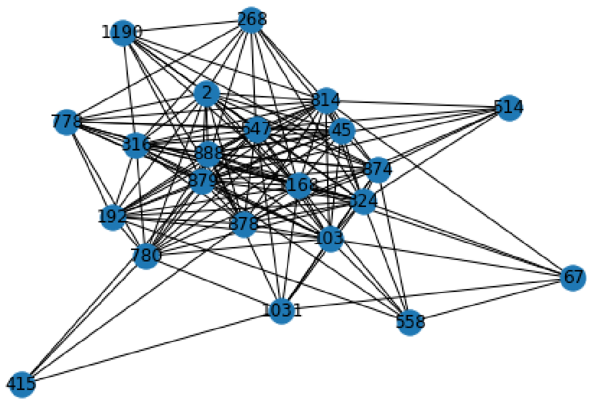
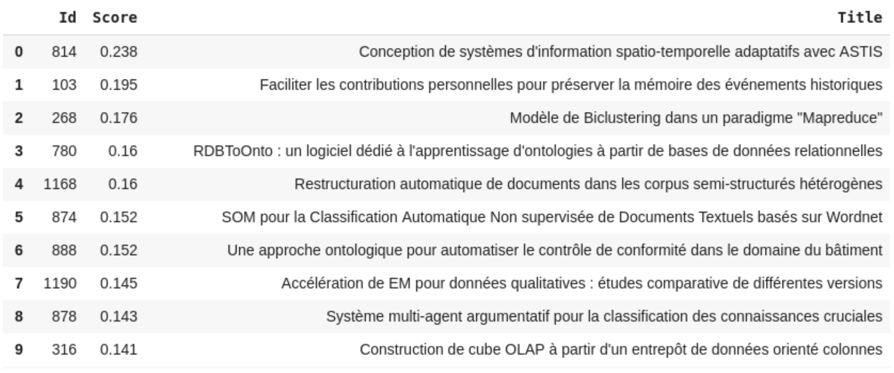


Figure 15: Screenshot Demo Candidate Graph H (Subgraph G)

Step 3:

Finally a Table of all recommendations will be displayed to your ordered according their respective ranking score.



	Id	Score	Title
0	814	0.238	Conception de systèmes d'information spatio-temporelle adaptatifs avec ASTIS
1	103	0.195	Faciliter les contributions personnelles pour préserver la mémoire des événements historiques
2	268	0.176	Modèle de Biclustering dans un paradigme "Mapreduce"
3	780	0.16	RDBToOnto : un logiciel dédié à l'apprentissage d'ontologies à partir de bases de données relationnelles
4	1168	0.16	Restructuration automatique de documents dans les corpus semi-structurés hétérogènes
5	874	0.152	SOM pour la Classification Automatique Non supervisée de Documents Textuels basés sur Wordnet
6	888	0.152	Une approche ontologique pour automatiser le contrôle de conformité dans le domaine du bâtiment
7	1190	0.145	Accélération de EM pour données qualitatives : études comparative de différentes versions
8	878	0.143	Système multi-agent argumentatif pour la classification des connaissances cruciales
9	316	0.141	Construction de cube OLAP à partir d'un entrepôt de données orienté colonnes

Figure 16: Screenshot Demo Recommendation Dataframe

The number of candidates for the recommendation can be altered based on the user needs in the implementation notebook.

5 Conclusion

In this paper we present two approaches to recommender system using graphical modeling and unsupervised machine learning. The graph modeling and the unsupervised text mining behave similar and return the same output even though there can be little changes according to the candidate ranking in the community detection as the clustering doesn't return the reference based on ranking. The clustering with text mining returns only the set of possible un-ranked recommender documents. This system can be expanded to detect diversified languages. Some papers contains lots of image text that we already skipped during the study as it is out of the scope of this research. But in future, this research can expand into text extracting from images as well through the existing APIs such as goole vision API, pytessaract or OCR techniques. In addition, during this research there were some difficulties to run clique community detection algorithm due to the size of the available RAM.

References

[1] Ryan Compton,Community detection and colored plotting in networkx, Web Article, http://ryancompton.net/2014/06/16/community-detection-and-colored-plotting-in-networkx/, 16 June 2014

[2] SuHun Han, Free Google Translate API for Python, Official Python Library Web Page, https://pypi.org/project/googletrans/, 2015

[3] Dean Malmgren, textract Documentation, Official Python Library Web Page, http://textract.readthedocs.io/en/latest/, 2014

[4] Agarwal, N. Haque, E. Liu, H. and Parsons L., Research Paper Recommender Systems: A Subspace Clustering Approach, In Advances in Web-Age Information Management, Springer: Heidelberg,2005, Web Page,https://link.springer.com/chapter/10.1007/11563952_42

[5] Gipp B., et.al, Scienstien: A research paper recommender system, Fraunhofer Institute for Telecommunications, Berlin, Germany, Web Page, https://pdfs.semanticscholar.org/deab/9886bd1f39bea5b85fa76ca8f705fec9a85c.pdf

[6] Diamantaras K.I, Kung S.Y. Principal component neural networks: theory and applications. John Wiley & Sons, Inc., ISBN:0-471-05436-4, New York, USA, 1996.

[7] Pearson K. On lineas and Planes of closest fit to systems of points in space. *Philos Mag A*, 6:559–572, 1901.

[8] Bernhard Schölkopf, Alexander Smola, KlausRobert Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5): 1299-1319, 1998.

[9] Weinberger, Kilan Q., Sha, Fei et al. Learning a kernel matrix for nonlin- ear dimensionality reduction. *Proceedings of the 21st International Conference on Machine Learning (ICML-04)*, pp. 839–846. ACM Press, 2004.

[10] Laurence Morissette and Sylvain Chartier. The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology.* 9(1): pp. 15-24.

[11] Inderjit S. Dhillon, Yuqiang Guan, Brian Kulis. Kernel k-means, Spectral Clustering and Normalized Cuts. *[Information Search and Retrieval.* 9(1): pp. 15-24.

[12] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. (2008) P10008

[13] Argimiro Arratia R. Ferrer-i-Canch,Community structure in networks, https://www.cs.upc.edu/ CSN/slides/07communities.pdf, 2019.

[14] NetworkX Developers, Python Community Library Official Documentation ,https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.comm
2019

[15] M. E. J Newman , Networks: An Introduction, page 224 Oxford University Press, 2011

[16] Clauset, A., Newman, M. E., Moore, C. Finding community structure in very large networks. Physical Review E 70(6), 2004.

[17] L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research 15(Oct):3221-3245, 2014.

[18] L.J.P. van der Maaten and G.E. Hinton. Visualizing Non-Metric Similarities in Multiple Maps. Machine Learning 87(1):33-55, 2012.