Data Mining for Big Data Project: Analysis on EGC Dataset

Mohammad Poul Doust, Robin Khatri, Samaneh Zareian Jahromi

January 25, 2020

Contents

Abstract

In this report, we have done data analysis on EGC conference dataset and we have extracted information from some aspects of the data recorded between 2004-2018. We first did an exploratory analysis to find distributions of authors and articles, then inferred the correlation among authors by calculating the Pearson's correlation coefficient; we clustered authors using Hierarchical Clustering. We also discovered frequent item-set applying apriori algorithm. Based on our findings on groups of authors writing together, we shed light on interesting facts on the network of authors of the EGC society. Lastly we explored the abstracts of papers with topic modeling using K-Means clustering and an LDA model.

keywords: Collaboration Network, Correlation, LDA

1 Introduction

La conférence Extraction et Gestion des Connaissances (EGC) is a francophone conference specialized in data mining and knowledge discovery. It is organized every year and it celebrates its 20th anniversary this year [1]. They have made data available of their articles since 2004 [2]. The description of the dataset is available in the Section 3. In this report, we have used this dataset for clustering authors and as well as documents. We also looked into the collaborations among authors that have published in the conference over the years. In order to identify the topics and their evolution, we also did topic modelling. In the following sections, we have discussed our methods and results in detail.

In the context of analysing co-authorship, collaboration networks have been an area of active research and many researchers over the years have looked into the networks of authors belonging to many domains *e.g.* science, social science etc [3][4][5]. Both temporal and centrality measures have been extensively studied [6] [7] [8]. On EGC dataset, we have adopted a bottom-top approach, where we have looked into very high level relationships and building hierarchical clusters of group of authors that actively publish in EGC together. We also looked into constructing networks that can be further analyzed. Our purpose in this examination was to investigate and identify active and regular collaborations.

In the second part of our studies, we have looked into both clustering the topic modelling through Latent Dirichlet Allocation [9]. Latent Dirichlet Allocation is a popular method of identifying topics that can be used to find similarity amongst documents and to cluster as well as seeing changes in those topics over time [9]. We have discussed briefly the theory of LDA in the Section 6.

In the following sections, we have looked into our objectives and dataset. Thereafter we have done studies towards fulfilling our objectives.

2 Objectives and methods

In this study, we had two objectives in mind: First, to understand the data and identify key information about authors. We also wanted to look into building a collaboration network. Second, we wanted to do topic modelling and see evolution of topics over time.

The methods used in the studies are described under respective sections. We used Python and R for this study. Python was used in Google Colab environment [10]. For topic modelling and Network building, we have followed some tutorials to understand them. Wherever needed, we have made a reference to those.

3 Data Description

The dataset analyzed in this report represents articles published in EGC conference from 2004 towards 2018. Each row in this file corresponds to an article represented by the following attributes: series, book title of the proceedings, year, title of the article, abstract, authors, URL of a pdf of the first page of the paper, and a URL of the pdf of the full paper.

Column	**Type**	**Description**
Series	Text	Conference series for the article
Booktitle	Text	Publication book (Constant - EGC)
Year	Numeric	Year of publication
Title	Text	Title of the article
Abstract	Text	Abstract of the article
Authors	Text	Names of the authors
Pdf1page	Text	Hyperlink for first page of the article
Pdfarticle	Text	Hyperlink for the complete article

Table 1: EGC Dataset Description. The Dataset has abstract as well as author names for articles published between 2004 and 2018. There are total **1269 samples** available.

Each line of the file is a record and the attributes are separated by horizontal tab characters. The data contains some errors (especially in the abstract field). There are several samples that have missing abstracts as observed in the following section. In the coming section, we have discussed some exploratory observations made on the dataset.

4 Exploratory Analysis

In this section, we looked into three things: distribution of authors as well as articles over years to get an overview of the dataset, and the trend in the average number of authors per year.

4.1 Distribution of authors over the years

We first extracted unique authors from the "author" column and afterwards a simple data preprocessing is done to split the authors (since the authors are comma delimited). Finally, we processed the obtained list of authors to throw away the duplicates. There are 2007 unique authors in the dataset. Their distribution is presented in the Figure 10.

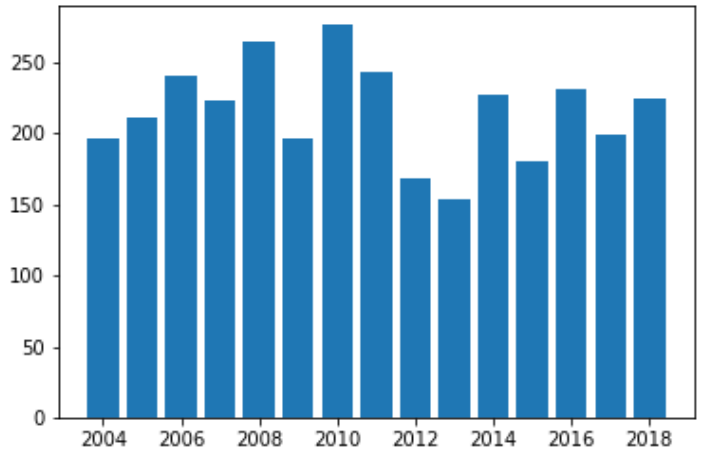

Figure 1: Distribution of authors per year

On average, there are around 216 unique authors per year. We can see that the number of authors reach its maximum in the year 2010 (more than 250 authors), while it is minimum is in the year 2013 (around 150 authors).

4.2 Distribution of articles over the years

We grouped the dataset on "year" column and we use the COUNT as an aggregation function on the grouped data.
We can see in the Figure 2. that most of the mass centered in year 2010 (more than 100 articles). On the other hand, the least number of articles were published in year 2013 (less than 60 articles). Moreover, on average there are about 85 articles published on ECG per year. Evidently, the distribution of articles over years follow the same

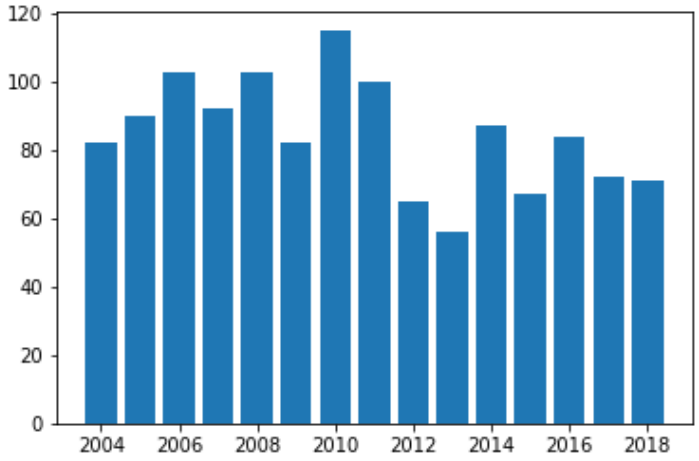

Figure 2: Distribution of articles per year

trends observed in section 4.1 for the distribution of authors per year. Therefore, there is no trend where we have large number of authors contributing to the same article.

4.3 Number of authors per paper

In order to get an idea about the intensity and trend in collaboration amongst authors, we looked into the trend of average number of authors collaborating on one paper. In the Figure 3, it is obvious that the average number of authors publishing a paper together in the conference has been rising (From 2.73 in 2004 to 3.44 in 2018). This motivated us to look into author-author collaboration further.

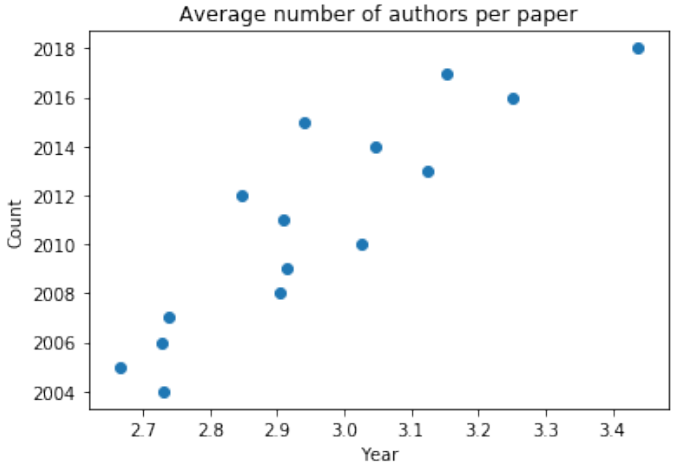

Figure 3: Distribution of average number of authors per paper per year.

5 Study on authors

In this section, we have explored the 'author' column of our datset with an objective to find the author-author correlations, authors that publish most, author-author collaboration and their degree in a co-authorship network.

5.1 Correlation among authors

The objective of this subsection is to study the correlation behavior between authors. This correlation was important to extract set of pairs of authors that write together more frequently.

In order to find the pair-wise correlation between authors, we first did a pre-processing to produce one-hot encoded columns with respect to the "author" column. To this goal, we added to the dataset new column for each author, this column was be a Boolean field with value 1 if same authors contributed to a certain article. see the Table 2.

Title	**Year**	**Abstract**	**Author_1**	**Author_2**	**Author_3**	**Author_4**	**Author_5**	....
A two level co-clustering ...	2018	.....	False	True	False	True	True	....
Analyse des sentiments...	2018	.....	True	False	False	False	False	....

Table 2: One hot encoding of authors.

After transforming the dataset, it was easy to study the pairwise correlation between authors by calculating the Pearson's correlation coefficient [11]. We excluded the self-correlations pairs. we calculated the correlations iteratively by merging each correlated author and re-calculate the correlations with the others. for this section we explored only the first-level correlations since we have exploited the authors levels through other methods (like clustering) as detailed further. We calculate the Pearson Coefficient according to the Equation 1:

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

Where:
cov is the covariance between X and Y
σ_X is the standard deviation of X
σ_Y is the standard deviation of Y

In Figure 4, we can see the heat-map of the pairwise correlation among authors. Since we would have a correlation of 1 for all authors who have published only 1 article, we tested different thresholds for the number of publications (we filter authors with few articles in all dataset). We can see that we have a complete correlation between several authors up to 8 articles. This represents the fact that there are several authors who have published 8 or more articles always together (since it a complete correlation).

Moreover, in all cases, we can see the diagonal red line corresponding to the self-correlation trend. Additionally, It is worth noticing that we have complete correlation (dark-red points) aligning vertically or horizontally (same X or Y Axis) creating a blob, those corresponds to the authors correlating to the same author forming a group of correlated authors. Moreover, Table 3 and Figure 5 show a sample of the pair-wise correlation (preserving authors that have exactly 8 articles). It is obvious that we have cross correlation between authors. Finally, the Figure 5 depicts the number of authors who have more than X articles, that is interesting because intuitively, since the data spans years from 2004 to 2018, we would expect that each author would have at most one paper in one year, but the figures show that we have around 4 people who has around 20 papers in total, hence, more than one paper in a year.

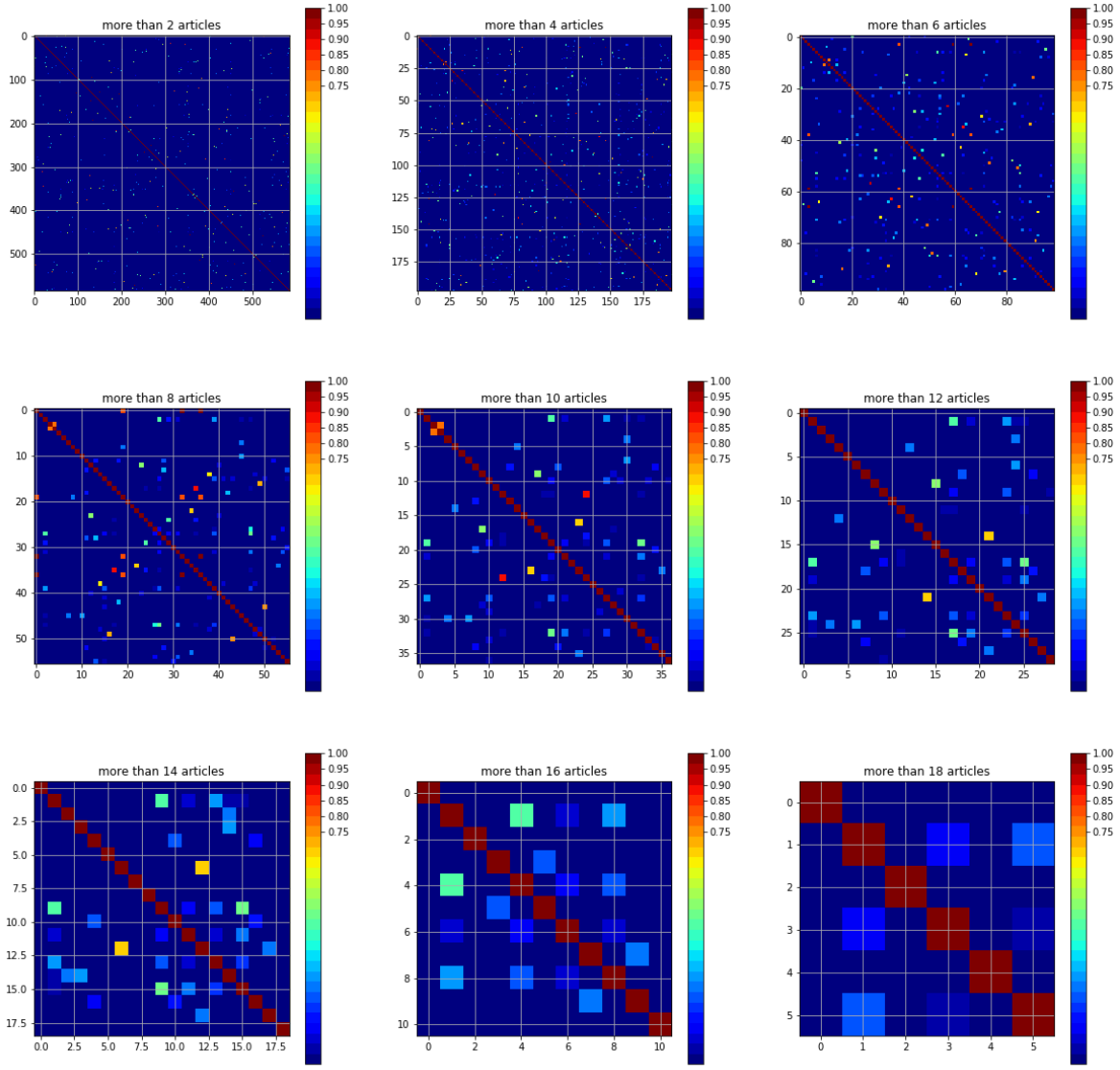

Figure 4: Pairwise author-author correlation with different thresholds (keeping authors that have articles more than a specific threshold).

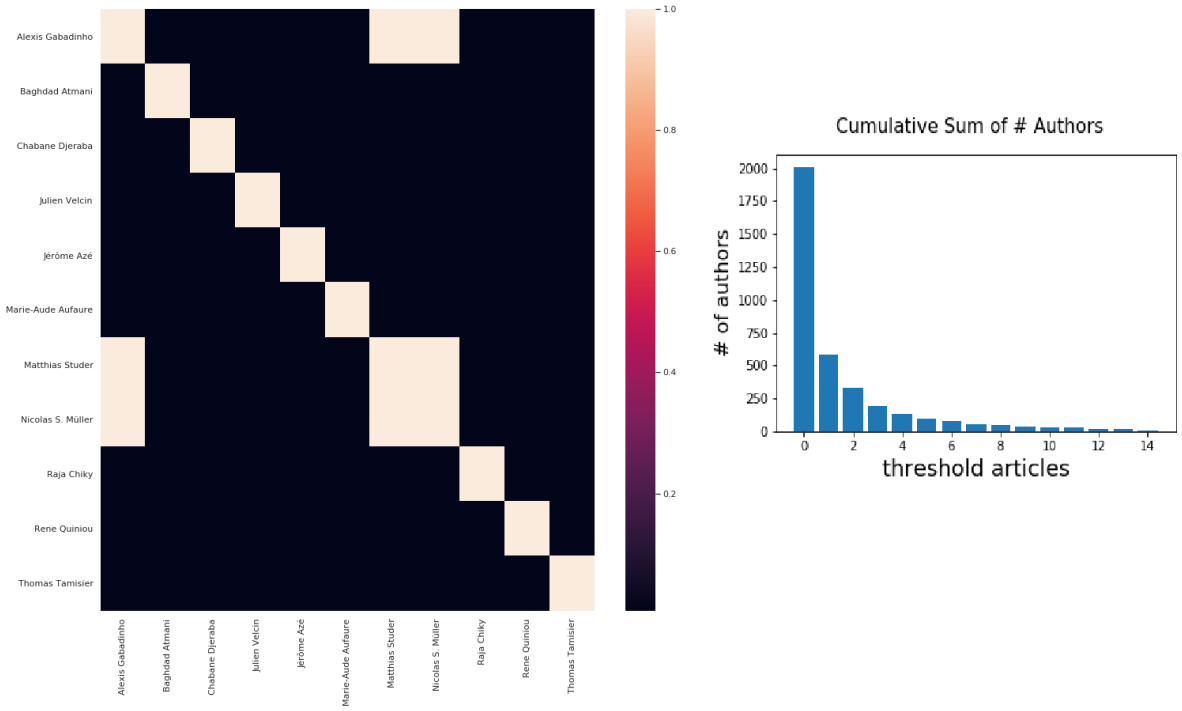

Figure 5: (Left:) Pairwise correlation matrix among authors with exactly 8 articles. (Right:) The cumulative sum of authors for different number of published articles.

Author 1	**Author 2**	**Correlation**
Matthias Studer	Nicolas S. Müller	1.0
Alexis Gabadinho	Matthias Studer	1.0
Alexis Gabadinho	Nicolas S. Müller	1.0
Julien Velcin	Jérôme Azé	0.06

Table 3: Sample of Correlated authors (Having at least 8 articles)

We can conclude that the dataset contains authors who are fully-correlated and have more than one paper in common (at most 8 papers in some cases). It means they write together always. Moreover, there are authors who contributed to more than one paper in the same conference.

Further, from the correlations heat-map, it is evident that we have concentrated blobs (aligned horizontally or vertically) - indicating high correlation. This represents groups of correlated authors (*i.e.* more than two collaborating authors). Therefore, we next applied hierarchical clustering algorithm on the correlation matrix to capture these groups (*i.e.* clusters).

5.2 Hierarchical clustering on the Author-Author Correlation Matrix

As a natural progression of our previous study on pairwise correlations among authors, we clustered authors using Hierarchical Clustering [12]. For this clustering, we used correlation coefficient defined in the Equation 1 to get a distance measure as defined in the Equation 3:

$$d_p(x,y) = 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2}$$

Hence:

$$d_p = 1 - \rho_{X,Y} \tag{3}$$

For hierarchial clustering, Agglomerative method [13] [14] has been used:

Algorithm 1: Hierarchical Agglomerative Clustering

Input: Data Vectors $\{x_n\}_{n=1}^{N}$
$\mathcal{A} \leftarrow \varnothing$;
for $n \leftarrow 1..N$ **do**
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\{\boldsymbol{x}_n\}\}$
end
$\mathcal{T} \leftarrow \mathcal{A}$
while $|\mathcal{A}| > 1$ **do**
  $G_1^\star, \mathcal{G}_2^\star \leftarrow \underset{\mathcal{G}_1,\mathcal{G}_2 \in \mathcal{A}}{\operatorname{argmin}} DIST(\mathcal{G}_1, \mathcal{G}_2)$
  $\mathcal{A} \leftarrow (\mathcal{A} \setminus \{\mathcal{G}_1^\star\}) \setminus \{\mathcal{G}_2^\star\}$
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathcal{G}_1^\star \cup \mathcal{G}_2^\star\}$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathcal{G}_1^\star \cup \mathcal{G}_2^\star\}$
end
Return *Tree* $\mathcal{T}$.

For merging two clusters, there are are 4 main criterions as following:

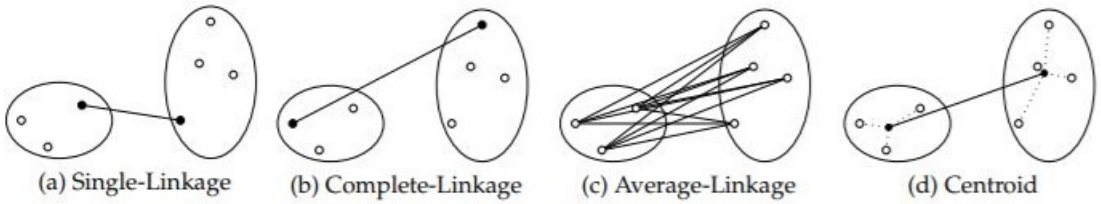

Figure 6: Linkage Types

Single-Linkage It merges two groups that have shortest distance (between all pairs

of elements):

$$\text{DIST-SINGLELINK } \left(\{\boldsymbol{x}_n\}_{n=1}^{N}, \{\boldsymbol{y}_m\}_{m=1}^{M}\right) = \min_{n,m} \|\boldsymbol{x}_n - \boldsymbol{y}_m\|$$

Complete-Linkage Groups are merged depending on the largest distance between elements as a distance between two groups:

$$\text{DIST-COMPLETELINK } \left(\{\boldsymbol{x}_n\}_{n=1}^{N}, \{\boldsymbol{y}_m\}_{m=1}^{M}\right) = \max_{n,m} \|\boldsymbol{x}_n - \boldsymbol{y}_m\|$$

Average-Linkage In this type, distances between elements are averaged and considered as a representative for the distance between two groups

$$\text{DIST} - \text{AVERAGE} \left(\{\boldsymbol{x}_n\}_{n=1}^{N}, \{\boldsymbol{y}_m\}_{m=1}^{M}\right) = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} \|\boldsymbol{x}_n - \boldsymbol{y}_m\|$$

Centroid-Linkage For this type, the distance between two centroids of two clusters are used as a distance measure between two clusters.

$$\text{DIST-CENTROID } \{x_n\}_{n=1}^{N}, \{y_m\}_{m=1}^{M} = \left\| \left(\frac{1}{N} \sum_{n=1}^{N} x_n\right) - \left(\frac{1}{M} \sum_{m=1}^{M} y_m\right) \right\|$$

In the results (Figure 7), complete-linkage has been used with different filtering methods has been applied to filter out authors who contributed to a very low number of articles (For example if two authors wrote only 1 article together they could be clustered together in one meaningless cluster). We can see from that for each threshold, we are able to identify some author-groups who tend to write together. Moreover, by choosing different thresholds, we can have different number of clusters (the higher the threshold, the less clusters we get). Also, we can see that our results in the Table 3 agree with this experiment (upper left-most picture). Where we have a perfect cluster of authors Matthias Studer, Alexis Gabadinho, Nicolas S. Muller.

(1).png

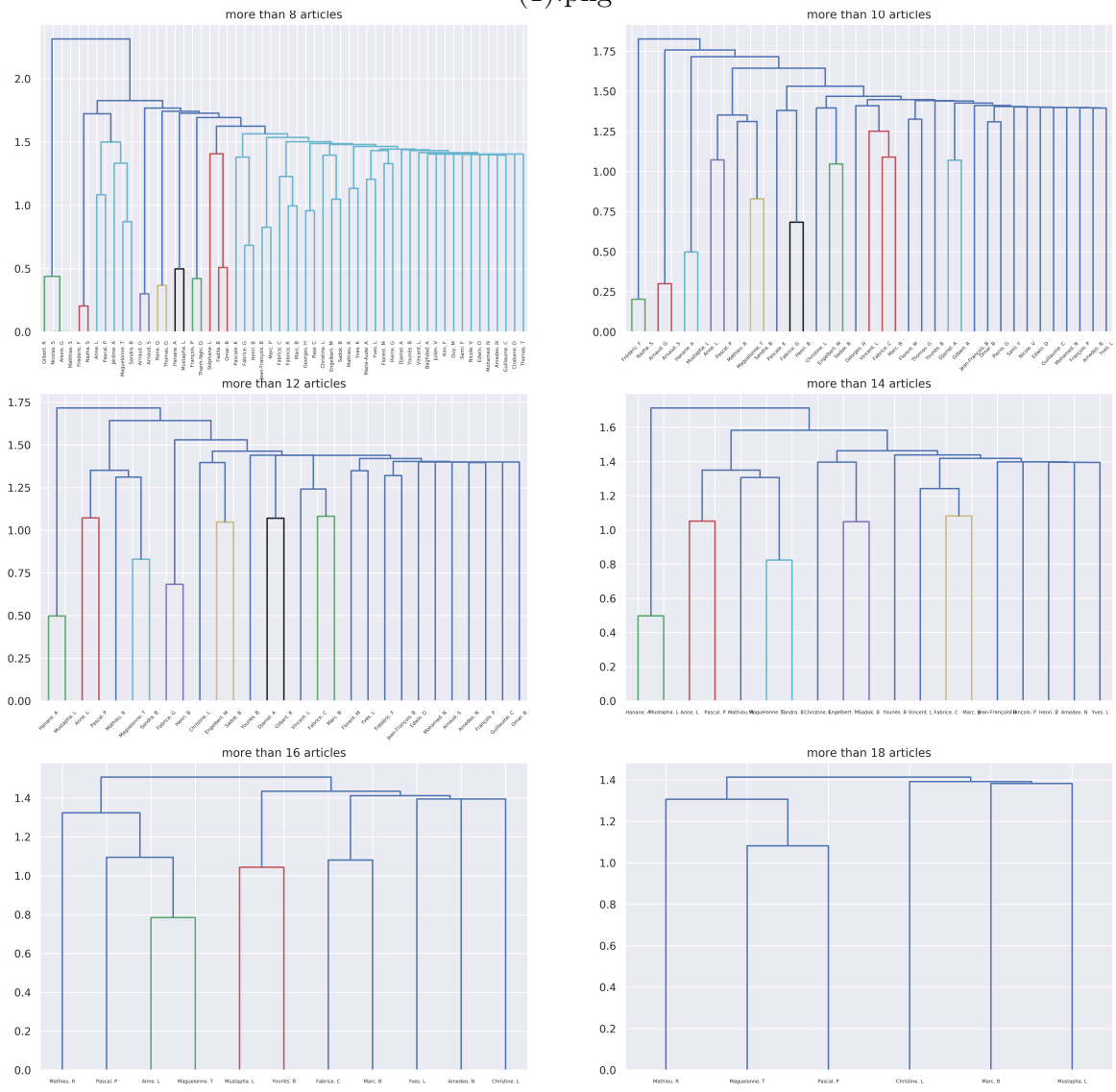

Figure 7: hierarchical clustering Dendrograms with 1-correlation as a distance measure. Authors who had very few papers have been filtered out.

In conclusion, using only Pearson correlation as a pair-wise similarity, we were able to create clusters of authors by using hierarchical clustering and these clusters are in line with what we observed studying correlations among authors.

5.3 Frequent groups of authors

We employed a data-mining approach to extract author-wise frequent patterns by treating each set of authors of a given articles as item-set. Then, we applied apriori algorithm [15] for frequent item-set mining. Our assumption, here is that we should be able to

find frequent groups of authors such that they have high correlation among each other. We extracted the authors of each article and encode them by integer. After that, a transaction table is created (Figure 8). Finally the transaction tables is passed to Apriori algorithm which is tuned by choosing a proper support threshold.

	Abdelaziz Bensrhair	Abdelaziz Marzak	Abdelfettah Feliachi	Abdelghani Bellaachia	Abdelhadi Fennan	Abdelhakim Artiba
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...

Figure 8: Transactions Table

From the results, we can see if we want all patterns with length more than 2, we can find pretty much the same results we found in the previous subsections. However, it is easier by this method as we did not have to apply any kind of filtering, and also, we can see the changes in frequent item-sets by changing support threshold. Using this algorithm may, therefore, be more efficient for this kind of problem. Moreover, in some scenarios it could be used as a clustering algorithm. The algorithm is efficient and parameterized by the support and length.

support	itemsets	length
0.006304177	({'Alexis Gabadinho', 'Gilbert Ritschard', 'Matthias Studer'})	3
0.006304177	({'Nicolas S. Müller', 'Gilbert Ritschard', 'Alexis Gabadinho'})	3
0.006304177	({'Nicolas S. Müller', 'Matthias Studer', 'Alexis Gabadinho'})	3
0.006304177	({'Nicolas S. Müller', 'Gilbert Ritschard', 'Matthias Studer'})	3
0.00630417	({'Nicolas S. Müller', 'Gilbert Ritschard', 'Matthias Studer', 'Alexis Gabadinho'})	4

Figure 9: Sample of Apriori output of item-sets with length more than 2 and support ≥ 0.004

5.4 Authors who have published the most

We also explored some information about the authors in the dataset who have published many articles. For this purpose, we used MapReduce. Through a mapper function, we extracted the number of times every author have published. Finally, in a reducer function we sorted the obtained list of authors to find the top authors. The top authors are listed in figure 10.

```
{'claudia marinica': 2,
 'julien longhi': 1,
 'nader hassine': 1,
 'abdulhafiz alkhouli': 1,
 'boris borzic': 1,
 'marius barctus': 1,
 'marc boullé': 36,
 'fabrice clérot': 16,
 'jean-benoît griesner': 3,
 'talel abdesssalem': 4,
 'hubert naacke': 4,
 'pierre dosne': 1,
 'abdeljalil elouardighi': 1,
 'mohcine maghfour': 1,
 'hafdalla hammia': 1,
 'fatima-zahra aazi': 3,
 'elyase lassouli': 1,
```

(a) Mapper output

```
[('marc boullé', 36),
 ('pascal poncelet', 25),
 ('maguelonne teisseire', 23),
 ('mustapha lebbah', 21),
 ('christine largeron', 18),
 ('mathieu roche', 18),
 ('amedeo napoli', 17),
 ('yves lechevallier', 17),
 ('anne laurent', 17),
 ('fabrice clérot', 16),
 ('younès bennani', 16)]
```

(b) Reducer output

Figure 10: The number of published articles per author through map-reduce functions

We have identified the top most authors in terms of number of publication in the conference. The author who published the most than others is Marc Boullé with 36 articles.

5.5 Co-authorship Network

After observing and identifying groups of authors writing together, we constructed a network of authors with edges if they have written at least one paper together in the conference. For this, we used networkDynamic [16] and statnet [17] packages of R[18]. We built networks based on the tutorial at UC Davis DataLab [19]. As seen in the Figure 11, since we have many authors, this is not an efficient way to visualize. Therefore, we used networkDynamic to create a dynamic HTML visualization. Its screenshot is presented below. The table 4 presents top 10 author(s) ranked according to their in-degree centrality.

Figure 11: Co-Authorship Network (2016-18). Node size is proportional to log-normalized in-degree.

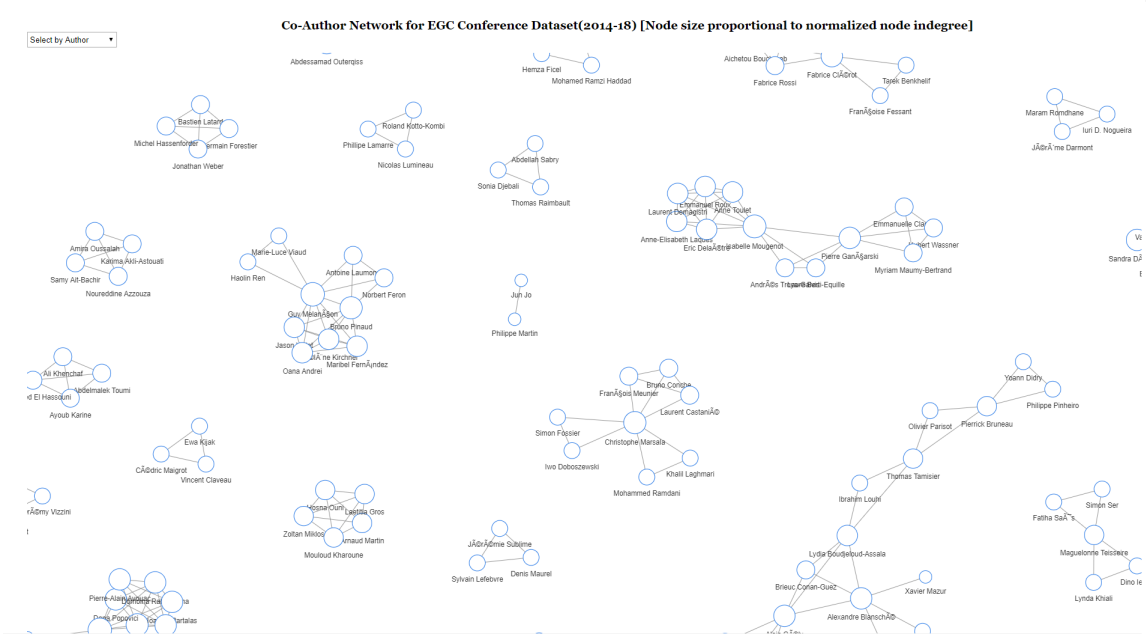

Figure 12: Interactive co-authorship network (2014-18).

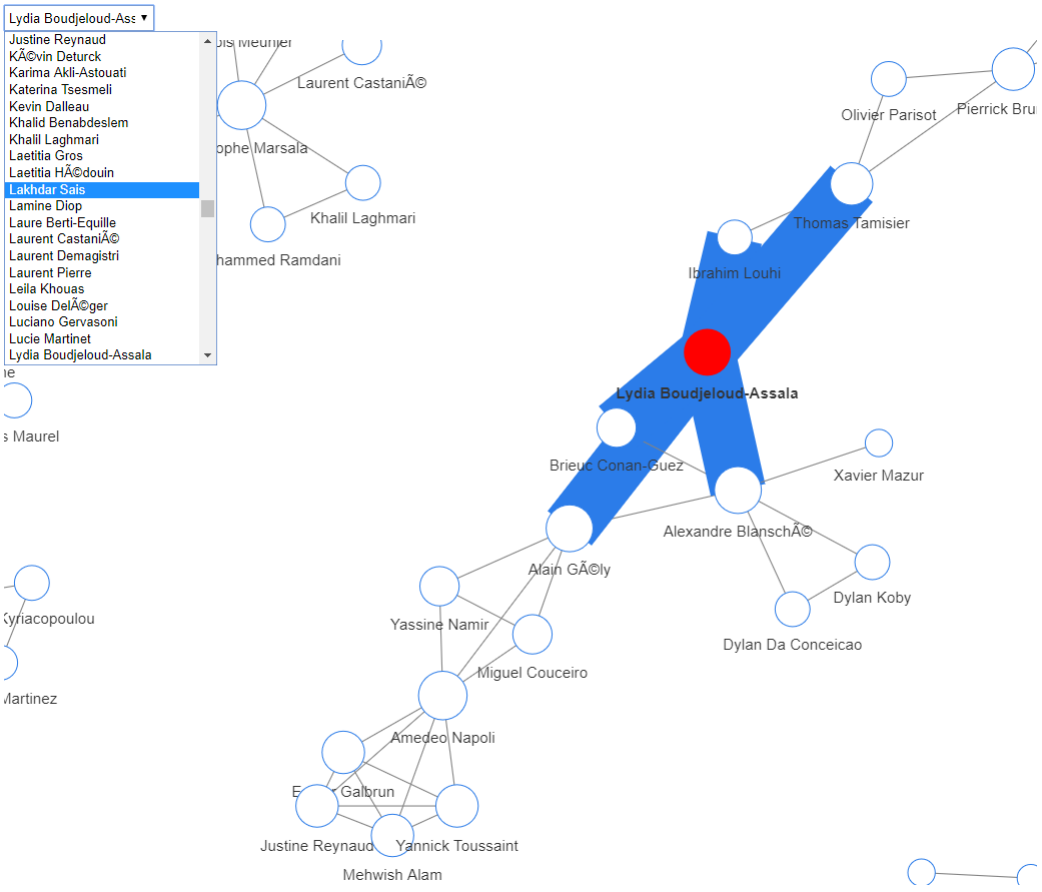

Figure 13: Interactive co-authorship network (2014-18). Different authors can be selected from the drop-down menu.

Author(s)	**Degree**
Pascal Poncelet	41
Mathieu Roche	40
Amedeo Napoli	39
Maguelonne Teisseire	31
Anne Laurent	30
Jean-François Boulicaut	29
Engelbert Mephu Nguifo , Yves Lechevallier	25
Henri Briand , Omar Boussaid , Pascale Kuntz , Sandra Bringay	24
Mustapha Lebbah	23
Sadok Ben Yahia, Thomas Guyet	22

Table 4: Authors with maximum degree (Between 2004-2018). Degree of a node refers to the number of links that connect with it. [20].

We can see that authors- Pascal Poncelet, Yves Lechevallier and Mustapha Lebbah are also the authors that are top authors in terms of number of articles published (See the Figure 10). Interestingly, however, there are some authors who have not published as much as the authors in the Figure 10 but they still have higher degree than the later. This is a proof that those authors mostly publish in collaborative groups.

6 Study on abstracts

In this section, we explored abstracts of papers with a view to obtain clusters and topic model for these abstracts with a view to better understand the terms as well as topics of the abstracts. We also looked at the time evolution of terms within topics over the years.

6.1 Preprocessing

In this subsection, we have defined our pre-processing as well as strategy on the abstract dataset.
First, we checked for missing abstracts and observed that there were 173 samples without any abstract. We removed these samples from our dataset. Then, in order to get a high-level overview about the abstracts, we used K-Means clustering with 5 clusters. Thereafter, we fitted an LDA model without any tuning, to see if we can staright away find some differences or some outlier. For every cluster, top terms in two topics are given below:

```
Topic #0: the of we
Topic #1: the of and

Topic #0: de et des
Topic #1: de des la

Topic #0: de les motifs
Topic #1: de des motifs

Topic #0: social de km
Topic #1: de des les

Topic #0: de la des
Topic #1: de les une
```

Evident from above, there is a cluster that contained English words. To see if there are some documents in English, we used langdetect library of Python. We found that there are 105 abstracts that are in English. Therefore, we extracted French and English subsets of our dataset and created two datasets. For our further clustering and topic modelling, we used only French documents as the size of English dataset was comparatively very small (991 French vs 105 English). Since, these abstracts were in different languages, to process these abstracts together for building a topic model would lead to mistakes. Hereafter, 'dataset' refers to the dataset containing abstracts in French.

Now, to cluster accurately, we lemmatized our French dataset and also removed stopwords. For stopwords, we used French stopwords from stopwords-iso [21] as the number of French stopwords in nltk [22] was quite less (157 as opposed to 690 in stopwords-iso). We also appended the stopwords list to add some other stopwords, as given below:

'donner', 'variable', 'variables', 'data', 'sciences', 'méthode', 'méthodes', 'technique', 'techniques', 'donnes', 'partir', 'article', 'articles', 'algorithme', 'algorithmes', 'approche', 'approches','système', 'papier', 'contribution', 'recherche', 'utiliser', 'données', 'nouvelle', 'proposer', 'rechercher', 'méthode', 'utilisér'

These stopwords are added because in our initial modelling, we noticed that many topics had words like 'variable' and 'papier' with high probability. We feel that we could have added more stopwords but since our first language is not French, it was difficult to identify stopwords further. For lemmatization, we used `spacy_lefff` [23] with POS tags. We also converted all texts in abstract to lowercase before removing stopwords and lemmatizating the texts. We also did a seperate processing with the stopwords from nltk and using only stemmizer. We tested LDA model on both these preprocessed datasets as explained in the coming sections.

6.2 K-Means Clustering

In this subsection, we have presented the results and analysis of K-Means clustering on the abstracts. The dataset used was cleaned with larger list of stopwords and it was also lemmatized as explained the previous subsection.
In order to find the optimal number of clusters for K-Means clustering [24] [25], we used elbow method [26]. However, after even 30 clusters, the model did not converge as visible in the Figure 14.

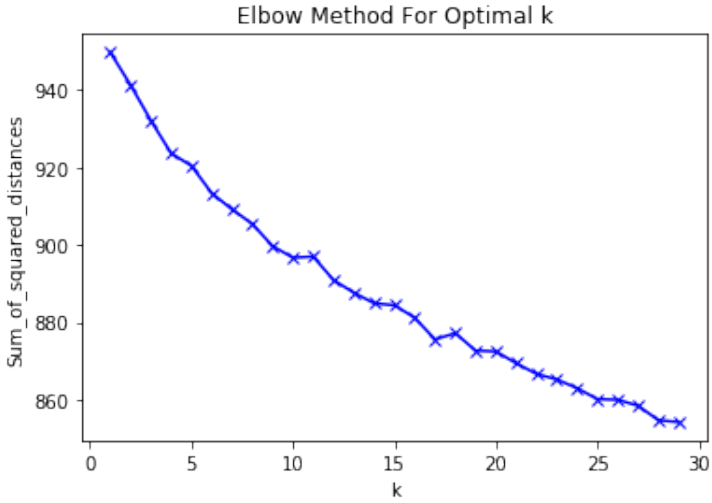

Figure 14: Finding Optimal number of clusters in K-Means.

To see whether the sum of squared distances converges, we did further testing with Elbow method using a minibatch samples of the abstracts with sklearn [27]. The result is presented in the Figure 15.

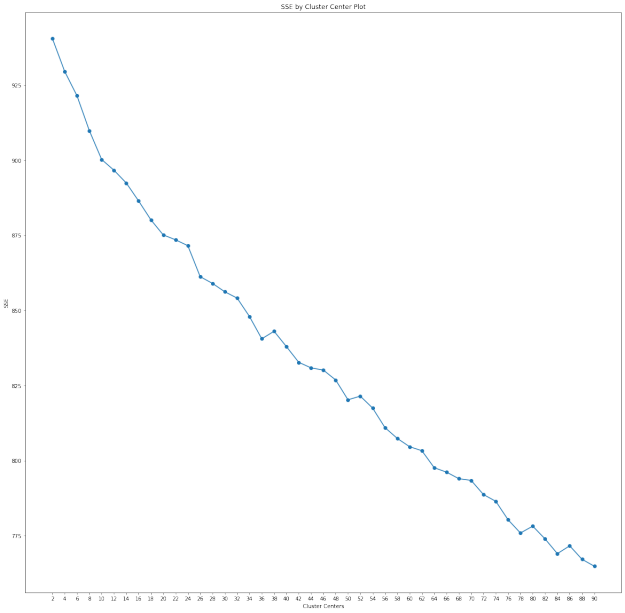

Figure 15: Finding Optimal number of clusters in K-Means.

Clearly, K-Means clustering did not converge till 100 clusters. Therefore, we decided that K-Means may not have been a suitable method for clustering on this dataset. We built a K-Means model for 40 clusters to see if there is any difference among clusters

when visualized. For this visualization, we used t-SNE [28] [29] on PCA-50 components [30]. As shown in the Figure 16

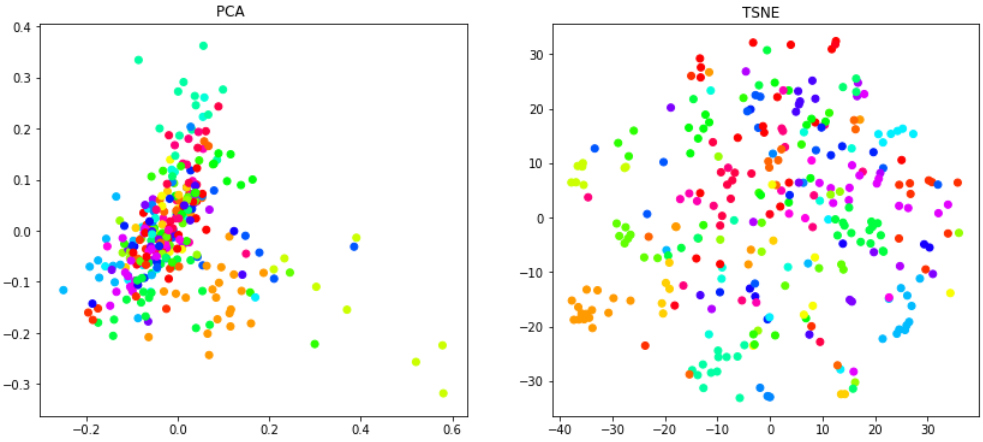

Figure 16: Visualization of clusters on PCA (50 components) and t-SNE on PCA.

Further, we used Lateral Dirichlet Allocation (LDA) [9]. for topic modelling described in the coming subsection.

6.3 Topic Modelling with Lateral Dirichlet Allocation (LDA)

In this study, we did topic modelling using both of the aforementioned preprocessed datasets. To build our LDA models, we used gensim library in Python [31].

We first, did this by using only stemming and using minimal stopwords from nltk to check our hypothesis that we may get mostly these stopwords in our topics. To visualize the results, we used Wordclouds visualization methods on topic words.

We started by cleaning the abstract fields from the stop words. Afterwords, we tokenized the text and unified cases (convert to lower case) and performed stemming using Snowball Stemmer[32]. After cleaning the dataset, it was ready for topic modelling.

LDA is an unsupervised leaning algorithm, Basically, it is a generative probabilistic model approach for collections of discrete data such as text corpora. [9]. LDA follows a generative story that considers each document generated by first choosing a topic distribution for this document. Moreover, each topic has its own word probability distribution.

1. Choose the number of target topics K

2. For each word w in the corpus, draw a topic from Multinomial distribution conditioned on topic

3. to optimize the assignment:

    - For each word w in d:
    - calculate probability p(topic t |document d): the probability distribution of words in document d that are assigned to topic t .
    - Compute p(word w |topic t): the probability distribution of topic assignments over documents d, from word w

4. Reassign word w a new topic t' where we choose t' according to the probability p(topic t' |document d) * P(word w |topic t')

The main goal is to balance two factors:

- For each document d, minimize the number of topics assigned to its words
- For each topic t, minimize the number of terms assigned to t

Figures 17, 18, 19, 20 shows the Wordclouds for different cases in the dataset. the Wordclouds were built using the topic words extracted from LDA algorithm by choosing on average 10 topics for each case.

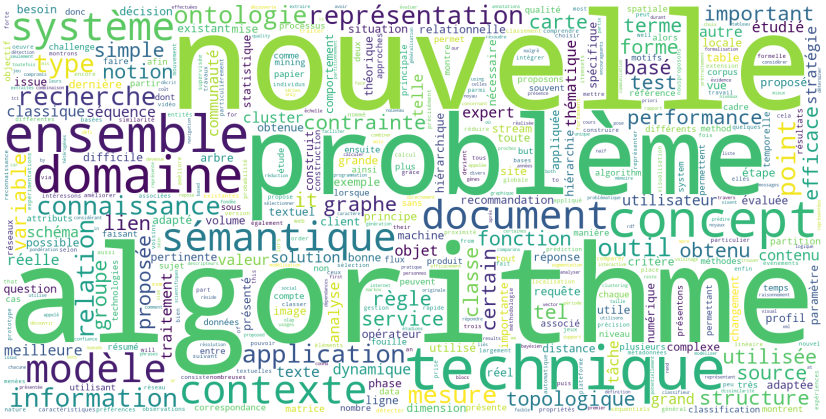

Figure 17: Wordcloud for topics in all years

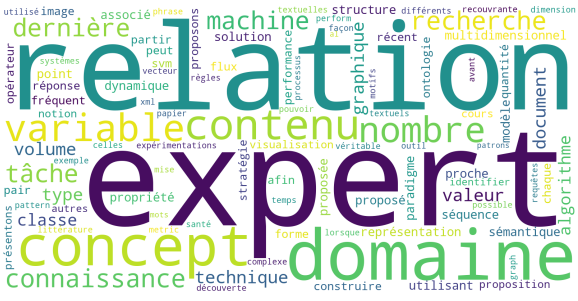

Figure 18: Wordcloud in 2010

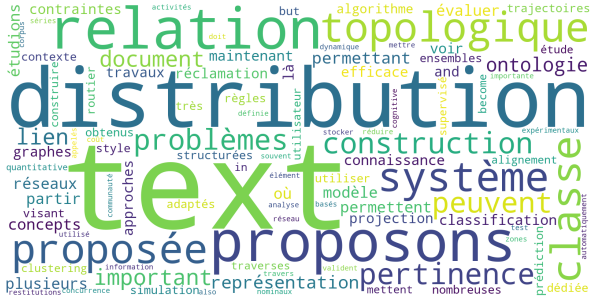

Figure 19: Wordcloud in 2013

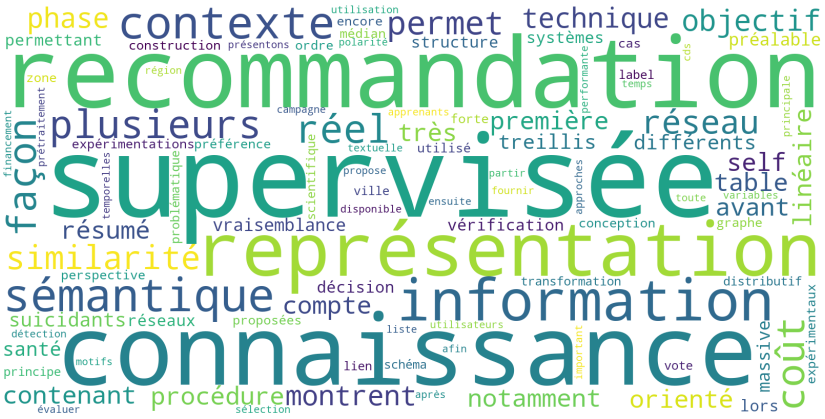

Figure 20: Wordcloud for topics in 2018

The Wordcloud visualization provides a neat way to get an abstract idea about the text data. Moreover, when combined with topic modelling it gives a more refined idea.

Evident from the above wordclouds, we are getting words like 'algorithme' and 'technique' in many topics with very high probability. Due to this, we decided to use our previously mentioned custom stopwords list. This time, we also did lemmatization. We started with 15 topics. And for this, we achieved a coherence (c_v) score of around 0.27. c_v measure lies between 0 and 1 and higher the number, the better the coherence between terms in a topic. For mathematical formulation of c_v, please refer to the Section II of [33]. To find the optimal number of topics, we, as before, used Elbow method. This time however, we looked at the coherence score vs. number of topics. The results is presented in the Figure 21

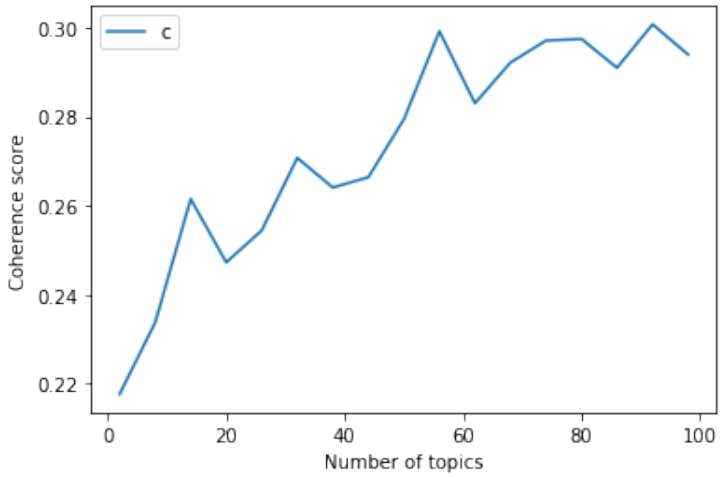

Figure 21: Coherence (c_v) vs. Number of Topics

We can see that for optimal number of topics lie beyond 90, which is not very meaningful

as we shall have many repeated words. Therefore, we chose the number of topics to be 25, where the graph had a local maxima. The visualized wordclouds are present in the Figure 22.

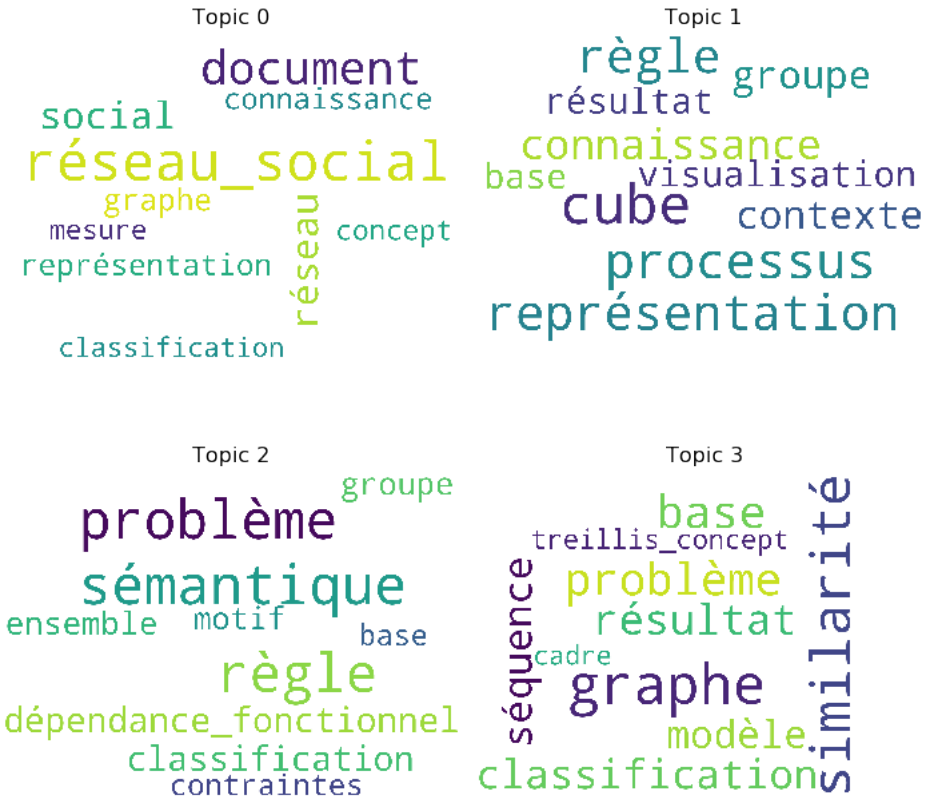

Figure 22: Word Cloud for LDA model for 4 topics.

We used PyLDAvis library [34] [35] to visualize the topics and associated terms dynamically. A screenshot is presented in the Figure 23.

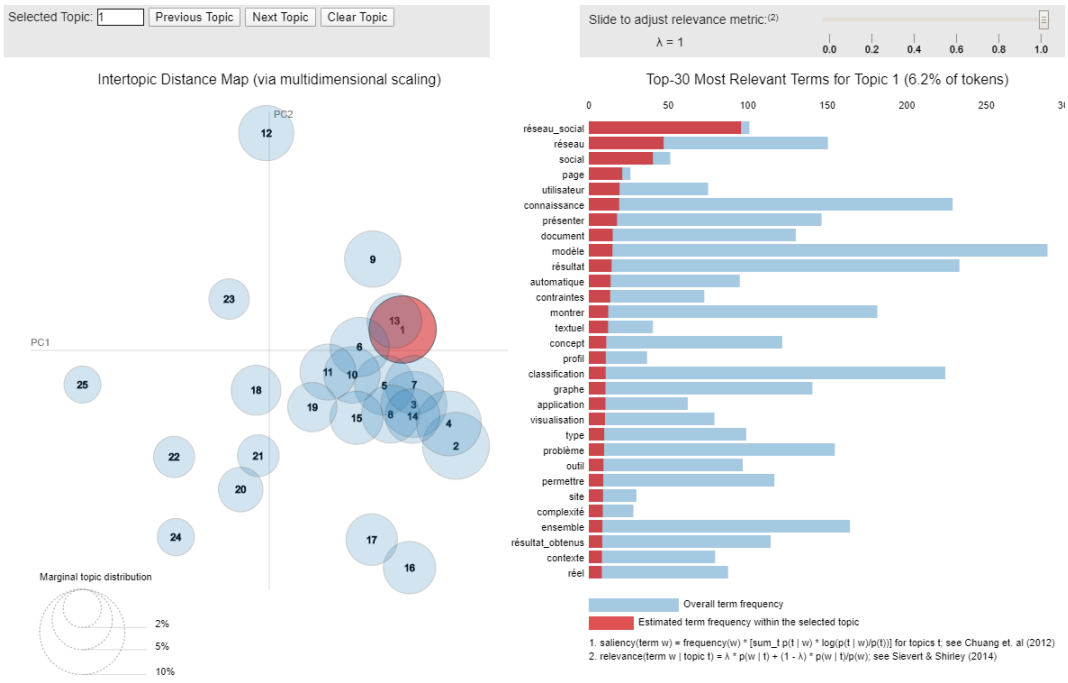

Figure 23: HTML page (Build through PyLDAvis) for visualizing our Topic Model.

6.4 Topic Evolution

To see the changes in topics over years. We built an LDA sequential model. This model gives same terms (however, probabilities may differ) per topic over different years and uses EM convergence for its generative process. The change in probabilities over the years can be seen as evolution of that term within that topic. For every topic (total 25) in our model, we created and saved csv datasets that contain terms as well as probabilties of those terms to belong to that topic for every year. A portion of these datasets looks as following:

	2004	prob_2004	2005	prob_2005	2006	prob_2006
0	réseau_social	0.04351333813697635	réseau_social	0.043787938657986816	réseau_social	0.043480745331111535
1	réseau	0.019177194976395365	réseau	0.019297098499359587	réseau	0.019319505072940092
2	social	0.017632471078113093	social	0.017731329980358188	social	0.017795618973628035
3	modèle	0.01604877761719592	modèle	0.016180949163813803	modèle	0.016292965894910763
4	relation	0.011551560311075052	relation	0.01156894582301278	relation	0.011657468351814188
5	détecter	0.006533818435417503	détecter	0.006515623174249211	détecter	0.006457124171543896
6	groupe	0.006272927543398028	groupe	0.006310110456015699	automatique	0.006298497361185085
7	automatique	0.006251446077754471	automatique	0.006265044325343253	groupe	0.006243949959801945
8	classification	0.006128916027347122	classification	0.006155910227971426	concept	0.006178297045749994
9	concept	0.006097159508991656 5	concept	0.006133057786284792	classification	0.006171089885684665
10	proposons	0.00603792011942558	proposons	0.006071023216094021	proposons	0.00611974555403541
11	domaine	0.006001083779614521	domaine	0.006005883693633736	domaine	0.006029957828307341
12	réel	0.005880806452050489	réel	0.005911661206328546	réel	0.005950775562664694
13	montrer	0.005701268906839046	montrer	0.005699878084859306	montrer	0.005715289674669973
14	ontologie	0.005692606150475508	ontologie	0.005675225140051 71	ontologie	0.005655957202357885 5

Figure 24: A portion of dataframe containing showing time evolution for terms in topic number 21.

We can also see how a topic changed over years. The Figures 25 and 26 show the evolution of some topics over time. We can see from these that terms for examples, that the probability of abstracts having the term 'arbre_décision' has been increased over time while the probability of the term 'réseau_social' has dropped.

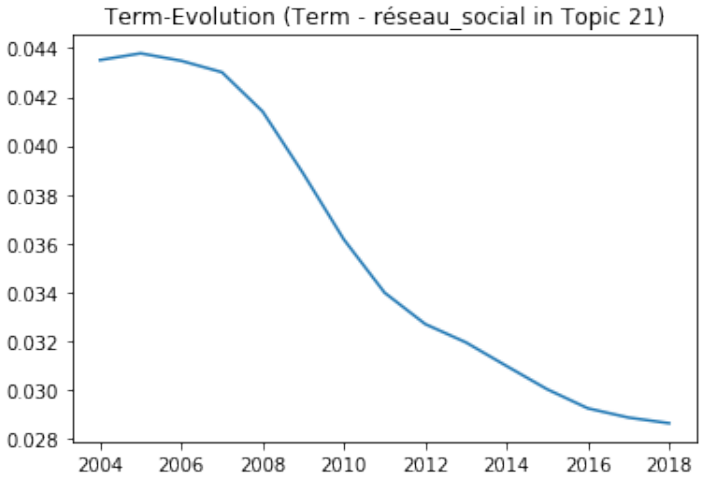

Figure 25: Evolution of term 'réseau$_s ocial' in topic 21.

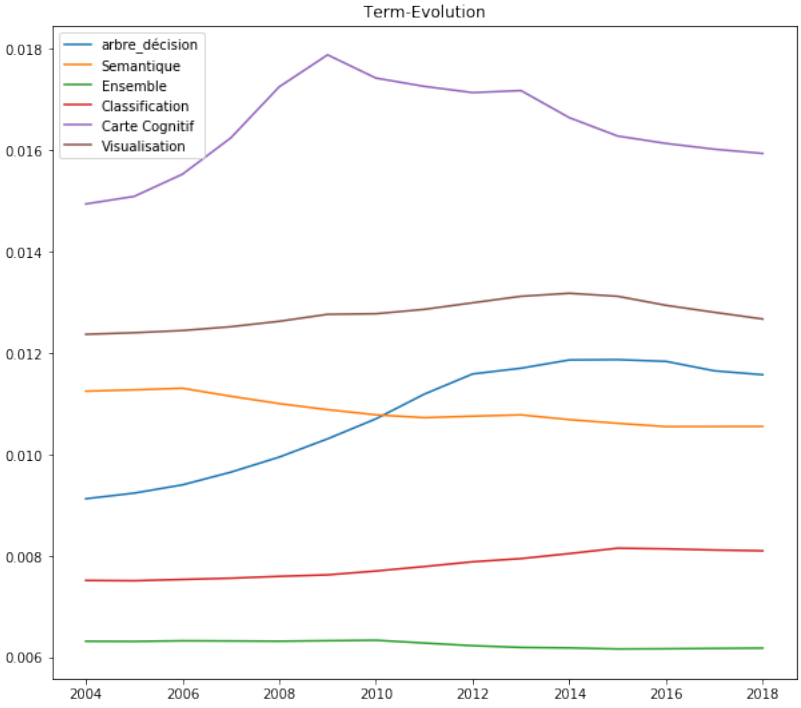

Figure 26: Evolution of terms 'arbre_décision', 'Semantique', 'Ensemble', 'Classification', 'Carte Cognitif', Visualisation'

7 Conclusion

We explored both the authors as well as abstract aspects of our dataset and we identified the authors who published most. We also investigated thoroughly about the relationships among different authors by means of exploring their collaborativeness. We used both correlation as well as clustering to identify these groups. Utilizing the MapReduce functionality, we mined the frequent item-sets. Finally we binded it all together for our analysis on authors by building a network and finding nodes with high degree centrality to understand the structure of the said network. On the second part of our studies, we followed clustering as well as topic modelling on abstracts. We used good preprocessing and we saw that we benefitted from it by our final topic model. Moreover, we also looked at the evolution of terms and topics over the two decades.

References

[1] 20ème édition de la conférence extraction et gestion des connaissances (egc). https://egc2020.sciencesconf.org/, journal=Sciencesconf.org.

[2] Jérôme David. Défi egc 2020: 20 ans d'histoire pour quel avenir? https://www.egc.asso.fr/manifestations/defi-egc/defi-egc-2020-20-ans-dhistoire-pour-quel-avenir.html.

[3] Alireza Abbasi, Kon Shing Kenneth Chung, and Liaquat Hossain. Egocentric analysis of co-authorship network structure, position and performance. *Information Processing & Management*, 48(4):671–679, 2012.

[4] Mark EJ Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the national academy of sciences*, 101(suppl 1):5200–5205, 2004.

[5] Wolfgang Glänzel and András Schubert. Analysing scientific networks through co-authorship. In *Handbook of quantitative science and technology research*, pages 257–276. Springer, 2004.

[6] Alireza Abbasi, Liaquat Hossain, and Loet Leydesdorff. Betweenness centrality as a driver of preferential attachment in the evolution of research collaboration networks. *Journal of Informetrics*, 6(3):403–412, 2012.

[7] David Laniado and Riccardo Tasso. Co-authorship 2.0: Patterns of collaboration in wikipedia. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, pages 201–210, 2011.

[8] Franc Mali, Luka Kronegger, Patrick Doreian, and Anuška Ferligoj. Dynamic scientific co-authorship networks. In *Models of science dynamics*, pages 195–232. Springer, 2012.

[9] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[10] https://colab.research.google.com/.

[11] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[12] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[13] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31(3):274–295, 2014.

[14] Michael Steinbach George Karypis, Vipin Kumar, and Michael Steinbach. A comparison of document clustering techniques. In *TextMining Workshop at KDD2000 (May 2000)*, 2000.

[15] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *European conference on principles of data mining and knowledge discovery*, pages 13–23. Springer, 2000.

[16] Carter T. Butts, Ayn Leslie-Cook, Pavel N. Krivitsky, and Skye Bender-deMoll. *networkDynamic: Dynamic Extensions for Network Objects*, 2019. R package version 0.10.0.

[17] David R Hunter, Mark S Handcock, Carter T Butts, Steven M Goodreau, and Martina Morris. ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of statistical software*, 24(3):nihpa54860, 2008.

[18] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[19] Creating co-author networks in r. `http://ds.lib.ucdavis.edu/2019/08/27/creating-co-author-networks-in-r/`, Aug 2019.

[20] Pavel L Krapivsky, Geoff J Rodgers, and Sidney Redner. Degree distributions of growing networks. *Physical Review Letters*, 86(23):5401, 2001.

[21] Stopwords-Iso. stopwords-iso/stopwords-fr. `https://github.com/stopwords-iso/stopwords-fr`, Oct 2016.

[22] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[23] Sammous. sammous/spacy-lefff. `https://github.com/sammous/spacy-lefff`, Jan 2020.

[24] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[25] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[26] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[27] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[29] sklearn.manifold.tsne. `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html`.

[30] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[31] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[32] Martin F Porter. Snowball: A language for stemming algorithms, 2001.

[33] Shaheen Syed and Marco Spruit. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In *2017 IEEE International conference on data science and advanced analytics (DSAA)*, pages 165–174. IEEE, 2017.

[34] Carson Sievert and Kenneth Shirley. Ldavis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*, pages 63–70, 2014.

[35] pyldavis. `https://pypi.org/project/pyLDAvis/`.