

A Multi-Criteria Document Clustering Method Based on Topic Modeling and Pseudoclosure Function

Quang Vu Bui

CHArt Laboratory EA 4004, Ecole Pratique des Hautes Etudes, PSL Research University, 75014, Paris, France
Hue University of Sciences, Vietnam
E-mail: quang-vu.bui@etu.ephe.fr

Karim Sayadi

CHArt Laboratory EA 4004, Ecole Pratique des Hautes Etudes, PSL Research University, 75014, Paris, France
Sorbonne University, UPMC Univ Paris 06, France
E-mail: karim.sayadi@upmc.fr

Marc Bui

CHArt Laboratory EA 4004, Ecole Pratique des Hautes Etudes, PSL Research University, 75014, Paris, France
E-mail: marc.bui@ephe.sorbonne.fr

Keywords: pretopology, pseudoclosure, clustering, k-means, latent Dirichlet allocation, topic modeling, Gibbs sampling

Received: July 1, 2016

We address in this work the problem of document clustering. Our contribution proposes a novel unsupervised clustering method based on the structural analysis of the latent semantic space. Each document in the space is a vector of probabilities that represents a distribution of topics. The document membership to a cluster is computed taking into account two criteria: the major topic in the document (qualitative criterion) and the distance measure between the vectors of probabilities (quantitative criterion). We perform a structural analysis on the latent semantic space using the Pretopology theory that allows us to investigate the role of the number of clusters and the chosen centroids, in the similarity between the computed clusters. We have applied our method to Twitter data and showed the accuracy of our results compared to a random choice number of clusters.

Povzetek: Predstavljena metoda grupira dokumente glede na semantični prostor. Eksperimenti so narejeni na podatkih s Twitterja.

1 Introduction

Classifying a set of documents is a standard problem addressed in machine learning and statistical natural language processing [13]. Text-based classification (also known as text categorization) examines the computer-readable ASCII text and investigates linguistic properties to categorize the text. When considered as a machine learning problem, it is also called statistical Natural Language Processing (NLP) [13]. In this task, a text is assigned to one or more predefined class labels (i.e. category) through a specific process in which a classifier is built, trained on a set of features and then applied to label future incoming texts. Given the labels, the task is performed within the supervised learning framework. Several Machine Learning algorithms have been applied to text classification (see [1] for a survey): Rocchio's Algorithm, N-Nearest Neighbors, Naive Bayes, Decision tree, Support Vector Machine (SVM).

Text-based features are typically extracted from the so-called word space model that uses distributional statistics to

generate high-dimensional vector spaces. Each document is represented as a vector of word occurrences. The set of documents is represented by a high-dimensional sparse matrix. In the absence of predefined labels, the task is referred as a clustering task and is performed within the unsupervised learning framework. Given a set of keywords, one can use the angle between the vectors as a measure of similarity between the documents. Depending on the algorithm, different measures are used. Nevertheless, this approach suffers from the curse of dimensionality because of the sparse matrix that represents the textual data. One of the possible solutions is to represent the text as a set of topics and use the topics as an input for a clustering algorithm.

To group the documents based on their semantic content, the topics need to be extracted. This can be done using one of the following three methods. (i) LSA [10] (Latent Semantic Analysis) uses the Singular Value Decomposition methods to decompose high-dimensional sparse matrix to three matrices: one matrix that relates words to topics, another one that relates topics to documents and a diagonal

matrix of singular value. (ii) Probabilistic LSA [8] is a probabilistic model that treats the data as a set of observations coming from a generative model. The generative model includes hidden variables representing the probability distribution of the words in the topics and the probability distribution of the topics in the words. (iii) Latent Dirichlet Allocation [4] (LDA) is a Bayesian extension of probabilistic LSA. It defines a complete generative model with a uniform prior and full Bayesian estimator.

LDA gives us three latent variables after computing the posterior distribution of the model; the topic assignment, the distribution of words in each topic and the distribution of the topics in each document. Having the distribution of topics in documents, we can use it as the input for clustering algorithms such as k-means, hierarchical clustering.

K-means uses a distance measure to group a set of data points within a predefined random number of clusters. Thus, to perform a fine-grained analysis of the clustering process we need to control the number of clusters and the distance measure. The Pretopology theory [3] offers a framework to work with categorical data, to establish a multi-criteria distance for measuring the similarity between the documents and to build a process to structure the space [11] and infer the number of clusters for k-means. We can then tackle the problem of clustering a set of documents by defining a family of binary relationships on the topic-based contents of the documents. The documents are not only grouped together using a measure of similarity but also using the pseudoclosure function built from a family of binary relationships between the different hidden semantic contents (i.e. topics).

The idea of using Pretopology theory for k-means clustering has been proposed by [16]. In this paper, the authors proposed the method to find automatically a number k of clusters and k centroids for k -means clustering by results from structural analysis of minimal closed subsets algorithm [11] and also proposed to use pseudoclosure distance constructed from the relationships family to examine the similarity measure for both numeric and categorical data. The authors illustrated the method with a toy example about the toxic diffusion between 16 geographical areas using only one relationship.

For the problem of classification, the authors of this work [2] built a vector space with Latent Semantic Analysis (LSA) and used the pseudoclosure function from Pretopological theory to compare all the cosine values between the studied documents represented by vectors and the documents in the labeled categories. A document is added to a labeled categories if it has a maximum cosine value.

Our work differs from the work of [2] and extends the method proposed in [16] with two directions: first, we exploited this idea in document clustering and integrated structural information from LDA using the pretopological concepts of pseudoclosure and minimal closed subsets introduced in [11]. Second, we showed that Pretopology theory can apply to multi-criteria clustering by defining the pseudo distance built from multi-relationships. In our pa-

per, we clustered documents by using two criteria: one based on the major topic of document (qualitative criterion) and the other based on Hellinger distance (quantitative criterion). The clustering is based on these two criteria but not on multicriteria optimization [5] for clustering algorithms. Our application on Twitter data also proposed a method to construct a network from the multi-relations network by choosing the set of relations and then applying strong or weak Pretopology.

We present our approach in a method that we named the Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM). MCPTM organizes a set of unstructured entities in a number of clusters based on multiple relationships between each two entities. Our method discovers the topics expressed by the documents, tracks changes step by step over time, expresses similarity based on multiple criteria and provides both quantitative and qualitative measures for the analysis of the document.

1.1 Contributions

The contributions of this paper are as follows.

1. We propose a new method to cluster text documents using Pretopology and Topic Modeling.
2. We investigate the role of the number of clusters inferred by our analysis of the documents and the role of the centroids in the similarity between the computed clusters.
3. We conducted experiments with different distances measures and show that the distance measure that we introduced is competitive.

1.2 Outline

The article is organized as follows: Section 2, 3 present some basic concepts such as Latent Dirichlet Allocation (Section 2) and the Pretopology theory (Section 3), Section 4 explains our approach by describing at a high level the different parts of our algorithm. In Section 5, we apply our algorithm to a corpus consisting of microblogging posts from Twitter.com. We conclude our work in Section 6 by presenting the obtained results.

2 Topic modeling

Topic Modeling is a method for analyzing large quantities of unlabeled data. For our purposes, a topic is a probability distribution over a collection of words and a topic model is a formal statistical relationship between a group of observed and latent (unknown) random variables that specifies a probabilistic procedure to generate the topics [4, 8, 6, 15]. In many cases, there exists a semantic relationship between terms that have high probability within the

same topic – a phenomenon that is rooted in the word co-occurrence patterns in the text and that can be used for information retrieval and knowledge discovery in databases.

2.1 Latent Dirichlet allocation

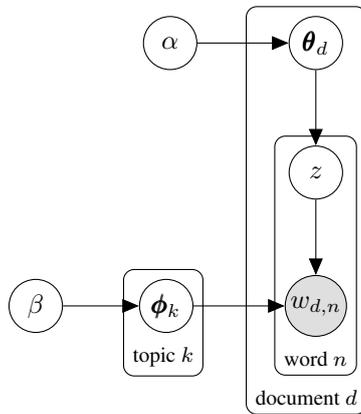


Figure 1: Bayesian Network (BN) of Latent Dirichlet Allocation.

Latent Dirichlet Allocation (LDA) by Blei et al. [4] is a generative probabilistic model for collections of grouped discrete data. Each group is described as a random mixture over a set of latent topics where each topic is a discrete distribution over the vocabulary collection. LDA is applicable to any corpus of grouped discrete data. In our work we refer to the standard Natural Language Processing (NLP) use case where a corpus is a collection of documents, and the discrete data are represented by the occurrence of words.

LDA is a probabilistic model for unsupervised learning, it can be seen as a Bayesian extension of the probabilistic Latent Semantic Analysis (pLSA) [8]. More precisely, LDA defines a complete generative model which is a full Bayesian estimator with a uniform prior while pLSA provides a Maximum Likelihood (ML) or Maximum a Posterior (MAP) estimator. For more technical details we refer to the work of Gregor Heinrich [7]. The generative model of LDA is described with the probabilistic graphical model [9] in Fig. 1.

In this LDA model, different documents d have different topic proportions θ_d . In each position in the document, a topic z is then selected from the topic proportion θ_d . Finally, a word is picked from all vocabularies based on their probabilities ϕ_k in that topic z . θ_d and ϕ_k are two Dirichlet distributions with α and β as hyperparameters. We assume symmetric Dirichlet priors with α and β having a single value.

The hyperparameters specify the nature of the priors on θ_d and ϕ_k . The hyperparameter α can be interpreted as a prior observation count of the number of times a topic z is sampled in document d [15]. The hyperparameter β can be interpreted as a prior observation count on the number of times words w are sampled from a topic z [15].

The advantage of the LDA model is that interpreting at the topic level instead of the word level allows us to gain more insights into the meaningful structure of documents since noise can be suppressed by the clustering process of words into topics. Consequently, we can use the topic proportion in order to organize, search, and classify a collection of documents more effectively.

2.2 Inference with Gibbs sampling

In this subsection, we specify a topic model procedure based on the Latent Dirichlet Allocation (LDA) and Gibbs Sampling.

The key problem in Topic Modeling is posterior inference. This refers to reversing the defined generative process and learning the posterior distributions of the latent variables in the model given the observed data. In LDA, this amounts solving the following equation:

$$p(\theta, \phi, z | w, \alpha, \beta) = \frac{p(\theta, \phi, z, w | \alpha, \beta)}{p(w | \alpha, \beta)} \quad (1)$$

Unfortunately, this distribution is intractable to compute [7]. The normalization factor in particular, $p(w | \alpha, \beta)$, cannot be computed exactly. However, there are a number of approximate inference techniques available that we can apply to the problem including variational inference (as used in the original LDA paper [4]) and Gibbs Sampling that we shall use.

For LDA, we are interested in the proportions of the topic in a document represented by the latent variable θ_d , the topic-word distributions $\phi^{(z)}$, and the topic index assignments for each word z_i . While conditional distributions - and therefore an LDA Gibbs Sampling algorithm - can be derived for each of these latent variables, we note that both θ_d and $\phi^{(z)}$ can be calculated using just the topic index assignments z_i (i.e. z is a sufficient statistic for both these distributions). Therefore, a simpler algorithm can be used if we integrate out the multinomial parameters and simply sample z_i . This is called a collapsed Gibbs sampler [6, 15].

The collapsed Gibbs sampler for LDA needs to compute the probability of a topic z being assigned to a word w_i , given all other topic assignments to all other words. Somewhat more formally, we are interested in computing the following posterior up to a constant:

$$p(z_i | z_{-i}, \alpha, \beta, w) \quad (2)$$

where z_{-i} means all topic allocations except for z_i .

Equation 3 shows how to compute the posterior distribution for topic assignment.

$$P(z_i = j | z_{-i}, w) \propto \frac{n_{-i,j}^{w_i} + \beta}{n_{-i,j}^{(\cdot)} + V\beta} \frac{n_{-i,j}^{d_i} + \alpha}{n_{-i,j}^{d_i} + K\alpha} \quad (3)$$

where $n_{-i,j}^{w_i}$ is the number of times word w_i was related to topic j . $n_{-i,j}^{(\cdot)}$ is the number of times all other words

Algorithm 1 The LDA Gibbs sampling algorithm.

Require: words $w \in$ corpus $\mathcal{D} = (d_1, d_2, \dots, d_M)$

- 1: **procedure** LDA-GIBBS(w, α, β, T)
- 2: randomly initialize z and increment counters
- 3: **loop** for each iteration
- 4: **loop** for each word w in corpus \mathcal{D}
- 5: **Begin**
- 6: word $\leftarrow w[i]$
- 7: $tp \leftarrow z[i]$
- 8: $n_{d, tp}^- = 1; n_{word, tp}^- = 1; n_{tp}^- = 1$
- 9: **loop** for each topic $j \in \{0, \dots, K - 1\}$
- 10: compute $P(z_i = j | z_{-i}, w)$
- 11: $tp \leftarrow \text{sample from } p(z|\cdot)$
- 12: $z[i] \leftarrow tp$
- 13: $n_{d, tp}^+ = 1; n_{word, tp}^+ = 1; n_{tp}^+ = 1$
- 14: **End**
- 15: Compute $\phi^{(z)}$
- 16: Compute θ_d
- 17: **return** $z, \phi^{(z)}, \theta_{\mathcal{D}}$ ▷ Output
- 18: **end procedure**

were related with topic j . $n_{-i, j}^{d_i}$ is the number of times topic j was related with document d_i . The number of times all other topics were related with document d_i is annotated with $n_{-i, \cdot}^{d_i}$. Those notations were taken from the work of Thomas Griffiths and Mark Steyvers [6].

$$\hat{\phi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(\cdot)} + V\beta} \tag{4}$$

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_{\cdot}^{(d)} + K\alpha} \tag{5}$$

Equation 4 is the Bayesian estimation of the distribution of the words in a topic. Equation 5 is the estimation of the distribution of topics in a document.

3 Pretopology theory

The Pretopology is a mathematical modeling tool for the concept of proximity. It was first developed in the field of social sciences for analyzing discrete spaces [3]. The Pretopology establishes powerful tools for conceiving a process to structure the space and infer the number of clusters for example. This is made possible by ensuring a follow-up of the process development of dilation, alliance, adherence, closed subset and acceptability [16, 12].

3.1 Pseudoclosure

Let consider a nonempty set E and $\mathcal{P}(E)$ which designates all the subsets of E .

Definition 1. A pseudoclosure $a(\cdot)$ is a mapping from $\mathcal{P}(E)$ to $\mathcal{P}(E)$, which satisfies following two conditions:

$$a(\emptyset) = \emptyset; \forall A \subset E, A \subset a(A) \tag{6}$$

A pretopological space (E, a) is a set E endowed with a pseudoclosure function $a(\cdot)$.

Subset $a(A)$ is called the pseudoclosure of A . As $a(a(A))$ is not necessarily equal to $a(A)$, a sequential appliance of pseudoclosure on A can be used to model expansions: $A \subset a(A) \subset a(a(A)) = a^2(A) \subset \dots \subset a^k(A)$

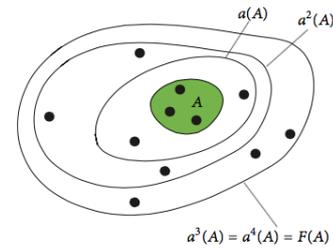


Figure 2: Iterated application of the pseudoclosure map leading to the closure.

Definition 2. Let (E, a) a pretopological space, $\forall A, A \subset E$. A is a closed subset if and only if $a(A) = A$.

Definition 3. Given a pretopological space (E, a) , call the closure of A , when it exists, the smallest closed subset of (E, a) which contains A . The closure of A is denoted by $F(A)$.

Remark:

- $F(A)$ is the intersection of all closed subsets which contain A . In the case where (X, a) is a “general” pretopological space, the closure may not exist.
- Closure is very important because of the information it gives about the “influence” or “reachability” of a set, meaning, for example, that a set A can influence or reach elements into $F(A)$, but not further (see Figure 2).

Hence, it is necessary to build a pretopological spaces in which the closure always exists. V -type pretopological spaces are the most interesting cases.

Definition 4. A Pretopology space (E, a) is called V -type space if and only if

$$\forall A \subset E, \forall B \subset E, (A \subset B) \Rightarrow (a(A) \subset a(B)) \tag{7}$$

Proposition 1. In any pretopological space of type V , given a subset A of E , the closure of A always exists.

The other reason why we use the spaces of type V is that we can build them from a family of reflexive binary relations on the finite set E . That thus makes it possible to take various points of view (various relations) expressed in a qualitative way to determine the pretopological structure placed on E . So, it can be applied on multi-criteria clustering or multi-relations networks.

3.2 Pretopology and binary relationships

Suppose we have a family $(R_i)_{i=1, \dots, n}$ of binary reflexive relationships on a finite set E . Let us consider $\forall i = 1, 2, \dots, n, \forall x \in E, V_i(x)$ defined by:

$$V_i(x) = \{y \in E | x R_i y\} \tag{8}$$

Then, the pseudoclosure $a_s(\cdot)$ is defined by:

$$a_s(A) = \{x \in E | \forall i = 1, 2, \dots, n, V_i(x) \cap A \neq \emptyset\} \tag{9}$$

Pretopology defined on E by $a_s(\cdot)$ using the intersection operator is called the strong Pretopology induced by the family $(R_i)_{i=1,\dots,n}$.

Similarly, we can define weak Pretopology from $a_w(\cdot)$ by using the union operator:

$$a_w(A) = \{x \in E \mid \exists i = 1, 2, \dots, n, V_i(x) \cap A \neq \emptyset\} \quad (10)$$

Proposition 2. $a_s(\cdot)$ and $a_w(\cdot)$ determine on E a pretopological structure and the spaces (E, a_s) , (E, a_w) are of V -type.

3.3 Minimal closed subsets

We denote \mathcal{F}_e as the family of elementary closed subsets, the set of closures of each singleton $\{x\}$ of $P(E)$. So in a V -type pretopological space, we get:

- $\forall x \in E, \exists F_x : \text{closure of } \{x\}$.
- $\mathcal{F}_e = \{F_x \mid x \in E\}$

Definition 5. F_{min} is called a minimal closed subset if and only if F_{min} is a minimal element for inclusion in \mathcal{F}_e .

We denote $\mathcal{F}_m = \{F_{m_j}, j = 1, 2, \dots, k\}$, the family of minimal closed subsets, the set of minimal closed subsets in \mathcal{F}_e .

4 Our approach

In our approach, we build The Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM) which clusters documents via Topic Modeling and pseudoclosure. MCPTM can be built by:

1. Defining the topic-distribution of each document d_i in corpus \mathcal{D} by document structure analysis using LDA.
2. Defining two binary relationships: R_{MTP} based on major topic and R_{d_H} based on Hellinger distance.
3. Building the pseudoclosure function from two binary relationships R_{MTP}, R_{d_H} .
4. Building the pseudoclosure distance from pseudoclosure function.
5. Determining initial parameters for the k-means algorithm from results of minimal closed subsets.
6. Using the k -means algorithm to cluster sets of documents with initial parameters from the result of minimal closed subsets, the pseudoclosure distance to compute the distance between two objects and the inter-pseudoclosure distance to re-compute the new centroids.

4.1 Document structure analysis by LDA

A term-document matrix is given as an input to LDA and it outputs two matrices:

- The document-topic distribution matrix θ .
- The topic-term distribution matrix ϕ .

The topic-term distribution matrix $\phi \in \mathbf{R}^{K \times V}$ consists of K rows, where the i -th row $\phi_i \in \mathbf{R}^V$ is the word distribution of topic i . The terms with high ϕ_{ij} values indicate that they are the representative terms of topic i . Therefore, by looking at such terms one can grasp the meaning of each topic without looking at the individual documents in the cluster.

In a similar way, the document-topics distributions matrix $\theta \in \mathbf{R}^{M \times K}$ consists of M rows, where the i -th row $\theta_i \in \mathbf{R}^K$ is the topic distribution for document i . A high probability value of θ_{ij} indicates that document i is closely related to topic j . In addition, documents with low θ_{ij} values over all the topics are noisy documents that belong to none of the topics. Therefore, by looking at the θ_{ij} values, one can understand how closely the document is related to the topic.

4.2 Defining binary relationships

By using LDA, each document may be characterized by its topic distribution and also be labeled by the topic with the highest probability. In this subsection, we use this information to define the relations between two documents based on the way we consider the "similarity" between them.

4.2.1 Based on major topic

Firstly, based on the label information, we can consider connecting the documents if they have the same label. However, in some cases such as noisy documents, the probability of label topic is very small and it is not really good if we use this label to represent a document. Hence, we just use the label information if its probability is higher than threshold p_0 . We define the major topic of each document as:

Definition 6. $MTP(d_i)$ is the major topic of document d_i if $MTP(d_i)$ is the topic with highest probability in the topic distribution of document d_i and this probability is greater than threshold p_0 , $p_0 \geq 1/K$, K is the number of topic.

$$MTP(d_i) = \{k \mid \theta_{ik} = \max_j \theta_{ij} \text{ and } \theta_{ik} \geq p_0\}.$$

Considering two documents d_m, d_n with their major topic $MTP(d_m), MTP(d_n)$, we see that document d_m is close to document d_n if they have the same major topic. So, we proposed a definition of binary relationship R_{MTP} of two documents based on their major topic as:

Definition 7. Document d_m has binary relationship R_{MTP} with document d_n if d_m and d_n have the same major topic.

4.2.2 Based on Hellinger distance

Secondly, we can use the topic distributions of documents to define the relation based the similarity between two real number vectors or two probability distributions. If we consider a probability distribution as a vector, we can choose some distances or similarity measures related to the vector distance such as Euclidean distance, Cosine Similarity, Jacard Coefficient, Pearson Correlation Coefficient, etc. But, it is better if we choose distances or similarity measures related to the probability distribution such as Kullback-Leibler Divergence, Bhattacharyya distance, Hellinger distance, etc. We choose the Hellinger distance because it is a metric for measuring the deviation between two probability distributions, easily to compute and especially limited in $[0, 1]$.

Definition 8. For two discrete probability distributions $P = (p_1, \dots, p_k)$ and $Q = (q_1, \dots, q_k)$, their Hellinger distance is defined as

$$d_H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^k (\sqrt{p_i} - \sqrt{q_i})^2}, \quad (11)$$

The Hellinger distance is directly related to the Euclidean norm of the difference of the square root vectors, i.e.

$$d_H(P, Q) = \frac{1}{\sqrt{2}} \|\sqrt{P} - \sqrt{Q}\|_2.$$

The Hellinger distance satisfies the inequality of $0 \leq d_H \leq 1$. This distance is a metric for measuring the deviation between two probability distributions. The distance is 0 when $P = Q$. Disjoint P and Q shows the maximum distance of 1.

The lower the value of the Hellinger distance, the smaller the deviation between two probability distributions. So, we can use the Hellinger distance to measure the similarity between two documents d_m, d_n . We then define the binary relationship R_{d_H} between two documents as:

Definition 9. Document d_m has binary relationship R_{d_H} with document d_n if $d_H(d_m, d_n) \leq d_0, 0 \leq d_0 \leq 1, d_0$ is the accepted threshold.

4.3 Building pseudoclosure function

Based on two binary relationships R_{MTP} and R_{d_H} , we can build the neighborhood basis (see Algorithm 2) and then build the pseudoclosures (see Algorithm 3) for strong (with intersection operator) and weak (with union operator) Pre-topology.

4.4 Building pseudoclosure distance

In standard *k-means*, the centroid of a cluster is the average point in the multidimensional space. Its coordinates are the arithmetic mean for each dimension separately over all the

Algorithm 2 Neighborhood Basis Using Topic Modeling.

Require: document-topic distribution matrix θ , corpus \mathcal{D}
Require: R_{MTP}, R_{d_H} : family of relations.
1: **procedure** NEIGHBORHOOD-TM($\mathcal{D}, \theta, R_{MTP}, R_{d_H}$)
2: **loop** for each relation $R_i \in \{R_{MTP}, R_{d_H}\}$
3: **loop** for each document $d_m \in \mathcal{D}$
4: **loop** for each document $d_n \in \mathcal{D}$
5: **if** $R_i(d_m, d_n)$ **then**
6: $B_i[d_m].append(d_n)$
7: **return** $B = [B_1, B_2]$ ▷ Output
8: **end procedure**

Algorithm 3 Pseudoclosure using Topic Modeling.

Require: $B = (B_1, B_2), \mathcal{D} = \{d_1, \dots, d_M\}$
1: **procedure** PSEUDOCLOSURE(A, B, \mathcal{D})
2: $aA = A$
3: **loop** for each document $d_n \in \mathcal{D}$
4: **if** $(A \cap B_1[d_n] \neq \emptyset \text{ or } A \cap B_2[d_n] \neq \emptyset)$ **then**
5: $aA.append(d_n)$
6: **return** aA ▷ Output
7: **end procedure**

points in the cluster which are not effective with categorical data analysis. On the other hand, the pseudoclosure distance is used to examine the similarity using both numeric and categorical data. Therefore, it can contribute to improving the classification with *k-means*.

Definition 10. We define $\delta(A, B)$ pseudoclosure distance between two subsets A and B of a finite set E :

$$k_0 = \min(\min\{k | A \subset a^k(B)\}, \infty)$$

$$k_1 = \min(\min\{k | B \subset a^k(A)\}, \infty)$$

$$\delta(A, B) = \min(k_0, k_1)$$

where $a^k(\cdot) = a^{k-1}(a(\cdot))$

Definition 11. We call $D_A(x)$ interior-pseudo-distance of a point x in a set A :

$$D_A(x) = \frac{1}{|A|} \sum_{y \in A} \delta(x, y).$$

In case where A and B are reduced to one element x and y , we get the distance $\delta(x, y)$. For clustering documents with *k-means* algorithm, we use the pseudoclosure distance $\delta(x, y)$ to compute distance between two documents (each document represented by its topic distribution is a point $x \in E$) and the interior-pseudo-distance $D_A(x)$ to compute centroid of A (x_0 is chosen as centroid of A if $D_A(x_0) = \min_{x \in A} D_A(x)$).

4.5 Structure analysis with minimal closed subsets

The two limits of the standard *k-means* algorithm are the number of clusters which must be predetermined and the randomness in the choice of the initial centroids of the clusters. Pretopology theory gives a good solution to omit these limits by using the result from minimal closed subsets. The algorithm to compute minimal closed subset is presented in algorithm 4.

Algorithm 4 Minimal closed subsets algorithm.

Require: corpus \mathcal{D} , pseudoclosure $aA()$

```

1: procedure MINIMAL-CLOSED-SUBSETS( $\mathcal{D}$ ,  $aA()$ )
2:   compute family of elementary closed subsets  $\mathcal{F}_e$ 
3:    $\mathcal{F}_m = \emptyset$ 
4:   loop until  $\mathcal{F}_e = \emptyset$ 
5:     Begin
6:       Choose  $F \subset \mathcal{F}_e$ 
7:        $\mathcal{F}_e = \mathcal{F}_e - F$ 
8:       minimal = True
9:        $\mathcal{F} = \mathcal{F}_e$ 
10:      loop until  $\mathcal{F} = \emptyset$  and not minimal
11:        Begin
12:          Choose  $G \in \mathcal{F}$ 
13:          If  $G \subset F$  then
14:            minimal=False
15:          Else
16:            If  $F \subset G$  then
17:               $\mathcal{F}_e = \mathcal{F}_e - \{G\}$ 
18:               $\mathcal{F} = \mathcal{F} - G$ 
19:            End
20:          End
21:        If minimal = True &&  $F \notin \mathcal{F}_m$  then
22:           $\mathcal{F}_m = \mathcal{F}_m \cup F$ 
23:        return  $\mathcal{F}_m$ 
24: end procedure

```

▷ Output

By performing the minimal closed subset algorithm, we get the family of minimal closed subsets. This family, by definition, characterizes the structure underlying the data set E . So, the number of minimal closed subsets is a quite important parameter: it gives us the number of clusters to use in the k -means algorithm. Moreover, the initial centroids for starting the k -means process can be determined by using the interior-pseudo-distance for each minimal closed subset $F_{m_j} \in \mathcal{F}_m$ (x_0 is chosen as centroid of F_{m_j} if $D_{F_{m_j}}(x_0) = \min_{x \in F_{m_j}} D_{F_{m_j}}(x)$).

4.6 MCPTM algorithm

In this subsection, we present The Method of Clustering Documents using Pretopology and Topic Modeling (MCPTM) which clusters documents via the Topic Modeling and pseudoclosure. At first, an LDA Topic Modeling is learned on the documents to achieve topic-document distributions. The major topic and Hellinger probability distance are used to define relations between documents and these relations are used to define a pretopological space which can be employed to get preliminarily clusters of a corpus and determine the number of clusters. After that, k -means clustering algorithm is used to cluster the documents data with pseudodistance and inter-pseudodistance. The MCPTM algorithm is presented in algorithm 5.

4.7 Implementation in python of the library AMEUR

In this part, we briefly present our *AMEUR* library written in python. *AMEUR* is a project connecting the tools that come from the framework of Pretopology, Topic Modeling, multi-relations networks analysis and semantic relationship. The library is composed of the following modules: *Pretopology*, *topicmodeling* and *nlp*.

Algorithm 5 The MCPTM algorithm: clustering documents using Pretopology and Topic Modeling.

Require: \mathcal{D} : corpus from set of documents

```

1: procedure MCPTM( $\mathcal{D}$ )
2:    $\theta_{\mathcal{D}} \leftarrow \text{LDA-GIBBS}(\mathcal{D}, \alpha, \beta, T)$ 
3:    $B \leftarrow \text{NEIGHBORHOOD-TM}(\mathcal{D}, \theta_{\mathcal{D}}, R_{MTP}, R_{dH})$ 
4:    $aA \leftarrow \text{pseudoCLOSURE}(B)$ 
5:    $\mathcal{F}_m \leftarrow \text{MINIMAL-CLOSED-SUBSETS}(\mathcal{D}, aA())$ 
6:    $k = |\mathcal{F}_m|$ : number of clusters
7:    $M = \{m_i\}_{i=1, \dots, k}$ ,  $m_i = \text{Centroid}(F_{m_i})$ 
8:   while clusters centroids changed do
9:     for each  $x \in E - M$  do
10:      compute  $\delta(x, m_i)$ ,  $i = 1, \dots, k$ 
11:      find  $m_0$  with  $\delta(x, m_0) = \min_{i=1, \dots, k} \delta(x, m_i)$ 
12:       $F_{m_0} = F_{m_0} \cup \{x\}$ 
13:     end for
14:     Recompute clusters centroids  $M$ .
15:   end while
16:   return  $Clusters = \{F_1, F_2, \dots, F_k\}$ 
17: end procedure

```

▷ Output

The *Pretopology* module implements the functions described in section III. The implementation of the Pretopology in the *AMEUR* library allows us to ensure the follow-up of step-by-step processes like dilatation, alliance, pseudoclosure, closure, family of minimal closed subsets, MCPTM and acceptability in multi-relations networks.

The *topicmodeling* module implements generative models like the Latent Dirichlet Allocation, LDA Gibbs Sampling that allows us to capture the relationships between discrete data. This module is used within the *AMEUR* library for querying purposes e.g. to retrieve a set of documents that are relevant to a query document or to cluster a set of documents given a latent topic query. These computations of these queries are ensured by the connection between the *topicmodeling* module and the *Pretopology* module.

The *nlp* (natural language processing) module implements the necessary functions for getting unstructured text data of different sources from web pages or social medias and preparing them as proper inputs for the algorithms implemented in other modules of the library.

5 Application and Evaluation

The microblogging service Twitter has become one of the major micro-blogging websites, where people can create and exchange content with a large audience. In this section, we apply the MCPTM algorithm for clustering a set of users around their interests. We have targeted 133 users and gathered their tweets in 133 documents. We have cleaned them and run the *LDA Gibbs Sampling* algorithm to define the topics distribution of each document and words distribution of each topic. We have used then, the *MCPTM* algorithm to automatically detect the different communities for clustering users. We present in the following, the latter steps in more details.

Table 1: Words - Topic distribution ϕ and the related users from the θ distribution

Topic 3					Topic 10				
Words	Prob.	Users	ID	Prob.	Words	Prob.	Users	ID	Prob.
paris	0.008	GStephanopoulos	42	0.697	ces	0.010	bxchen	22	0.505
charliehebdo	0.006	camanpour	23	0.694	people	0.007	randizuckerberg	102	0.477
interview	0.006	AriMelber	12	0.504	news	0.006	NextTechBlog	88	0.402
charlie	0.005	andersoncooper	7	0.457	media	0.006	lheron	71	0.355
attack	0.005	brianstelter	20	0.397	tech	0.006	LanceUlanoff	68	0.339
warisover	0.004	yokoono	131	0.362	apple	0.006	MarcusWohlsen	74	0.339
french	0.004	piersmorgan	96	0.348	facebook	0.005	marissamay	76	0.334
today	0.004	maddow	72	0.314	yahoo	0.005	harrymccracken	43	0.264
news	0.004	BuzzFeedBen	21	0.249	app	0.005	dens	33	0.209
police	0.003	MichaelSteele	81	0.244	google	0.004	nickbilton	89	0.204

Table 2: Topics - document distribution θ

User ID 02		User ID 12		User ID 22		User ID 53		User ID 75		User ID 83	
Topic	Prob.										
10	0.090	3	0.504	10	0.506	17	0.733	19	0.526	8	0.249
16	0.072	19	0.039	3	0.036	1	0.017	2	0.029	0	0.084
12	0.065	10	0.036	19	0.034	18	0.016	3	0.029	11	0.06
18	0.064	15	0.035	14	0.031	13	0.016	5	0.028	7	0.045
0	0.058	13	0.032	4	0.03	11	0.015	105	0.028	12	0.043

5.1 Data collection

Twitter is a micro-blogging social media website that provides a platform for the users to post or exchange text messages of 140 characters. Twitter provides an API that allows easy access to anyone to retrieve at most a 1% sample of all the data by providing some parameters. In spite of the 1% restriction, we are able to collect large data sets that contain enough text information for Topic Modeling as shown in [14].

The data set contains tweets from the 133 most famous and most followed public accounts. We have chosen these accounts because they are characterized by the heterogeneity of the tweets they posts. The followers that they aim to reach comes from different interest areas (i.e. politics, technology, sports, art, etc..). We used the API provided by Twitter to collect the messages of 140 characters between January and February 2015. We gathered all the tweets from a user into a document.

5.2 Data pre-processing

Social media data and mainly Twitter data is highly unstructured: typos, bad grammar, the presence of unwanted content, for example, humans expressions (happy, sad, excited, ...), URLs, stop words (the, a, there, ...). To get good insights and to build better algorithms it is essential to play with clean data. The pre-processing step gets the textual data clean and ready as input for the MCPTM algorithm.

5.3 Topic modeling results

After collecting and pre-processing data, we obtained data with 133 documents, 158,578 words in the corpus which averages 1,192 words per document and 29,104 different words in the vocabulary. We run LDA Gibbs Sampling from algorithm 1 and received the output with two matrices: the document-topic distribution matrix θ and the dis-

Table 3: Classifying documents based on their major topic

Major Topic	prob ≥ 0.3	0.15 < prob < 0.3
Topic 0	112,85,104	-
Topic 1	44,129,114	61
Topic 2	101,108,91	90
Topic 3	42,23,12,7,20, 131,96,72	21,81,93,10
Topic 4	125,36,123,0	-
Topic 5	82,126	62
Topic 6	127,37,26	92
Topic 7	118,106,32	70,4
Topic 8	113	83,55,59
Topic 9	67,122	111,100
Topic 10	22,102,88,71,74, 68,76	43,89,33,65
Topic 11	54,51,121	29,94
Topic 12	50	12
Topic 13	16,35	38
Topic 14	31,98	-
Topic 15	66,73,34,	48
Topic 16	99	-
Topic 17	53,30	-
Topic 18	47,128,1,124,5	78,115
Topic 19	14,80,39,75,18,103	-
None	remaining users (probability < 0.15)	

tribution of terms in topics represented by the matrix ϕ . We present in Table 1 two topics from the list of 20 topics that we have computed with our LDA implementation. A topic is presented with a distribution of words. For each topic, we have a list of users. Each user is identified with an ID from 0 to 132 and is associated with a topic by an order of probabilities. The two lists of probabilities in topic 3, 10 are extracted respectively from θ and ϕ distributions. The topic 3 and topic 10 are of particular interest due to the important number of users that are related to them. Topic 3 is about the terrorist attack that happened in Paris and topic 10 is about the international Consumer Electronics Show (CES). Both events happened at the same time that we collected our data from Twitter. We note that we have more users for these topics than from other ones. We can conclude that these topics can be considered as hot topics at this moment.

Due to the lack of space, we could not present in details

all the topics with their distribution of words and all topic distributions of documents. Therefore, we presented six topic distributions θ_i (sorted by probability) of six users in the table 2. A high probability value of θ_{ij} indicates that document i is closely related to topic j . Hence, user ID 12 is closely related to topic 3, user ID 22 closely related to topic 10, etc. In addition, documents with low θ_{ij} values over all the topics are noisy documents that belong to none of the topics. So, there is no major topic in user ID 02 (the max probability < 0.15).

We show in Table 3 clusters of documents based on their major topics in two levels with their probabilities. The documents with the highest probability less than 0.15 are considered noisy documents and clustered in the same cluster.

5.4 Results from the k-means algorithm using Hellinger distance

After receiving the document-topic distribution matrix θ from LDA Gibbs Sampling, we used the k-means algorithm with Hellinger distance to cluster users. The table 4 presents the result from the k-means algorithm using Hellinger distance with a number of clusters $k=13$ and random centroids. Based on the mean value of each cluster, we defined the major topic related to the clusters and attached these values in the table. We notice that different choices of initial seed sets can result in very different final partitions.

Table 4: Result from k-means algorithm using Hellinger distance

Cluster	Users	Major Topic
1	67, 111, 122	TP 9 (0.423)
2	34, 48, 66, 73	TP 15 (0.315)
3	10, 22, 33, 43, 65, 68, 71, 74, 76, 88, 89, 98, 102	TP 10 (0.305)
4	26, 92	TP 6 (0.268)
5	16, 35, 44, 90, 91, 101, 108, 114, 129	TP 2 (0.238)
6	4, 32, 70, 106, 118	TP 7 (0.345)
7	37, 127	TP 6 (0.580)
8	14, 18, 39, 75, 80, 103	TP 19 (0.531)
9	1, 5, 47, 78, 124, 128	TP 18 (0.453)
10	30, 53	TP 17 (0.711)
11	7, 12, 20, 21, 23, 42, 72, 81, 93, 96, 131	TP 3 (0.409)
12	0, 31, 36, 82, 123, 125	TP 4 (0.310)
13	remaining users	None

5.5 Results from the MCPTM algorithm

After getting the results (e.g table 2) major our LDA implementation, we defined two relations between two documents, the first based on their major topic R_{MTP} and the second based their Hellinger distance R_{d_H} . We then built the weak pseudoclosure with these relations and applied it to compute pseudoclosure distance and the minimal closed subsets. With this pseudoclosure distance, we can use the MCPTM algorithm to cluster sets of users with multi-relationships.

Figure 4 shows the number of elements of minimal closed subsets with different thresholds p_0 for R_{MTP} and

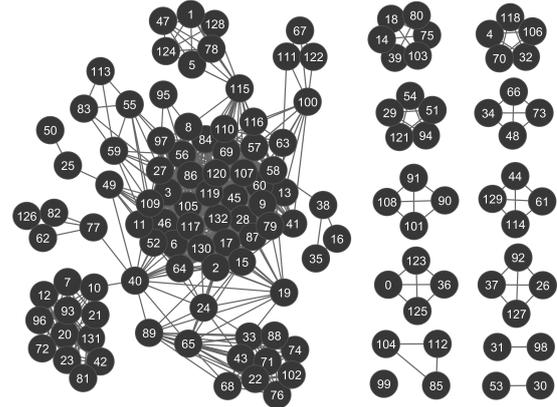


Figure 3: Network for 133 users with two relationships based on Hellinger distance ($distance \leq 0.15$) and Major topic (probability ≥ 0.15).

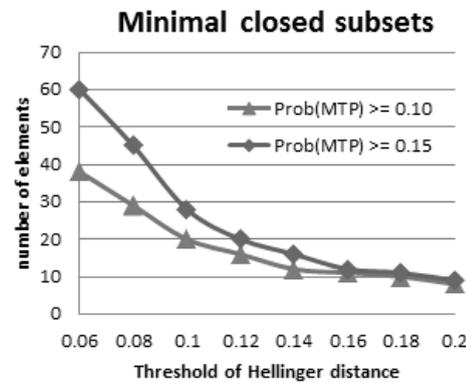


Figure 4: Number of elements of Minimal closed subsets with difference thresholds p_0 for R_{MTP} and d_0 for R_{d_H} .

d_0 for R_{d_H} . We used this information to choose the number of clusters. For this example, we chose $p_0 = 0.15$ and $d_0 = 0.15$ i.e user i connects with user j if they have the same major topic (with probability ≥ 0.15) or the Hellinger distance $d_H(\theta_i, \theta_j) \leq 0.15$. From the network (figure 3) for 133 users built from the weak pseudoclosure, we chose the number of clusters $k = 13$ since the network has 13 connected components (each component represents an element of the minimal closed subset). We used inter-pseudoclosure distance to compute initial centroids and received the result:

$$\{0, 52, 4, 14, 26, 29, 30, 31, 34, 44, 85, 90, 99\}$$

Table 5 presents the results of the MCPTM algorithm and the k -means algorithm using Hellinger distance. We notice that there is almost no difference between the results from two methods when using the number of clusters k and initial centroids above.

We saw that the largest connected component in the users network (fig. 3) has many nodes with weak ties. This component represents the cluster 13 with 89 elements. It contains the 8 remaining topics that were nonsignificant

Table 5: Result from k-means algorithm using Hellinger distance and MCPTM

Cluster	K-means & Hellinger		MCPTM Algorithm	
	Users	Topic	Users	Topic
1	0,36,123,125	TP 4 (0.457)	0,36,123,125	TP 4
2	4,32,70,10,118	TP 7 (0.345)	4,32,70,10,118	TP 7
3	14,18,39,75,80,103	TP 19 (0.531)	14,18,39,75,80,103	TP 19
4	26,37,92,127	TP 6 (0.424)	26,37,92,127	TP 6
5	29,51,54,94,121	TP 11 (0.345)	29,51,54,94,121	TP 11
6	30,53	TP 17 (0.711)	30,53	TP 17
7	31	TP 14 (0.726)	31,98	TP 14
8	34,48,66,73	TP 15 (0.315)	34,48,66,73	TP 15
9	44,61,114,129	TP 1 (0.413)	44,61,114,129	TP 1
10	85,104,112	TP 0 (0.436)	85,104,112	TP 0
11	67,90,91,101,108	TP 2 (0.407)	90,91,101,108	TP 2
12	99	TP 16 (0.647)	99	TP 16
13	remaining users	None	remaining users	None

or contains noisy documents without major topics. Hence, we used the *k-means* algorithm with Hellinger distance for clustering this group with number of clusters $k = 9$, centroids:

$$\{23, 82, 113, 67, 22, 50, 16, 47, 2\}$$

and showed the result in the table 6.

Table 6: Result from k-means algorithm using Hellinger distance for cluster 13 (89 users)

Cluster	Users	Major Topic
13.1	7, 12, 20, 21, 23, 42, 72, 81, 93, 96, 131	TP 3 (0.409)
13.2	62, 77, 82, 126	TP 5 (0.339)
13.3	27, 55, 59, 83, 113	TP 8 (0.218)
13.4	67, 111, 122	TP 9 (0.422)
13.5	22, 33, 43, 65, 68, 71, 74, 76, 88, 89, 102	TP 10 (0.330)
13.6	50	TP 12 (0.499)
13.7	16, 35	TP 13 (0.576)
13.8	1, 5, 47, 78, 124, 128	TP 18 (0.453)
13.9	remaining users	None

5.6 Evaluation

In this part of the article, we conducted an evaluation of our algorithm by comparing similarity measure of MCPTM (using the pseudoclosure distance with information from results of minimal closed subsets) and k-means with random choice. The evaluation is performed as follows: we firstly discovered the similarity measure of k-means using three distances: Euclidean distance, Hellinger distance and pseudoclosure distance; we then compared similarity measures among three distances and the similarity measure when we use the number of clusters and the initial centroids from the result of minimal closed subsets. We used the similarity measure proposed by [17] to calculate the similarity between two clusterings of the same dataset produced by two different algorithms, or even the same K-means algorithm. This measure allows us to compare different sets of clusters without reference to external knowledge and is called internal quality measure.

5.6.1 Similarity measure

To identify a suitable tool and algorithm for clustering that produces the best clustering solutions, it becomes neces-

sary to have a method for comparing the different results in the produced clusters. To this matter, we used in this article the method proposed by [17].

To measure the "similarity" of two sets of clusters, we define a simple formula here: Let $C = \{C_1, C_2, \dots, C_m\}$ and $D = \{D_1, D_2, \dots, D_n\}$ be the results of two clustering algorithms on the same data set. Assume C and D are "hard" or exclusive clustering algorithms where clusters produced are pair-wise disjoint, i.e., each pattern from the dataset belongs to exactly one cluster. Then the similarity matrix for C and D is an $m \times n$ matrix $S_{C,D}$.

$$S_{C,D} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & \dots & S_{1n} \\ S_{21} & S_{22} & S_{23} & \dots & S_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ S_{m1} & S_{m2} & S_{m3} & \dots & S_{mn} \end{bmatrix} \quad (12)$$

where $S_{ij} = \frac{p}{q}$, which is Jaccard's Similarity Coefficient with p being the size of the intersection and q being the size of the union of cluster sets C_i and D_j . The similarity of clustering C and clustering D is then defined as

$$Sim(C, D) = \frac{\sum_{1 \leq i \leq m, 1 \leq j \leq n} S_{ij}}{\max(m, n)} \quad (13)$$

5.6.2 Discussion

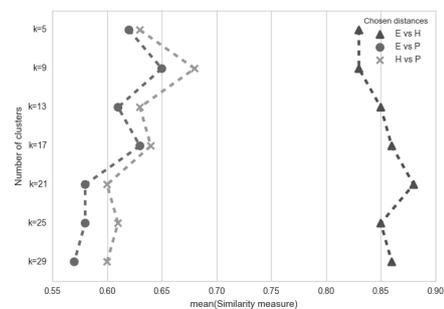


Figure 5: Illustration of the similarity measure where we have the same initial centroids. The abbreviation E stands for Euclidean distance, H for Hellinger distance and P for the pseudoclosure distance.

Table 7: The results of the clustering similarity for K-means with different distance measures. The abbreviation E stands for Euclidean distance, H for Hellinger distance (see definition 8) and P for the pseudoclosure distance (see definition 10 and 11).

k	Same algorithm			Same centroids			Different centroids			Inter-pseudo centroids		
	E	H	P	E vs H	E vs P	H vs P	E vs H	E vs P	H vs P	E vs H	E vs P	H vs P
5	0.423	0.454	0.381	0.838	0.623	0.631	0.434	0.373	0.383	-	-	-
9	0.487	0.544	0.423	0.831	0.665	0.684	0.495	0.383	0.447	-	-	-
13	0.567	0.598	0.405	0.855	0.615	0.633	0.546	0.445	0.469	0.949	0.922	0.946
17	0.645	0.658	0.419	0.861	0.630	0.641	0.641	0.493	0.518	-	-	-
21	0.676	0.707	0.445	0.880	0.581	0.604	0.687	0.478	0.491	-	-	-
25	0.736	0.720	0.452	0.856	0.583	0.613	0.715	0.519	0.540	-	-	-
29	0.723	0.714	0.442	0.864	0.578	0.600	0.684	0.4885	0.511	-	-	-
mean	0.608	0.628	0.423	0.855	0.611	0.629	0.600	0.454	0.480	0.949	0.922	0.946

We have compared the similarity measure between three k-means algorithms with different initializations of the centroids and different numbers of clusters k . We plotted the similarity measure between the clusters computed with the three k -means algorithms with the same initial centroid in Figure 5 and the three k-means algorithms with different initial centroids in Figure 6.

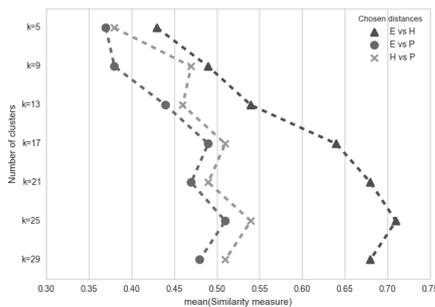


Figure 6: Illustration of the similarity measure where we have different initial centroids. The abbreviation E stands for Euclidean distance, H for Hellinger distance and P for the pseudoclosure distance.

We notice that in the both figures, the Euclidean Distance and the Hellinger distance have higher similarity measure. This is due to the fact that both distances are similar. In Figure 5, we see a big gap between the clusters of Euclidean distance, Hellinger distance and the clusters from Pseudoclosure distance. This gap is closing in Figure 6 and starts opening again from $k = 17$. With a different initial centroids the pseudoclosure distance closed the gap between the k-means algorithms using Euclidean and Hellinger distance. But, when $k > 13$, the number of closed subsets, the gap between the pseudoclosure and the other distances starts opening again. In table 7 where we applied the same algorithm twice, the similarity measure between two clusters results from k-means is low for all three distances: Euclidean, Hellinger, pseudoclosure distance. The different choices of initial centroids can result in very different final partitions.

For k-means, choosing the initial centroids is very important. Our algorithm MCPTM offers a way to compute the centroids based on the analysis of the space of data (in this case text). When we use the centroids com-

puted from the results of minimal closed subsets that we present in Table 5, we have the higher similarity: 0,949 for Euclidean vs Hellinger; 0,922 for Euclidean vs pseudoclosure and 0,946 for Hellinger vs pseudoclosure. It means that the results from k-means using the centroids $\{0, 52, 4, 14, 26, 29, 30, 31, 34, 44, 85, 90, 99\}$ is very similar with all three distances Euclidean, Hellinger, pseudoclosure. We can conclude that the result that we obtained from our MCPTM algorithm is a good result for clustering with this Twitter dataset.

6 Conclusion

The major finding in this article is that the number of clusters and the chosen criterias for grouping the document is closely tied to the accuracy of the clustering results. The method presented here can be considered as a pipeline where we associate Latent Dirichlet Allocation (LDA) and pseudoclosure function. LDA is used to estimate the topic-distribution of each document in corpus and the pseudoclosure function to connect documents with multi-relations built from their major topics or Hellinger distance. With this method both quantitative data and categorical data are used, allowing us to have multi-criteria clustering. We have presented our contribution by applying it on microblogging posts and have obtained good results. In future works, we want to test these results on large scale and more conventional benchmark datasets. And we intend also to parallelize the developed algorithms.

References

- [1] C. C. Aggarwal and C. Zhai. A survey of text classification algorithms. In *Mining text data*, pages 163–222. Springer, 2012.
- [2] M. Ahat, B. Amor S., M. Bui, S. Jhean-Larose, and G. Denhiere. Document Classification with LSA and Pretopology. *Studia Informatica Universalis*, 8(1), 2010.
- [3] Z. Belmandt. *Basics of Pretopology*. Hermann, 2011.

- [4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [5] A. Ferligoj and V. Batagelj. Direct multicriteria clustering algorithms. *Journal of Classification*, 9(1):43–61, 1992.
- [6] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.
- [7] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004.
- [8] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99*, pages 50–57, New York, NY, USA, 1999. ACM.
- [9] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
- [10] T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240, 1997.
- [11] C. Langeron and S. Bonnevey. A pretopological approach for structural analysis. *Information Sciences*, 144(1–4):169 – 185, 2002.
- [12] V. Levorato and M. Bui. Modeling the complex dynamics of distributed communities of the web with pretopology. *Proceedings of the 7th International Conference on Innovative Internet Community Systems*, 2007.
- [13] C. D. Manning and P. Raghavan. *An Introduction to Information Retrieval*, 2009.
- [14] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter's streaming API with twitter's firehose. *arXiv:1306.5204 [physics]*, June 2013. arXiv: 1306.5204.
- [15] M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.
- [16] N. K. Thanh Van Le and M. Lamure. A clustering method associating pretopological concepts and k-means algorithm. *The 12th International Conference on Applied Stochastic Models and Data Analysis*, 2007.
- [17] G. J. Torres, R. B. Basnet, A. H. Sung, S. Mukkamala, and B. M. Ribeiro. A similarity measure for clustering and its applications. *Int J Electr Comput Syst Eng*, 3(3):164–170, 2009.

Appendix

List of Notations:

Notation	Meaning
K	number of topics
V	number of words in the vocabulary
M	number of documents
M	number of documents
\mathcal{D}	corpus
$\phi_{j=1,\dots,K}$	distribution of words in topic j
$\theta_{d=1,\dots,M}$	distribution of topics in document d
$a(A)$	pseudoclosure of A
$F(A)$	closure of A
\mathcal{F}_e	family of elementary closed subset
\mathcal{F}_m	family of minimal closed subset
$\delta(A, B)$	pseudodistance between A, B
$D_A(x)$	interior-pseudodistance of x in A
$MTP(d)$	major topic of document d
$d_H(P, Q)$	Hellinger distance between P, Q
R_{MTP}	relationships based on major topic
R_{dH}	relationships based on Hellinger distance
k	number of clusters