

# Efficient Clustering from Distributions over Topics

Carlos Badenes-Olmedo  
cbadenes@fi.upm.es  
Ontology Engineering Group  
Universidad Politécnica de Madrid  
Boadilla del Monte, Spain

José Luis Redondo-García  
jluisred@amazon.com  
Amazon Research  
Cambridge, UK

Oscar Corcho  
ocorcho@fi.upm.es  
Ontology Engineering Group  
Universidad Politécnica de Madrid  
Boadilla del Monte, Spain

## ABSTRACT

There are many scenarios where we may want to find pairs of textually similar documents in a large corpus (e.g. a researcher doing literature review, or an R&D project manager analyzing project proposals). To programmatically discover those connections can help experts to achieve those goals, but brute-force pairwise comparisons are not computationally adequate when the size of the document corpus is too large. Some algorithms in the literature divide the search space into regions containing potentially similar documents, which are later processed separately from the rest in order to reduce the number of pairs compared. However, this kind of unsupervised methods still incur in high temporal costs. In this paper, we present an approach that relies on the results of a topic modeling algorithm over the documents in a collection, as a means to identify smaller subsets of documents where the similarity function can then be computed. This approach has proved to obtain promising results when identifying similar documents in the domain of scientific publications. We have compared our approach against state of the art clustering techniques and with different configurations for the topic modeling algorithm. Results suggest that our approach outperforms ( $> 0.5$ ) the other analyzed techniques in terms of efficiency.

## CCS CONCEPTS

•Mathematics of computing  $\rightarrow$  Probability and statistics  
; •Information systems  $\rightarrow$  Document topic models; •Applied computing  $\rightarrow$  Document management and text processing;

## KEYWORDS

topic models; semantic similarity; large-scale text analysis; scholarly data

## 1 INTRODUCTION

Given the huge amount of information about any domain that is being produced or captured daily, it becomes crucial to provide mechanisms for automatically identifying the elements that can bring value for the involved agents (general consumers, experts,

companies, investors...) and discard the noisy, non-relevant information. Much of the information is presented in the form of textual documents, making necessary for experts to browse through many of these texts to find relevant data. A way to explore the knowledge inside collection of documents is by moving from one information element to another based on certain criteria that relates them. This approach requires to calculate a similarity matrix with all possible comparisons between elements, so we can later select the most pertinent ones. Since computing a  $n \times n$  matrix takes  $O(n^2)$  time, obtaining all possible pairs of similarities in a large collection of documents can be unfeasible because of the exponential cost of comparing every pair of elements.

Our work is derived from a real need in the domain of digital libraries, where we targeted the task of finding relations among texts based on similar content inside a corpus containing 7,487 digital books and 97,532 chapters (104,960 documents in total). Since the time consumed in calculating the similarity score between two documents was  $t = 7.62 * 10^{-4}$  seconds in a 15x CPU@2.30Ghz and 64GB RAM server, the total time to compute all combinations over the whole corpus went up to around 5 days. Considering that other tasks leveraging the entire collection such as training a Topic Model only required 48 minutes to be executed, calculating the similarity scores between pairs of documents becomes a significant bottleneck when making sense of big collections of documents.

One possible way of finding similarity-based links between pair of documents, is to 1) process the items following different annotation techniques (entities, keywords, etc) that allow machines to programmatically leverage on their content. 2) create a vectorial representation based on those features for each document and 3) compare them following some distance/divergence functions [21].

In order to reduce the execution time, some approaches have introduced mechanisms (mainly clustering algorithms and pre-election methods) to alleviate the problem of making this calculation over the whole set of pairs in the collection. However those methods are still quite costly.

A novel clustering technique based on topic model distributions is proposed in this paper, in order to reduce the required time to find relations between documents in a large corpus of textual documents without compromising efficiency.

We leverage on Probabilistic Topic Models (PTM) [6] as representational models and, in particular, Latent Dirichlet Allocation (LDA) [10] as the way to make this process of finding relations among documents in a corpus more agile and computationally feasible. Probabilistic Topic Modeling techniques [7] are statistical methods that analyze the words of the original texts to discover the themes that run through them. Based on these insights, we can further study how those subjects are connected to each other, and how they change over time. Originally developed as a text-mining tool,

This work is supported by project Datos 4.0 with reference TIN2016-78011-C4-4-R, financed by the Spanish Ministry MINECO and co-financed by FEDER..

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

K-CAP 2017, Austin, TX, USA

© 2017 ACM. 978-1-4503-5553-7/17/12...\$15.00

DOI: 10.1145/3148011.3148019

topic models are now being used to detect instructive structures in data [5] such as computer vision to classify images [24], connect images and captions [8], or build image hierarchies [4] [22]; population genetics [25], and social networks [20]. LDA reduces each document to a vector composed by a fixed set of real numbers, each of which represents a probability distribution of a given topic.

One of the main advantages is that PTM's do not require any prior annotations or labeling of the documents. The topics emerge, as hidden structures, from the analysis of the original texts. The *topics* produced by topic modeling techniques are clusters of similar words. A topic model captures this intuition in a mathematical framework, which allows examining a set of documents and discover, based on the statistics of the words contained in each, what the topics might be and what is the topic balance for each document. Those topics offer a much more intuitive, yet sophisticated way of performing knowledge discovery tasks in big collections of documents.

In contrast to existing unsupervised approaches based on centroids or density measures, our algorithm relies on the outcomes of PTM's to assign each document to a cluster without having to consider the other elements in the corpus. Thus, it only takes  $O(n)$  time to compute all clusters.

In the following section, we provide an overview of the problem to be solved along with existing solutions. After that, a detailed description of our algorithm is given in Section 3. We then (Section 4) experimentally verify the efficiency and effectiveness of our clustering algorithms using real data, and demonstrate that our approach is competitive enough against both a centroid-based and a density-based clustering baselines. Finally, the most relevant results and conclusions are presented together with some future lines work in Section 5.

## 2 BACKGROUND

Traditional retrieval tasks over large collections of textual documents [18] highly rely on individual features like term frequencies (TF-IDF). However, new ways of characterizing documents based on the automatic generation of models surfacing the main subjects covered in the corpus have been developed during recent years. Probabilistic Topic Modeling [6] algorithms are statistical methods that analyze the words of the original texts to discover the themes that run through them, how those themes are connected to each other, or how they change over time.

Probabilistic topic models do not require any prior annotations or labeling of the documents. The topics emerge, as hidden structures, from the analysis of the original texts. These structures are topics distributions, per-resource topic distributions or per-resource per-word topic assignments. In turn, a topic is a distribution over terms that is biased around those words associated to a single theme. This interpretable hidden structure annotates each resource in the collection and these annotations can be used to perform deeper analysis about relationships between resources. In this way, topic modeling provides us an algorithmic solution to organize and annotate large collections of textual documents according to their topics.

The simplest generative topic model is *Latent Dirichlet Allocation* (LDA) [10]. This and other topic models such as *Probabilistic*

*Latent Semantic Analysis* (PLSA) [19] are part of the field known as probabilistic modeling. They are well-known latent variable models for high dimensional data, such as the bag-of-words representation for textual data or any other count-based data representation. While LDA has roots in *Latent Semantic Analysis* (LSA) [14] and PLSA (it was proposed as a generalization of PLSA), it was also influenced by the generative Bayesian framework to avoid some of the over-fitting issues that were observed with PLSA.

This statistical model tries to capture the intuition that documents can exhibit multiple topics. Each document exhibits each topic in different proportion, and each word in each document is drawn from one of the topics, where the selected topic is chosen from the per-document distribution over topics. All the documents in the collection share the same set of topics, but each document exhibits these topics in a different proportion. Documents are represented as a vector of counts with  $W$  components, where  $W$  is the number of words in the vocabulary. Each document in the corpus is modeled as a mixture over  $K$  topics, and each topic  $k$  is a distribution over the vocabulary of  $W$  words. Formally, a *topic* is a multinomial distribution over words of a fixed vocabulary representing some concept. Each topic is drawn from a Dirichlet distribution with parameter  $\beta$ , while each document's mixture is sampled from a Dirichlet distribution with parameter  $\alpha$ . These two priors,  $\alpha$  and  $\beta$ , are also known as hyper-parameters and they are estimated following some heuristic.

A Dirichlet distribution is a continuous multivariate probability distribution parameterized by a vector of positive reals whose elements sum to 1. It is *continuous* because the relative likelihood for a random variable to take on a given value is described by a probability density function, and also it is *multivariate* because it has a list of variables with unknown values. In fact, the Dirichlet distribution is the conjugate prior of the categorical distribution and multinomial distribution.

Unlike a restrictive clustering model, where each document is assigned to one cluster, LDA allows documents to exhibit multiple topics. Moreover, since LDA is unsupervised, the topics covered in a set of documents are discovered from the own corpus; the mixed-membership assumptions lead to sharper estimates of word co-occurrence patterns.

### 2.1 Similarity Measures Across Documents

In a *Topic Model* the feature vector is a topic distribution expressed as vector of probabilities. Taking into account this premise, the similarity between two topic-based resources will be based on the distance between their topic distributions, which can be also seen as two probability mass functions. A commonly used metric is the *Kullback-Liebler* (KL) divergence. However, it presents two major problems: (1) when a topic distribution is zero, KL divergence is not defined and (2) it is not symmetric, which does not fit well with semantic similarity measures that are usually symmetric [27].

*Jensen-Shannon* (JS) divergence [26][23] solves these problems considering the average of the distributions as below [12]:

$$JS(p, q) = \sum_{i=1}^K p_i * \log \frac{2 * p_i}{p_i + q_i} + \sum_{i=1}^K q_i * \log \frac{2 * q_i}{q_i + p_i} \quad (1)$$

where  $K$  is the number of topics and  $p, q$  are the topics distributions. It can be transformed into a similarity measure as follows [13]:

$$\text{sim}_{JS}(D_i, D_j) = 10^{-JS(p, q)} \quad (2)$$

where  $D_i, D_j$  are the documents and  $p, q$  the topic distributions of each of them.

*Hellinger* (He) distance is also symmetric and is used along with JS divergence in various fields where a comparison between two probability distributions is required [9] [17] [11]:

$$\text{He}(p, q) = \frac{1}{\sqrt{2}} \cdot \sqrt{\sum_{i=1}^K (\sqrt{p_i} - \sqrt{q_i})^2} \quad (3)$$

It can be transformed into a similarity measure by subtracting it from 1 [27] such that a zero distance means max. similarity score and vice versa:

$$\text{sim}_{He}(D_i, D_j) = 1 - \text{He}(p, q) \quad (4)$$

### 3 THE APPROACH

Our algorithms draw inspiration from other clustering techniques to divide the initial space of elements into smaller sub-groups where the complexity of calculating all possible distances is significantly reduced. Existing unsupervised approaches based on centroids or density measures require to make comparisons between elements to find groups of similar elements in the collection. They normally follow an iterative methodology to produce the final solution, based on calculating distances between the elements inside each intermediate state. A naïve approach would need to calculate all possible distances between elements, which takes  $O(n^2)$  time for a  $n \times n$  matrix. That makes it impossible to apply such techniques on large collections of documents, since the cost of comparing each element with the others escalates quickly. For those big volumes of data, a clustering task that only takes linear time to discover the clusters can significantly alleviate this problem. For example, a classification method that does not require any other data except the element information to assign the item to the corresponding cluster will take  $O(n)$  time to compose those groups.

The classification method needs to take advantage of both the vectorial representations of the documents and the similarity measure used to relate them in a corpus. Since the representational model considered is based on Probabilistic Topic Models (and more specifically on LDA), the classification method leverages on the particular behavior of Dirichlet distributions, which describes each document by a density vector where the sum of all the probability values must be equal to 1.0. Thus, analyzing the relations between the topics that compose a topic distribution becomes more important than comparing their probability values with another topic distribution.

Our hypothesis is that, given a collection of topic distributions, an unsupervised classification with high precision and linear computing time can be performed by considering only the topic distribution of each document and without needing to further compare it with other document's distributions.

All algorithms have been compared in terms of *cost*, *effectiveness* and *efficiency* [16]. *Cost* is based on the number of pairwise

similarity values. *Effectiveness* handles relevance measures such as *precision* and *recall*. And *efficiency* tries to measure the overall balance between *cost* and *effectiveness*. More details about those measures will be included in Section 4.

#### 3.1 Trends-based Clustering

Topic distributions are formalized as probability distributions following a Dirichlet distribution, so their probability values sum to 1. In this way, the relevance of a topic is influenced and at the same time influences the relevance of the others items in the distribution. Our first approach named *Trends on Dirichlet distribution-based Clustering* (TDC) considers changes in the relevance, i.e. probability values of the topics instead of directly relying on the scores associated to a given topic distribution. It expresses the oscillations between topic weights considering a fixed order between them. The order can be any, as long as it remains constant in all distributions. Thus, a *probability-vector* composed by  $n$  density values is translated to a *trend-expression* made out of  $n - 1$  trend-values such as (1) upward, (2) downward and (0) sustained. This *trend-expression* will identify the cluster the distribution falls into, and therefore the corresponding item belongs to. TDC is defined as:

$$\text{TDC}(P) = T \quad (5)$$

where:  $T_i = 1$ , when  $P_i < P_{i+1}$   
 $T_i = 2$ , when  $P_i > P_{i+1}$   
 $T_i = 0$ , when  $P_i = P_{i+1}$

For example, given the distribution  $P_1 = [0.23, 0.18, 0.33, 0.13, 0.13]$ , the assigned cluster will be  $T = 2120$ . The first value is 2 because 0.23 is greater than 0.18 (same for other values).

#### 3.2 Ranking-based Clustering

We propose a clustering technique named *Ranking on Dirichlet distribution-based Clustering* (RDC) that only considers the top  $n$  topics from the ranked list of probability distributions to classify similar topic distributions. It is based on the focal document selection proposed by [29] to validate LDA-based similarity algorithms against human perception of similarity. RDC is defined as:

$$\text{RDC}(P) = R \quad (6)$$

where  $\forall i \in R, R_i \geq R_{i+1}$  and  $\forall j \in P, R_1 \geq P_j$

This is based on the assumption that the highest weighted topics have a high influence in the rest of topics in terms of calculating distances, when comparing continuous multivariate probability distributions. Since similarity measures (Section 2.1) based on probability distributions are oriented to determine the uncertainty of the distribution, when a mixture of probability distributions is considered, as in the case of Topic Models, the top  $n$  distributions (i.e. the most relevant topics) should be sufficient to allow us grouping similar distributions. Taking into account the above considerations, the RDC algorithm classifies a topic distribution according to only  $n$  highest probability values. For instance, given the following topic distribution:  $P_2 = [0.23, 0.18, 0.33, 0.13, 0.13]$ , the assigned cluster is 3 from RDC-1 because that is the topic with the highest weight.

### 3.3 Cumulative Ranking-based Clustering

A variant of the previous algorithm, named *Cumulative Ranking on Dirichlet distribution-based Clustering* (CRDC), also aims to discover the most representative topics that can help to group similar topic distributions. While RDC is based on a fixed number of topics, CRDC is based on the cumulative sum of the weights of the highest topics. The number of topics is now dynamically determined by a threshold, and once this threshold is reached no more topics are considered. CRDC is defined as:

$$CRDC(P) = C \quad (7)$$

where  $\forall i \in C, C_i \geq C_{i+1}$  and  $\sum_{i=1}^T C_i \geq w$  with  $T$  size of  $C$ , and  $w$  a cumulative weight threshold.

For instance, considering a CRDC algorithm considering a cumulative weight threshold of 0.9, and the following topic distribution:  $P_3 = [0.36, 0.58, 0.05, 0.01]$ . The assigned cluster will be 2–1. To come up with this cluster, a ranked list of topics based on their weights is first calculated,  $R_p = 2|1|3|4$ . Then, a sum of weights according to the order described by  $R_p$  is performed. When the accumulated sum is greater than the threshold, the topics taking part of the sum will be selected to “label” the cluster. In this case, the cumulative weight threshold is 0.9 therefore using only the first two topics we exceed the threshold:  $w = 0.58 + 0.36 = 0.94$

## 4 EXPERIMENTS

In this section we present the experimental setup for evaluating our trends-based (TDC), ranking-based (RDC) and cumulative ranking-based (CRDC) clustering approaches, considering both JS divergence and He distance as similarity measures. We describe the datasets and baseline algorithms that will be used for comparison.

### 4.1 Datasets

We used two datasets to evaluate the performance of the algorithms. The first dataset, DIRICHLET-RANDOM-MIXTURE (DRM), is synthetic [2]. To generate the dataset, we sampled  $k$  probabilistic distributions from a *randomly k-dimensional selector* based on Dirichlet distributions. This implies that all probabilities must sum to 1 for each sampled point. The number of sampled points from this mixture of Dirichlet distributions is  $n = 1000$ .

The second dataset has been created from a collection of research papers published in the *Advances in Engineering Software* (AIES) journal. They were retrieved from the Springer API by using the libAry [1] framework and a Topic Model based on LDA was created from them. The sample is also composed by  $n = 1000$  documents.

Topic models were trained from these datasets by using the criteria described by [28]:  $\alpha = 50/k$ ,  $\beta = 0.01$  and  $k = 2 * (\sqrt{n/2})$ , where  $k$  is the number of topics and  $n$  is the number of documents. Since both datasets contain 1000 documents ( $n$ ), the hyperparameters  $\alpha$  and  $\beta$  are assigned as follow:  $\alpha = 1.136$ , and  $\beta = 0.01$ , and the number of topics is fixed to  $k = 44$ . Further tuning of the settings is not crucial in this evaluation process, because we are not focusing on the quality of the model but on the efficiency when calculating similarities from their representational distributions.

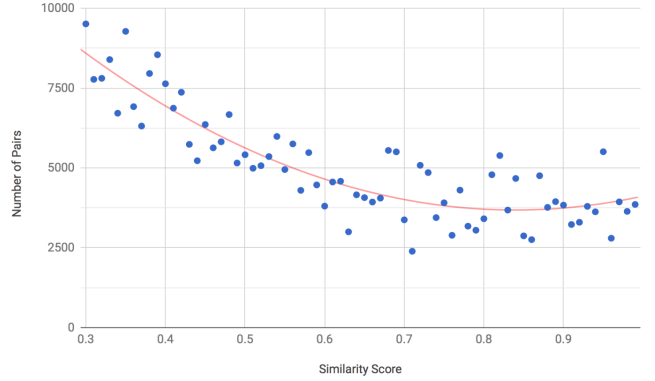


Figure 1: Similarity values grouped by frequency in AIES

### 4.2 Similarity Threshold

Since there is no unified criteria to select a threshold inside the distance scores spectrum that allows us to determine when two documents are similar, we decided to study the distribution of similarity values calculated from all pairwise comparisons. In Figure 1, the result of grouping all similarities by the two most representative decimals, i.e. the first two decimals of the similarity value, is shown. Then, a polynomial function (red line) is approximated to describe the trend of these values. In this function, the similarity score 0.83 emerges as a global minimum and has been used for filtering out the non-similar document pairs.

### 4.3 Baselines

We compare the performance of TDC, RDC and CRDC algorithms against the following baselines:

- *K-Means* as a centroid-based clustering approach.
- *DBSCAN* as a density-based clustering approach.
- *Random*, which randomly selects  $R$  from the dataset

Initially, *K-Means* [3] randomly composes a set of centroids and assigns each point of the sample to its nearest cluster based on a distance measure. Then, a new set of centroids is calculated from the previous ones according to the assigned points. This process is repeated until the set of centroids does not change significantly between consecutive iterations or a maximum number of iterations is reached. The *scalable K-Means* approach used in our experiments is an improved version of *k-means* which obtains an initial set of centers ideally close to the optimum solution. The algorithm implemented at the Apache Commons Math library<sup>1</sup> was used in the experiments. Based on empirical results, the best configuration is:  $k = \text{number} - \text{of} - \text{topics} = 44$  and  $\text{maxIterations} = 50$

A widely known density-based algorithm is *DBSCAN* [15], which compose clusters from the neighborhood of each point considering at least a minimum number of points and a given radius. Thus, it requires to specify the radius of the point’s neighborhood, *Eps*, and the minimum number of points in the neighborhood *MinPts*. Based on empirical results, the best results were obtained with the following configuration:  $\text{eps} = 0.1$  and  $\text{minPts} = 50$

<sup>1</sup> <http://commons.apache.org/proper/commons-math/>

The *Random* algorithm takes as input a parameter  $m$  and randomly divides the dataset into  $m$  equal-sized groups of similar documents. For the evaluation,  $m$  was set to the number of topics, the dimension of the dataset.

With respect to the proposed algorithms and taking into account empirical results, the RDC algorithm is set to use the *top1* highest topics, and the cumulative weight threshold for the CRDC algorithm is set to 0.9.

#### 4.4 Measure

A gold-standard is created for each dataset and distance metric considered. They are created by calculating all pairwise similarities from their documents. Since the  $n \times n$  similarity matrix requires  $O(n^2)$  time to be calculated, the selected size of datasets has not been too large  $n = 1000$ .

We considered three measures to evaluate our algorithms with respect to the baseline:

- **cost**: based on the number of similarity score calculations required by the algorithm:

$$\text{cost} = (\text{reqSim} - \text{minSim}) / (\text{totalSim} - \text{minSim}) \quad (8)$$

The *minSim* corresponds to the number of similar documents obtained from using the *threshold* score previously mentioned in section 4.2. The *totalSim* corresponds to the Cartesian product of existing documents:  $\text{totalSim} = n * n = 1,000,000$ . And the *reqSim* corresponds to the number of similarities calculated by the algorithm.

- **effectiveness**: based on *precision* and *recall*. It expresses the quality of the algorithm:

$$\text{effectiveness} = \frac{\text{precision}^2 + \text{recall}^2}{2} \quad (9)$$

- **efficiency**: based on the previous ones, it express a compromise between quality and performance:

$$\text{efficiency} = \text{effectiveness} - \text{cost} \quad (10)$$

#### 4.5 Results

The code used to evaluate the algorithms along with the results obtained are available on GitHub [2].

**In terms of effectiveness** (Figures 2 and 3), the results highlight that *K-Means* and *CRDC* outperform the other algorithms. *K-Means* was expected to be a top performer because the algorithm itself performs comparisons to map clusters. The fact that *CRDC* has such good performance encourages us to think that, in fact, the most relevant topics when they altogether exceed a certain high weight threshold, are those that best represent the document and allow to group together similar documents. However, as shown in tables 1, 2, 4 and 3, considering a fixed number of more relevant topics (*RDC*) or considering the trend of their weights (*TDC*) does not seem to perform so well on aggregating similar documents, since their *precision* and *recall* values are very low in both cases. It is surprising that the *DBSCAN* has such low value. Taking a look at its *precision* and *recall* values, and also seeing the number of groups that each algorithm has created (Figure 4), we believe that having a corpus containing a very cohesive set of documents (all papers in corpus belong to the same journal) affects the performance of this algorithm since it divides the corpus into a lower number of

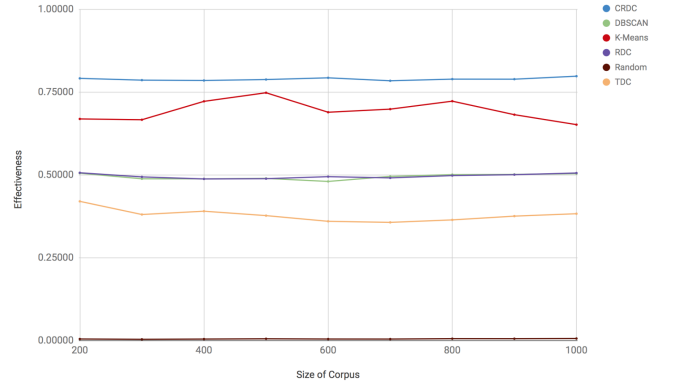


Figure 2: Effectiveness (JS-based) in AIES

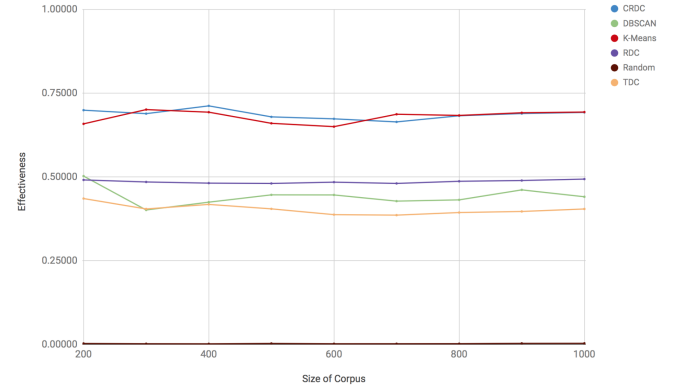


Figure 3: Effectiveness (He-based) in AIES

groups. This way, it obtains high values of *recall* because most of the pair-wise distances are computed, but very low *precision*.

The results also show that the behavior of the algorithms does not differ significantly when using different similarity measures, for example JS divergence (Figure 2) and He distance (Figure 3). This highlights the importance of the documents' topic distributions to successfully classify them into smaller groups of similar items, while other particular aspects such as the distance or similarity metric used to compare them are less influential.

Size	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.94	0.10	<b>0.96</b>	0.31	0.42	0.12
300	0.93	0.15	<b>0.94</b>	0.30	0.39	0.08
400	<b>0.93</b>	0.15	0.89	0.29	0.39	0.09
500	<b>0.92</b>	0.30	0.90	0.28	0.38	0.09
600	<b>0.92</b>	0.19	0.88	0.28	0.38	0.08
700	<b>0.92</b>	0.20	0.91	0.28	0.38	0.09
800	<b>0.92</b>	0.12	0.89	0.30	0.39	0.10
900	<b>0.92</b>	0.13	0.87	0.30	0.40	0.10
1000	<b>0.93</b>	0.13	0.90	0.30	0.40	0.10

Table 1: Precision (JS-based) in AIES

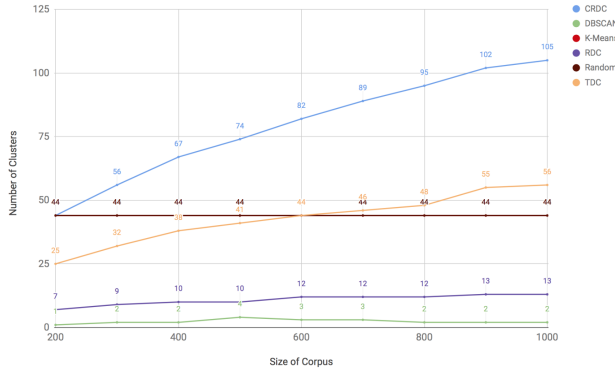


Figure 4: Clusters in AIES

Size	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.75	0.07	<b>0.84</b>	0.23	0.08	0.33
300	0.74	0.08	<b>0.83</b>	0.23	0.06	0.32
400	<b>0.76</b>	<b>0.09</b>	0.76	0.22	0.06	0.32
500	0.73	0.08	<b>0.74</b>	0.21	0.08	0.31
600	0.72	0.08	<b>0.73</b>	0.21	0.06	0.30
700	0.71	0.10	<b>0.76</b>	0.21	0.06	0.30
800	0.73	0.11	<b>0.78</b>	0.22	0.07	0.31
900	0.73	0.12	<b>0.80</b>	0.22	0.08	0.32
1000	0.74	0.15	<b>0.77</b>	0.23	0.08	0.32

Table 2: Precision (He-based) in AIES

Size	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.92	<b>1.00</b>	0.79	0.96	0.02	0.87
300	0.91	0.89	0.84	<b>0.96</b>	0.02	0.84
400	0.92	0.92	0.90	<b>0.96</b>	0.02	0.86
500	0.91	0.94	0.88	<b>0.96</b>	0.03	0.85
600	0.91	0.94	0.87	<b>0.96</b>	0.02	0.83
700	0.91	0.92	0.90	<b>0.96</b>	0.02	0.83
800	0.92	0.92	0.88	<b>0.96</b>	0.02	0.83
900	0.92	0.95	0.86	<b>0.96</b>	0.02	0.83
1000	0.92	0.93	0.89	<b>0.97</b>	0.02	0.84

Table 3: Recall (JS-based) in AIES

Size <sup>c</sup>	CRDC	DBSCAN	K-Means	RDC	TDC	Random
200	0.84	<b>1.00</b>	0.65	0.96	0.02	0.82
300	0.84	<b>0.98</b>	0.76	0.95	0.02	0.78
400	0.84	<b>0.98</b>	0.79	0.94	0.02	0.79
500	0.85	0.94	0.87	<b>0.95</b>	0.02	0.78
600	0.86	<b>0.96</b>	0.80	0.95	0.02	0.76
700	0.85	<b>0.98</b>	0.80	0.95	0.02	0.76
800	0.85	<b>0.99</b>	0.81	0.95	0.02	0.76
900	0.85	<b>0.99</b>	0.75	0.95	0.02	0.77
1000	0.86	<b>1.00</b>	0.74	0.96	0.02	0.78

Table 4: Recall (He-based) in AIES

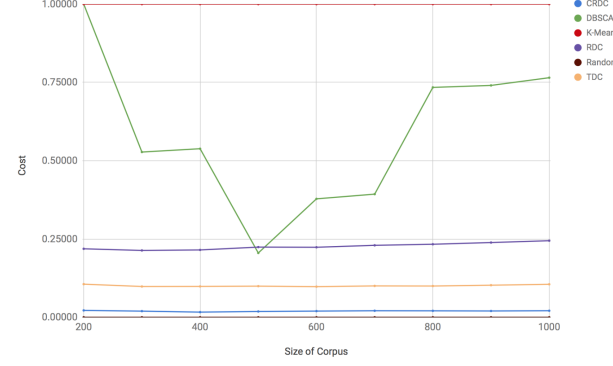


Figure 5: Cost (JS-based) in AIES

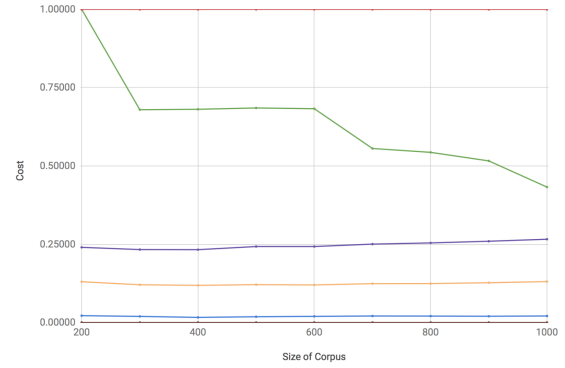


Figure 6: Cost (He-based) in AIES

In terms of *cost* (Figures 5 and 6), the best clustering algorithm, as expected, is based on *random* selection. This is due to the fact that the number of pairs compared by this algorithm is always the minimum, given the dataset is simply randomly divided into  $m$  equal-sized groups, where  $m$  is equals to the number of topics, i.e. dimension of the dataset. Since *K-Means* and *DBSCAN* make comparisons between documents until their internal condition is satisfied, they are the most inefficient approaches. *K-Means* involves the highest cost because it compares all the documents with the 44 centroids in each iteration.

Among our proposals, the main reason for an algorithm to present a higher cost is due to the number of groups the corpus is divided into (see Figure 4). The greater the number of groups, the fewer the number of later comparisons that have to be made and, therefore, the lower the cost of the algorithm.

The behavior of the *DBSCAN* algorithm depends remarkably on the similarity metric used. We think that this may be due to the way in which both measures satisfy the triangle inequality condition, since one is based on divergence (JS) and the other on distance (He). This property, which defines  $distance(a, b) \leq distance(a, c) + distance(c, b)$ , is very important in the calculations that *DBSCAN* makes to discover the groups, since it only calculates the distances between near points.

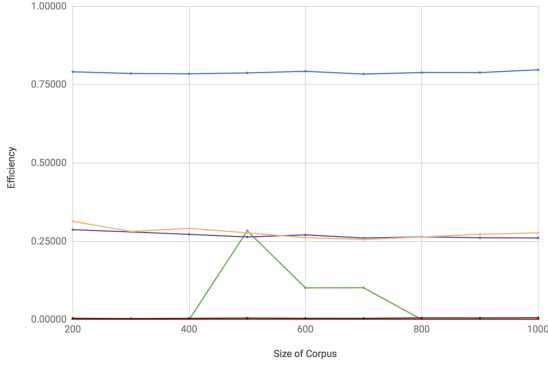


Figure 7: Efficiency (JS-based) in AIES

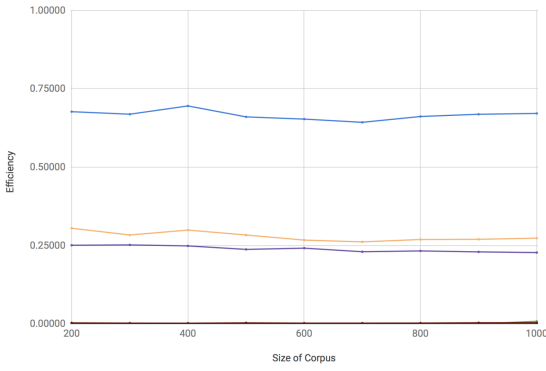


Figure 8: Efficiency (He-based) in AIES

Finally, **in terms of efficiency** (Figures 7, 8), regardless of the similarity measure used, the algorithm that yields the best performance according to the results obtained is *CRDC*. Overall, *CRDC* demonstrates a high accuracy classification and a lower cost by improving the performance offered by centroid-based or density-based approaches.

We have also created a synthetic dataset, DRM (Section 4.1), composed of 1000 Dirichlet distributions with the same dimensions than topics in AIES:  $k = 44$ . Unlike AIES, topic distributions have been randomly generated which imply that the similarity values are not so high:  $\min = 0.06$ ,  $\text{mean} = 0.18$  and  $\max = 0.61$ . Following the same criteria than before (Section 4.2), the similarity threshold is now fixed to 0.34 (Figure 9). Results **in terms of effectiveness** (Figure 10) show a poor performance of the RDC and CRDC algorithms. The reason is that both are based on the fact that the highest weighted topics are shared between similar distributions. However, this condition is not satisfied when the similarity value between them is low.

To confirm this behavior, we created a third dataset (DRM2) with the same size but with only 4 dimensions (4 topics). The goal is to achieve more similar distributions than in DRM even though they are also randomly generated. Since the similarity values range from  $\min = 0.04$ ,  $\text{mean} = 0.34$  to  $\max = 0.99$ , the similarity threshold is now fixed to 0.66 (more details in section 4.2). The results (Figure 11)

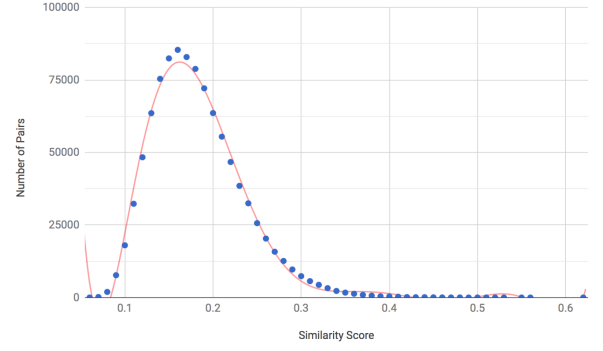


Figure 9: Similarity values grouped by frequency in DRM

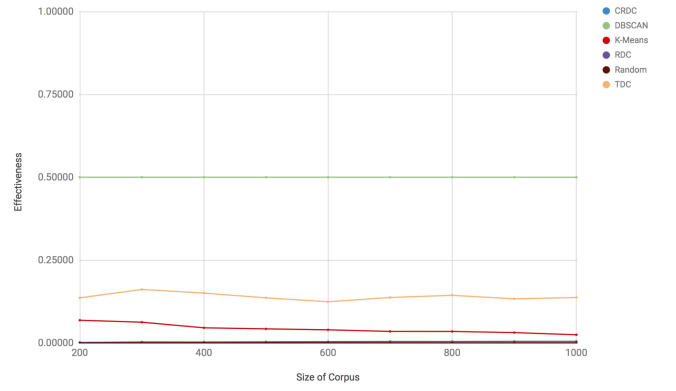


Figure 10: Effectiveness (JS based) in DRM

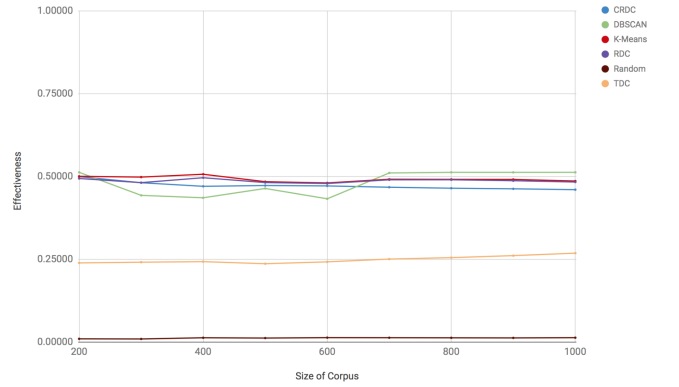


Figure 11: Effectiveness (JS based) in DRM2

show an improvement in the accuracy of both the RDC and CRDC algorithms. Although scores are still not as high as for the AIES dataset, the increase compared to the DRM dataset shows that their *precision* and *recall* improve when the similarity threshold is higher. On the other hand, both the DBSCAN and TDC algorithms show similar behavior in both datasets, which means that their performance is not affected by the similarity threshold.

## 5 CONCLUSIONS AND FUTURE WORK

Processing a continuously growing collection of human generated documents requires techniques that divide the space into smaller regions containing potentially similar documents. Some algorithms in the literature tackle this problem from an unsupervised point of view, but they incur in high temporal costs and may not be suited for the domain being studied.

Three novel unsupervised clustering algorithms, *TDC*, *RDC* and *CRDC*, are described in this paper relying on the distributions inferred from a topic modeling algorithm (LDA). They are presented as a means to identify a smaller set of documents where only the similarity function has to be computed. They leverage on the particular behavior of Dirichlet distributions describing topic distributions, where the highest weighted topics have a high influence on the rest of topics. This also means that given a topic distribution, the relations between their topic weights such as order or trends between them, are more important than the density values.

Although we initially thought that using only a fixed number of topics with higher weights of a topic distribution (*RDC*), or taking into account only the trend changes between the weights of consecutive topics (*TDC*), could be enough to classify similar topic distributions, the results obtained have shown that these properties are not sufficient. Results in terms of *efficiency*, *effectiveness* and *cost* have been shown comparing the proposed algorithms with existing centroid-based and density-based clustering techniques. They reveal that obtaining the most representative topics of a topic distribution by comparing the sum of their weights with respect to the rest (*CRDC*) is a promising approach, which improves the *efficiency* obtained by other centroid-based and density-based approaches. While *K-Means* takes  $O(n^k * \log n)$  and *DBSCAN* takes  $O(n * \log n)$  time to classify  $n$  documents in a collection, the proposed algorithms only take linear time ( $O(n)$ ) because they do not require any other data except their own topic distribution to assign it to a cluster.

A hierarchical approach for *RDC* algorithm was also considered but it did not produce good results. Hybrid methods combining some of these novel approaches with existing techniques will be performed in future work on the same line.

## REFERENCES

- [1] Carlos Badenes-Olmedo, Jose Luis Redondo-Garcia, and Oscar Corcho. 2017. Distributing Text Mining tasks with librAiry. In *Proceedings of the 17th ACM Symposium on Document Engineering (DocEng)*. <https://doi.org/10.1145/3103010.3121040>
- [2] Carlos Badenes-Olmedo, Jose Luis Redondo-Garcia, and Oscar Corcho. 2017. librAiry/eval-similarity-calculation. (2017). <https://doi.org/10.5281/zenodo.931305>
- [3] Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. 2012. Scalable K-Means++. *Proceedings of the VLDB Endowment (PVLDB)* 5 (2012), 622–633. <https://doi.org/10.14778/2180912.2180915>
- [4] Evgeniy Bart, Max Welling, and Pietro Perona. 2011. Unsupervised organization of image collections: Taxonomies and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 11 (2011), 2302–2315. <https://doi.org/10.1109/TPAMI.2011.79>
- [5] David Blei. 2012. Introduction to Probabilistic Topic Modeling. *Commun. ACM* 55 (2012), 77–84. <https://doi.org/10.1145/2133806.2133826>
- [6] David Blei, Lawrence Carin, and David Dunson. 2010. Probabilistic topic models. *IEEE Signal Processing Magazine* 27, 6 (2010), 55–65. <https://doi.org/10.1109/MSP.2010.938079>
- [7] David Blei, Lawrence Carin, and David Dunson. 2010. Probabilistic Topic Models: A focus on graphical model design and applications to document and image analysis. *IEEE signal processing magazine* 27, 6 (2010), 55–65. <https://doi.org/10.1109/MSP.2010.938079>
- [8] David M Blei and Michael I Jordan. 2003. Modeling Annotated Data University of California Computer Science Division and Department of Statistics University of California Modeling Annotated Data. *Science* (2003).
- [9] David M. Blei and John D. Lafferty. 2007. A correlated topic model of Science. *The Annals of Applied Statistics* 1, 1 (2007), 17–35. <https://doi.org/10.1214/07-AOAS136>
- [10] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3, 4-5 (2003), 993–1022. <https://doi.org/10.1162/jmlr.2003.3.4-5.993>
- [11] Jordan Boyd-Graber and Philip Resnik. 2010. Holistic Sentiment Analysis Across Languages: Multilingual Supervised Latent Dirichlet Allocation. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* October (2010), 45–55. <https://doi.org/10.1017/CBO9781107415324.004>
- [12] a Celikyilmaz, D Hakkani-Tur, and Gokhan Tur. 2010. LDA Based Similarity Modeling for Question Answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*. 1–9.
- [13] Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-Based Models of Word Cooccurrence Probabilities. *Machine Learning* 34, 1-3 (1999), 43–69. <https://doi.org/10.1023/A:1007537716579>
- [14] S Deerwester, S T Dumais, G W Furnas, T K Landauer, and R Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41 (1990), 391fi?407. [https://doi.org/10.1002/\(SICI\)1097-4571\(199009\)41:6<391::AID-ASLI>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASLI>3.0.CO;2-9)
- [15] Martin Ester, Hans-peter Kriegel, Jrg S, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. (1996), 226–231. <https://doi.org/citeulike-article-id:3509601>
- [16] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. 2001. On clustering validation techniques. *Journal of Intelligent Information Systems* 17, 2-3 (2001), 107–145. <https://doi.org/10.1023/A:1012801612483> arXiv:astro-ph/0005074v1
- [17] David Hall, Dan Jurafsky, and Christopher D Manning. 2008. Studying the History of Ideas Using Topic Models. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, October 2008* October (2008), 363–371. <https://doi.org/10.3115/1613715.1613763>
- [18] Marti A Hearst and South Hall. 1999. Untangling Text Data Mining. In *The 37th Annual Meeting of the Association for Computational Linguistics*. 1–13.
- [19] Thomas Hofmann. 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning* 42, 1-2 (2001), 177–196.
- [20] Shuhui Jiang, Xueming Qian, Jialie Shen, Yun Fu, and Tao Mei. 2015. Author topic model-based collaborative filtering for personalized POI recommendations. *IEEE Transactions on Multimedia* 17, 6 (2015), 907–918. <https://doi.org/10.1109/TMM.2015.2417506>
- [21] Tom Kenter and Maarten de Rijke. 2015. Short Text Similarity with Word Embeddings Categories and Subject Descriptors. *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM 2015)* (2015), 1411–1420. <https://doi.org/10.1145/2806416.2806475>
- [22] Li-jia Li, Chong Wang, Yongwhan Lim, David M Blei, and Li Fei-fei. 2010. Building and Using a Semantivisual Image Hierarchy. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 3336–3343.
- [23] Jianhua Lin. 1991. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory* 37, 1 (1991), 145–151. <https://doi.org/10.1109/18.61115>
- [24] Wenhan Luo, Bjrn Stenger, Xiaowei Zhao, and Tae-Kyun Kim. 2015. Automatic Topic Discovery for Multi-Object Tracking. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. 3820–3826.
- [25] Jonathan K. Pritchard, Matthew Stephens, and Peter Donnelly. 2000. Inference of population structure using multilocus genotype data. *Genetics* 155, 2 (2000), 945–959.
- [26] C Radhakrishna Rao. 1982. Diversity: Its Measurement, Decomposition, Apporportionment and Analysis. *Sankhyā: The Indian Journal of Statistics, Series A* 44, 1 (1982), 1–22.
- [27] Vasile Rus, Nobal Niraula, and Rajendra Banjade. 2013. Similarity Measures Based on Latent Dirichlet Allocation. In *Computational Linguistics and Intelligent Text Processing*. 459–470.
- [28] M Steyvers, Grif Ths, and T. 2006. Probabilistic topic models. In *Landauer, T., McNamara, D., Dennis, S., and Kintsch, W., editors, Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum. Tang, Z. and MacLennan, J (2006).
- [29] W Ben Towne, Carolyn P Rosé, and James Herbsleb. 2016. Measuring Similarity Similarly: LDA and Human Perception. *ACM Transactions on Intelligent Systems and Technology ACM Reference Format ACM Trans. Intell. Syst. Technol* 7, 2 (2016), 1–25.