Facilitating Understanding of Large Document Collections

Jae Hyeon Bae
Texas Advanced Computing Center
The University of Texas at Austin
Austin, Texas
jaybae@tacc.utexas.edu

Weijia Xu
Texas Advanced Computing Center
The University of Texas at Austin
Austin, Texas
xwj@tacc.utexas.edu

Maria Esteva
Texas Advanced Computing Center
The University of Texas at Austin
Austin, Texas
maria@tacc.utexas.edu

Abstract—**Large document collections containing multiple topics can be overwhelming to understand, requiring librarians and archivists significant time and efforts to develop access points. Efficient computational methods can aid this process by uncovering groups of documents that can be described for access. We investigate the use of density based clustering with document segmentation to identify points of access as dense clusters of information. The method returns stories and classes of cohesive clusters that can be described as precise points of access. We found that our method performs more efficiently than K-means clustering and topic model using Latent Dirichlet Allocation (LDA). We use Hadoop to process a large document collection.**

Keywords: density based clustering, information retrieval, distributed processing, Hadoop/MapReduce, digital archives

I. Introduction

As more document collections are born digital or digitized and their size grows exponentially, understanding their contents to provide precise access is a challenging problem for digital archivists. To address this problem, we investigate a computational method to automatically identify stories and classes of information that can be used to derive collections descriptions. The method maps *finding aids* [1], in which archivists describe the relevant contents of a collection in parts and in whole, allowing users to navigate and understand the collection easily.

A document collection refers to a set of documents that belong to the same provenance. This entails that the documents within have intrinsic relationships. In turn, these relationships are more pronounced between some documents that relate to a same target activity, or belong to a same function. In contrast to keyword-based indexing and retrieval models, finding aids describe groups of thematically tight related documents. These descriptions are used as access points to help users find information within smaller sets of documents. Traditionally, building a finding aid is a manual process that requires reading the documents and making inferences about their relationships to generate descriptions [2].

To facilitate the process of understanding a large collection of documents in order to produce a finding aid, thematically related groups of documents can be identified automatically. In this project we address the challenge of generating clusters containing documents that cohesively reflect activities, projects, and transactions recorded in large collections. We developed a density based clustering method with document segmentation that receives large amounts of documents as input, and narrows the data to clusters containing cohesive stories and classes of information. Once these clusters are identified, describing their contents as access points is feasible.

Density based clustering is a method to identify tight clusters out of a set of data points. As opposed to just assigning each data point to a cluster, this algorithm assigns relevant data points to a corresponding cluster. Documents that are not associated with any cluster are treated as noise points. In turn, noise points are considered less relevant to obtaining precise notions about the collection. Another advantage of density based clustering is that a-priori knowledge of the number of cluster seeds is not required.

Documents in a collection may have irregular sizes and diversity of contents. To account for these variations, we introduce a document segmentation step to form the clusters [3]. Document segmentation finds similarities between documents that may refer to more than one topic or activity (and therefore may belong to more than one cluster), and identifies common themes between documents that differ in size. During pre-processing, documents are first divided into segments with small length variations, and the similarity scores between segments are used for clustering. For scalability purposes we utilize Hadoop for distributed computing [4].

We tested our method with a large email collection that presents most of the challenges involved in document collections such as: diversity in document sizes and topics, repetitive themes, and duplicate documents. We compared the results obtained between segmented and non-segmented density based clusters and between segmented density based clustering with K-means clustering and topic extraction using Latent Dirichlet Allocation (LDA).

Our main contribution is the development and implementation of a scalable density-based clustering method that uses paragraph segmentation to generate clusters. These clusters contain classes of information as well as stories about projects and transactions. Described as access points, the clusters provide an overall understanding of the collection. Following we detail our implementation and discuss the results obtained.

II. Related Work

Our work is related to topic detection and tracking in information retrieval, as well as to large-scale density based

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.268

clustering algorithm applications.

A wide range of research has been devoted to discover topics in document collections using clustering algorithms [5]. A hierarchal clustering method was applied to document categorization for collection browsing and navigation [6]. Latent Dirichlet Allocation (LDA) is a generative probability model used to extract topics over text corpus in an unsupervised way [7]. LDA can cluster semantically related words into topics, and documents into a random mixture of latent topics. [8] uses LDA to identify the number of topics in the Enron email collection. Although LDA based approaches can identify topics in a large corpus through dimensionality reduction, the results consist of a set of disjoint words that cannot be easily traced back to a set of documents. Thus the results are hard to use as access points for our problem.

Topic Detection and Tracking (TDT) [9] has been used to analyze continuous text streams. A common process includes segmenting text streams to detect and cluster topically homogeneous constituent stories using methods such as Hidden Markov Model, decision tree, or K-means. Different from the TDT methods described, our method uses density based clustering [10] to identify relevant subsets of the collection from which subsequent smaller parts can be found and used as points of access.

Su et al. applies a recursive density based clustering method to cluster large sets of web documents based on distance measures calculated from web log data [11]. The method is implemented for serial processing. [12] proposes using the scalable density based clustering method called Hierarchical Density Shaving to detect high density regions from a tera-scale astronomical data set. The HDS algorithm can detect clusters of different densities in a hierarchical way and disregard noisy background data. [13] implements a Friends-of-Friends (FoF) clustering approach using DryadLINQ for astronomy data. Although FoF is a special case of density-based clustering, the model assumes no noise points and therefore is not suitable for our problem. Our method is based on a general case of density-based clustering algorithm where noise points are present. It is implemented using Hadoop, an open-source distributed programming framework [4] for both pairwise document similarity calculation [15], and clustering algorithms [16].

III. Methods

Our data processing workflow consists of the following steps:

a. Document pre-processing and segmentation.
b. Computation of pairwise similarity matrices between document segments
c. Identification of dense document clusters

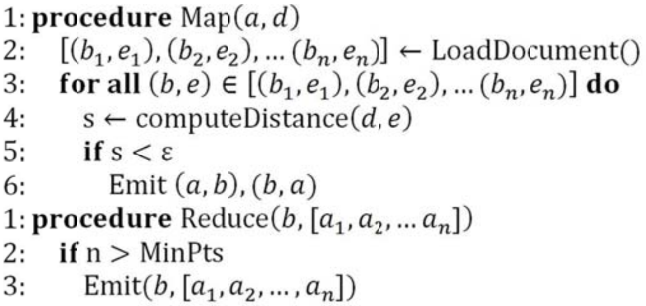
A. *Document Preprocessing and Segmentation*

We divide long documents into several segments based on a Minimum Number of Character Threshold (MNCT). The MNCT value is determined by the distribution of the number of characters in each paragraph of the document collection. We then convert each segment into a TFIDF vector in the vector space model [17] after stop words removal. We use the Mahout [18] library in Hadoop for stop words removal and vector conversion.

B. *Density Based Clustering in Hadoop*

A dense cluster is usually defined by two parameters: a) a distance value that defines a neighborhood surrounding each point called Eps, and b) a minimum number of points within each neighborhood called MinPts. If a point has more than MinPts points within the neighborhood defined by Eps, the point is labeled as a core point. If the neighborhood of a point contains less than MinPts, but contains at least one core point, that point is labeled as a boundary point. Otherwise, the data point is treated as noise. A basic density based clustering algorithm is used to define all the core points with an overlapping neighborhood as a cluster.

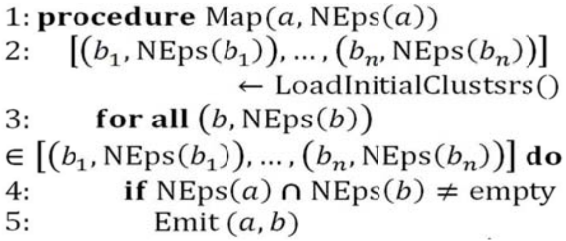
To effectively address large-scale document collections, our method is implemented in Hadoop for distributed processing, and based on the MapReduce model [14]. In the map stage, all pairs of document vectors are divided and distributed to the available computing nodes. Each node computes pairwise similarities independently, and identifies the Eps-neighborhood of all the points containing more than MinPts of points. In the Reduce stage, the results of the distributed density-connected clusters are combined to form the final clustering result.



```
procedure Map$(a, d)$
    $[(b_1, e_1), (b_2, e_2), \dots (b_n, e_n)] \leftarrow$ LoadDocument()
    for all $(b, e) \in [(b_1, e_1), (b_2, e_2), \dots (b_n, e_n)]$ do
        $s \leftarrow$ computeDistance$(d, e)$
        if $s < \varepsilon$
            Emit $(a, b), (b, a)$
procedure Reduce$(b, [a_1, a_2, \dots a_n])$
    if $n >$ MinPts
        Emit$(b, [a_1, a_2, \dots, a_n])$
```

Figure 1 Pseudo code for the Map and Reduce module in Hadoop

Figure 1 shows the pseudo code of the Map and Reduce processing implemented in Hadoop. In the Map pseudo code, a is a document id, and d is a document representation as a term TFIDF vector. With the same notation, at $(b_i, e_i) b_i$ is a document id and e_i is a term vector. In the Reduce pseudo code, $[a_1, a_2, \dots, a_n]$ is Eps-neighborhood of document b and if the size of NEps(a) is greater than MinPts, this is an initial cluster.



```
procedure Map$(a, \mathrm{NEps}(a))$
    $[(b_1, \mathrm{NEps}(b_1)), \dots, (b_n, \mathrm{NEps}(b_n))]$
            $\leftarrow$ LoadInitialClustsrs()
        for all $(b, \mathrm{NEps}(b))$
$\in [(b_1, \mathrm{NEps}(b_1)), \dots, (b_n, \mathrm{NEps}(b_n))]$ do
            if $\mathrm{NEps}(a) \cap \mathrm{NEps}(b) \neq$ empty
                Emit $(a, b)$
```

Figure 2 Pseudo code for combined cluster

Two clusters share common documents are merged into an augmented connected set. Figure 2 shows the pseudo code to check whether two clusters have documents in common. To save memory usage, we encode each cluster with a BitSet. This process is repeated until convergence.

C. *Parameter Selection*

Density based clustering is sensitive to its two parameters, Eps and MinPts. In [10], an interactive approach is used to determine those parameters based on the distribution of the distance between points and their k-th nearest neighbor. Similar to this approach, we first compute the distance distribution of all the points to their k-th neighbors. We then plot the distance distribution in a histogram to determine a suitable Eps value.

D. *Post Processing*

After completing the density based clustering we obtain a list of clusters, containing paragraph IDs. We then use that information to recover the list of documents in the cluster.

IV. Method Implementation and Results Analysis

A. *Data Set*

To test this method we used the Enron email database built in [19]. The dataset contains 255,636 email messages with email headers removed. It's size is 219MB with gzip compression.

B. *Document Segmentation and Parameter Selection*

To determine the adequate size of the segments we looked at the documents length distribution and found that, 38% of documents are shorter than 500 characters, and 49% of documents are shorter than 750 characters. As a result, we used 750 as the MNCT value, which generated a total 719,786 paragraphs. After removing exact duplicate paragraphs we ended with 61,7937 paragraphs.

Given that there is no good method to decide the best values of MinPts and Eps simultaneously, we determine the parameter value based on the number of clusters to obtain. Once we choose the MinPts parameter, we find the appropriate Eps value for that predetermined MinPts value. With small MinPts values of 10 or 20, we can obtain a very large number of clusters. For example, for MinPts of 10, Eps is determined as 0.45 by our heuristic, a combination that generated more than 3000 clusters.

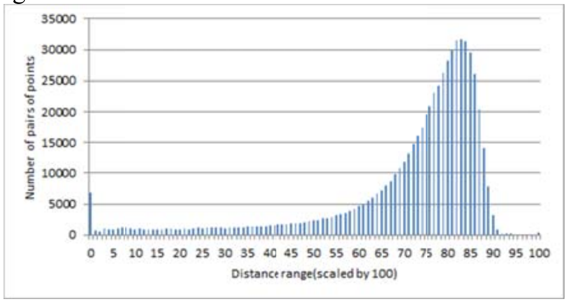


Figure 3. Distribution of distance distribution to 50th neighbor

To obtain a manageable amount of clusters and facilitate their evaluation we chose a MinPts of 50. The Eps value is estimated as 0.60 using the k-distance graph method [10]. Fig. 3 shows the distance distribution of each point to its 50th neighbor. The X-axis is the distance value range scaled by 100. The Y-axis is the number of pairs of points that fall in each 0.01 distance range.

C. *Clustering Statistics*

Using MinPts=50 and Eps=0.60 as clustering parameters, 33% (203,946 out of 617,937) paragraphs and 54% (140,148 out of 255,504) emails were clustered, and the remaining paragraphs and emails were regarded as a noise. Figure 4 shows the size of each cluster. In this run we obtained 31 moderate size clusters and one large cluster. The largest cluster has 197,708 paragraphs and 137,120 emails. The rest of the clusters contain about 1% of the total number of emails.

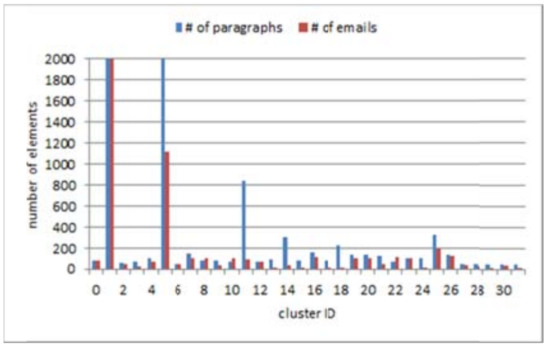


Figure 4 Size of the initial clustering results

D. *Clustering Analysis*

With the help of an archivist we conducted a qualitative evaluation to verify the contents of the 31 moderate-sized clusters. A qualitative evaluation is necessary to identify if they contain cohesive information and to assess their relevance as access points. Upon reading them we classified the clusters as: a) stories, b) similar, c) category and d) anomaly. "Stories" are clusters whose emails contain different aspects of a target activity. For example, cluster 15 contains emails of different sizes, sent and received by different people about gas transactions in two parishes in Louisiana. Albeit repetitions reflected in the forwarded sections of the messages, the majority of the emails contain new details about the companies involved and about their transactions. "Similar" clusters contain messages sent and received by different people in which the text contains minimal variations. Among these we found a cluster with the notices of stock transactions sent to clients, and another with the chain of emails sent to Ken Lay by different angry citizens. The bulk of the clusters fall in the "category" type. The latter includes clusters with sports, weather, travel, and business news, as well as Enron's information summaries sent to the executives.

Cluster types: similar, stories and category are thematically tight and do not contain any noise. We consider these as good clusters that can aid in identifying relevant contents of a collection. We also obtained an anomaly cluster consisting of one large email, and a large and non-cohesive cluster 1. Table 1 below shows the types of clusters inferred from this evaluation. Types will be different for different test collections containing types of documents and different contents.

Table 1 Qualitative cluster evaluation

Clusters	Typology
15, 30, 16	Stories
5, 27, 3, 12, 0, 25	Similar
1	Large
31, 24, 13, 18, 29, 21, 14, 9, 6, 2, 4, 11, 10, 7, 8, 23, 20, 22, 19, 26, 28	Category
17	Anomaly

E. Comparision with Non-segmented Results

For a quantitative evaluation we compared the results between non-segmented and segmented emails using the same MinPts and Eps parameters. Without segmentation, we obtained 48 moderate sized clusters and one large cluster. 30% (77,657 out of 255,504) of the emails are clustered in the moderate sized clusters containing 3% (7,691 out of 255,504) of the emails. The largest cluster contains 69,966 emails. Compared to the results of the segmented emails, syntactically similar and semantically different emails are clustered. The averages of intra-cluster distance are 0.595 and 0.617 for segmented and non-segmented respectively. The results indicate decreased cluster purity when documents are clustered without segmentation.

After a qualitative evaluation of the results we made two observations that support the effectiveness of the segmented results. First, in the segmented result 94 emails from INO.com are separated in two clusters whose messages differ from each other. Meanwhile, in the non-segmented version, those emails are assigned to a single cluster. Second, in the non-segmented results, emails from bluemountain.com and match.com are together. They are clearly separated in the segmented results. Our method also produced a large non-tight cluster. A possible cause is the presence of duplicate parts such as footers and introductory paragraphs, which may weight more than the message contents to establish clustering association.

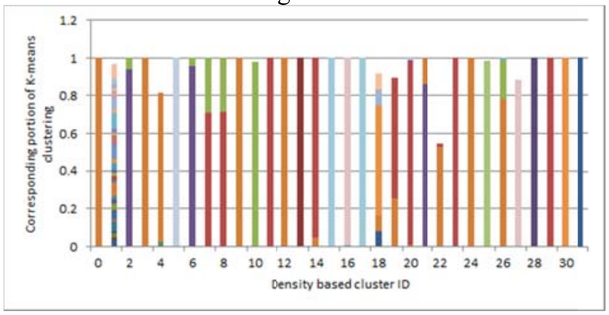


Figure 5 Relationship between density based clustering and K-means clustering results.

F. Comparison with Alternative Methods

We compared our method with K-means clustering and with Topic Model clustering using LDA. In both cases we used the libraries available in Apache Mahout [20]. Our goal was to compare emails in the clusters obtained with our method and in those obtained with the alternative methods.

For K-means clustering, the same number of clusters obtained with density based clustering (32) is used as the value of K. Figure 5 is a stacked column graph, where each colored fragment represents the portion of emails in a density based cluster that are also found in a unique K-means cluster. Each color represents a specific K-means cluster ID. A solid color column means that the density-based cluster is a subset of a specific K-means cluster. Conversely, columns with multiple colored fragments indicate that the correspondent density based cluster is dispersed through several K-means clusters. From the result, there are 11 density-based clusters dispersed in more than one K-means clusters. Cluster 1, with many colored fragments, is the largest cluster and it is dispersed in many K-means clusters. With the exception of cluster 1, we conclude that density based clustering is more effective to merge cohesive documents together. For instance, cluster 19 is composed of news from Multex investor, but K-means clustering fails to merge those emails in one cluster. Adding to that conclusion, Figure 6 shows that K-means generated big clusters that merged several density-based clusters. Given the different topics included, it would be very difficult to understand and describe these clusters coherently.

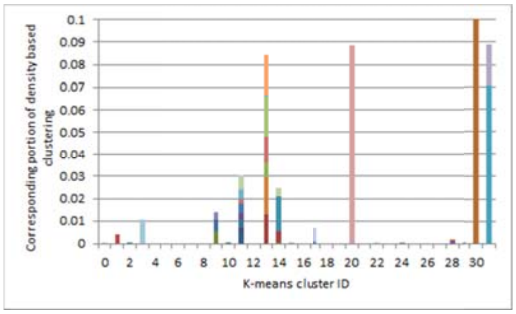


Figure 6 Mapping from K-means to density based clustering results

We also run the LDA method to infer 32 topics from the document collection. For comparison, we applied LDA to each density-based cluster with number of topics as 1. We compared the top 20 keywords found in each topic obtained in both methods to determine how many are shared. Figure 7 shows how the keywords in our approach map with the keywords extracted with LDA. We found that each cluster shares keywords among a wide range of LDA topics. This is expected since the LDA method assumes that documents can be represented as random mixtures of latent topics. For columns with height greater than 1.0 it means that in the LDA results several keywords belong to multiple topics simultaneously.

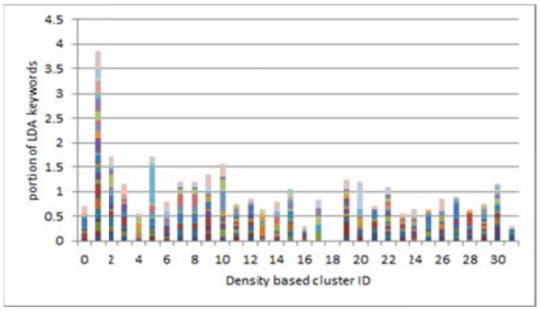


Figure 7 Mapping from density based to LDA clustering results

V. Discussion and Ongoing Work

We investigate the use of density based clustering with document segmentation to identify points of access in large document collections. We tested our method with the Enron email collection, and evaluated the results qualitatively and quantitatively. Compared to results of density-based clustering with non-segmented emails, to K-means, and to LDA, our method constitutes an improvement. Emails are tightly clustered as a function of the segmentation and the fact that noise points are discarded.

Using this method we found meaningful clusters revealing relevant information about activities and communication practices. Archivists can easily describe these clusters as access points in a finding aid. In turn, the combination of the different stories and classes of information provides a general description of the collection.

To address all the information contained in the collection, we are evaluating an iterative refinement solution in which we apply density based clustering to the large clusters to find sub-clusters. At each iteration we decrease the MNCT and Eps, and increase MinPts. We applied this approach to the large cluster mentioned in the results section and generated 91 sub-clusters covering 7.8% paragraphs and 13.1% emails. A preliminary evaluation indicates that the emails in those sub-clusters are cohesive with each other. We expect to treat noise points as one cluster in a similar fashion.

Acknowledgment

This work was supported by a National Archives and Records Administration (NARA) supplement to the National Science Foundation (NSF) Cooperative Agreement [OCI-0504077] and by the Longhorn Innovation Fund for Technology (LIFT) from the University of Texas at Austin.

References

[1] Archives Standards. http://www.icacds.org.uk/eng/standards.htm

[2] Jennifer Meehan, "Making the Leap from Parts to Whole: Evidence and Inference in Archival Arrangement and Description," *American Archivist*, vol. 72, no. 1, pp. 72-90, Spring/Summer 2009.

[3] M. Esteva and W. Xu, "Finding stories in the archive through paragraph alignment," in *Digital Humanities 2010 (DH2010)* , London, UK, July 7 to 10 2010.

[4] Apache Hadoop. [Online]. http://hadoop.apache.org

[5] J. Jayabharathy, S. Kanmani, and A. A. Parveen, "A survey of document clustering algorithms with topic discovery," *Journal of Computing*, vol. 3, 2011.

[6] Jian-Wu Xu, Vartika Singh, Venu Govindaraju, and Depankar Neogi, "A Hierarchical Classification Model for Document Categorization," in *10th International Conference on Document Analysis and Recognition*, 2009, pp. 486-490.

[7] D. M. Blei, A. Y. NG, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, no. 3, pp. 993-1022, 2003.

[8] C. Ozcaglar, "Classification of email messages into topics using Latent Dirichlet Allocation," Rensselaer Polytechnic Institute, New York, M.S. Thesis 2008.

[9] J. Allan, *Topic Detection and Tracking: Event-based Information Organization*.: Kluwer Academic Press, 2002.

[10] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A Density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining (KDD 96)*, pp. 226-231.

[11] Z. Su, Q. Yang, H. Zhang, X. Xu, and Y. Hu, "Correlation-based document clustering using web logs," in *34th Hawaii International Conference On System Sciences(HICSS-34)*, 2001.

[12] S. Daruru et al., "Distributed, scalable clustering for detecting halos in terascale astronomy datasets," in *IEEE International Conference on Data Mining Workshops(ICDMW 10)*, pp. 138-147.

[13] YC. Kwon et al., "Scalable clustering algorithm for n-body simulations in a shared-nothing cluster," in *International Conference on Scientific and Statistical Database Management(SSDBM 10)*, pp. 132-150.

[14] J. Dean and S. Ghemawat, "“MapReduce: Simplied data processing on large clusters," in *Conference on Symposium on Opearting Systems Design & Implementation(OSDI 04)*, 2004.

[15] J. Lin, "Brute Force and Indexed approaches to pairwise document similarity comparisons with MapReduce," in *ACM International Conference on Research and Development in Information Retrieval (SIGIR 09)*, pp. 155-162.

[16] S. Papadimitriou and J. Sun, "DisCo: Distributed co-clustering with MapReduce," in *IEEE International Conference on Data Mining(ICDM 08)*, 2008, pp. 512-521.

[17] G. Salton and Michael J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[18] Apache Mahout. [Online]. http://mahout.apache.org

[19] Enron Mysql. [Online]. http://bailando.sims.berkeley.edu/enron

[20] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*.: Manning Publications, 2010.