Interactive Document Clustering Revisited: A Visual Analytics Approach

Ehsan Sherkat
Dalhousie University
Halifax, Canada
ehsansherkat@dal.ca

Seyednaser Nourashrafeddin
Dalhousie University
Halifax, Canada
nourashr@cs.dal.ca

Evangelos E. Milios
Dalhousie University
Halifax, Canada
eem@cs.dal.ca

Rosane Minghim
Universidade de São Paulo
São Paulo, Brazil
rminghim@icmc.usp.br

ABSTRACT

UPDATED—February 19, 2018. Document clustering is an efficient way to get insight into large text collections. Due to the personalized nature of document clustering, even the best fully automatic algorithms cannot create clusters that accurately reflect the user's perspectives. To incorporate the user's perspective in the clustering process and, at the same time, effectively visualize document collections to enhance user's sense-making of data, we propose a novel visual analytics system for interactive document clustering. We built our system on top of clustering algorithms that can adapt to user's feedback. First, the initial clustering is created based on the user-defined number of clusters and the selected clustering algorithm. Second, the clustering result is visualized to the user. A collection of coordinated visualization modules and document projection is designed to guide the user towards a better insight into the document collection and clusters. The user changes clusters and key-terms iteratively as a feedback to the clustering algorithm until the result is satisfactory. In key-term based interaction, the user assigns a set of key-terms to each target cluster to guide the clustering algorithm. A set of quantitative experiments, a use case, and a user study have been conducted to show the advantages of the approach for document analytics based on clustering.

Author Keywords

Interactive document clustering; key-term; visualization; document projection; user study; text.

INTRODUCTION

A huge amount of text documents are produced each day, resulting in the generation of many documents per subject. One of the most effective solutions to manage and get insight into this huge amount of data is by automatically clustering them into meaningful clusters. There exist several clustering algorithms for document clustering, which try to group similar documents together with the help of different similarity measures. However, it is not always possible to automatically create the clustering that best matches the user's point of view or the application's goals, even with state-of-the-art clustering algorithms. A sensible solution for this problem is involving the human in the loop of clustering. In this approach, it is possible to generate clusters close to the user's perspectives, while in the process supporting the human in making sense of the document collection [23, 18, 5, 4].

There are several semi-supervised clustering algorithms that use a few labeled instances to improve the quality of their results [6, 39]. Some other algorithms exploit the user's feedback in terms of the split or merge signals [4]. In that case, the user only asks to merge two clusters or split a cluster without specifying how to split the cluster. Most of these approaches have been tested in theory, but not used in everyday activities. In practice, an interactive clustering algorithm that gives the user an intuitive way of interaction and is feasible in terms of user effort would be more appropriate. On the other hand, the constrained clustering algorithms assume that the constraints or the user feedback are known before clustering starts, and they do not support, in their set up, interaction during clustering or progressive clustering improvement by the user.

Topic modeling, sometimes viewed as soft clustering of documents, is a different way of categorizing similar content by extracting meaningful topics from the document collection. In reality it is sometimes hard to know how separated the topics generated with topic modeling are, which makes it difficult for user interactions. A few research papers have proposed interactive topic modeling systems for the end user by providing different visualization modules [23, 14]. In spite of these progresses, there is still a need to propose more effective visualization and interaction components to bring better insight into a document collection for the user.

IUI'18, March 7–11, 2018, Tokyo, Japan

 ISBN 978-1-4503-4945-1/18/03…$15.00
DOI: https://doi.org/10.1145/3172944.3172964

We propose a practical solution for interactive document clustering by combining an interactive clustering algorithm with a visual interface. We have chosen key-term based interaction because it has been demonstrated that not only is it effective in improving the results but it is more intuitive for the user to interact with the clustering algorithm [7, 29, 25, 31]. In our key-term based interaction method, the user only assigns a short list of key-terms (less than five) to the desired cluster to guide the clustering algorithm. Ability to interact with the clustering algorithm with a very short list of key-terms distinguishes the proposed method from other key-term based interactive systems. Additionally, interactions are convertible to each other. For example, document based interaction can be achieved easily by assigning top key terms of the selected document(s) to a cluster as opposed to just directly linking the document(s) to a cluster.

Lexical Double Clustering (LDC) [30] and a novel KMeans style interactive clustering algorithm called iKMeans are incorporated in our proposed system. We have select document clustering algorithms instead of topic modeling ones to avoid problems of inconsistency after several runs, empirical convergence criteria and difficulty to interact with their complicated formula and algorithm. LDC has shown better performance compared to state-of-the-art topic modeling algorithms such as Latent Dirichlet Allocation (LDA) [9] and Non-negative Matrix Factorization (NMF) [32] on several standard document clustering datasets. Both LDC and iKMeans are deterministic in each interaction and generate similar results for the same key-term based user feedback. Because of the simple nature of KMeans-like algorithms, it is easy to incorporate user feedback and scale the algorithm to very large datasets with a short running time, which is hard to achieve in topic modeling algorithms.

In the proposed system, it is possible to switch between the LDC and iKMeans algorithms on demand. Interactively switching algorithms during clustering is not used in any other interactive clustering system. For example, the user can start with the LDC algorithm which is robust to outliers and when the user gets sufficiently familiar with the dataset, switch to iKMeans to generate highly customized clusters. The ability to alternate between different clustering algorithms demonstrates the independence of our system of the specific clustering algorithm, which means it can incorporate any other key-term based interactive clustering algorithm.

We have designed a visual interface to integrate the key-term based user interactions with the clustering algorithms as well as to allow the users to explore the data set and the relationships between documents and clusters. Several visual components are developed to guide the user for more effective interaction with the clustering algorithm and give the user better insight into the document collection. We combined the t-distributed stochastic neighbor embedding (t-SNE) [27] with the Force-directed placement [16] in a novel way to better improve cluttering typical of such embeddings and provide a better distinction among clusters. We have conducted various experiments using real-world datasets including a use case and a user study to demonstrate the effectiveness of the proposed system. All the datasets and code developed or used in this paper are available on Github[1].

RELATED WORK

Each interactive clustering system consists of two integrated parts: an interactive clustering algorithm and a visual interface. The main focus of some research papers is in the clustering algorithm, some in the visualization part and others focus on the integration of both. In the following we targeted these three types of related work.

Constrained Clustering Algorithms

Semi-supervised clustering algorithms use labeled data to guide the clustering process. Two semi-supervised clustering algorithms, Seeded-KMeans and Constrained-KMeans, inspired by the KMeans algorithm, were introduced in [6]. In Seeded-KMeans, instead of randomly initializing the KMeans, the labeled data points are used to initiate the clustering. Constrained-KMeans is similar to the seeded one but it keeps the label of seeds unchanged in every iteration of the algorithm. These two algorithms are modeled based on the Expectation-Maximization (EM) algorithm on a mixture of k (number of clusters) Gaussians. Each data point has k possible conditional distributions and the initial supervision is to determine these conditional distributions for seed points. These two algorithms showed better performance in comparison to COP-KMeans [39]. The supervision in COP-KMeans is in terms of *must-link* (two data points must be in the same cluster) and *cannot-link* (two data points cannot be in the same cluster) constraint. In each step of KMeans, partitions are generated in a way that satisfies all the given constraint. The problem with these algorithms is that they assume that all the constrains are known in advance, hence they are not designed for interactive use.

Interactive Clustering Algorithms

A class of interactive clustering algorithms captures all the necessary interactions between the user and the clustering algorithm as cluster *split* and *merge* queries [5]. The user sends a Merge request if two clusters are a subset of a target cluster and a split request if a cluster contains data points belonging to distinct targeted clusters. The user does not enter how to split or point out possible mistakes in the have led to the request. Several different merge and split operations are introduced by Awasthi et al. [4] to reduce the number of merge and split requests. We believe that for textual data the user should not only be able to send the split request but also to specify how to split the cluster by providing some key-terms, aiming to reduce the number of split and merge requests. These algorithms have not been equipped with visualization or tested by end users.

Constrained Topic Modeling

Latent Dirichlet Allocation (LDA) [9] has been widely used for topic modeling. LDA models topics as distributions over words and documents as distributions over topics. One can consider the most probable topic of a document as a document cluster label. LDA has been extended to incorporate domain knowledge into the topic modeling through must-link

[1] https://github.com/ehsansherkat/IDC

and cannot-link constraints over terms [3]. The *must-link* constraint between two terms indicates that they tend to be generated by the same topic, while the cannot-link constraint says that two terms tend to be generated by different topics. The *Dirichlet forest prior* is used to encode must-link and cannot-link constraints in which the prior is a mixture of Dirichlet tree distributions. The must-link constraints only were used as a supervision to the LDA in [41]. They used a collection of tree-structured (based on *must-link* terms) multinomial distributions instead of multinomial distributions over words. Must-link and cannot-link constraints over documents were incorporated in the LDA model by changing the Gibbs sampling of the original LDA [41]. Contrary to the original LDA, the document topic distribution prior is updated in every iteration based on user feedback. They applied this system to improve topic model stability after adding new documents to the model.

A supervised version of LDA that incorporates labeled documents in the model has been proposed by [34]. In this model, the user can improve the quality of LDA by providing some topics for documents. These algorithms are not designed to obtain the user feedback in an interactive way. One of the major problems of topic modeling algorithms is inconsistency of results after each iteration which makes them difficult to use interactively. Algorithms described in this subsection have not been tested by the end user with the help of a visualization.

Interactive Topic Modeling Systems

iVisClustering is an interactive clustering system that introduced a visual interface on top of the LDA algorithm to involve the user in the loop [23]. The interaction with the LDA is performed only by changing terms' weights. This system has several different visualization modules called views. The *Graph View* using force-directed layout shows the general view of the document collection. The summary of clusters with their top terms are depicted in rectangle shaped boxes called *Cluster Summary View* beside the hierarchical style visualization of clusters (*Cluster View*). The user hands over a cluster in *Cluster Summary View* and the system shows document grids with the color spectrum which depicts their relatedness to the cluster. By clicking on a grid, the relatedness of that document to other clusters will be shown in a *Parallel Coordinates View* plus the content of documents in a *Document View*. The list of top terms of each cluster is in the *Term-Weight View* in which the user can change the weight of each term and impact the result of clustering. Because of the complicated nature of LDA formulas the interaction with this algorithm is not straightforward so changing term weights may confuse the user. In our system, the user only needs to define a set of key-terms without any need to assign a weight to them. Relative importance of terms is given by their order, without having to assing values. In addition to their lack of intuitiveness, topic modeling algorithms suffer from inconsistency in the results after each interaction. In contrast to *iVisClustering*, in our proposed system, all the visualization modules are coordinated with each other.

UTOPIAN [14] is an interactive topic modeling system, which is based on Non-negative Matrix Factorization (NMF) [32] instead of LDA. The interaction with NMF is based on changing term weights. Author´s used Graph View, Document View, and Term-Weight View as supporting visualizations. In the Graph View, the location of nodes is assigned by t-SNE [27]. The t-SNE algorithm is a method for dimensionality reduction mostly used for 2D and 3D projection of data points. UTOPIAN is developed by the same group of developers of iVisClustering system. In this system, they used NMF instead of LDA to handle the inconsistency problem of LDA algorithms. UTOPIAN still suffers from the empirical convergence problem and difficulty to meaningfully interact because of their complicated formula and algorithm. Unlike our proposed system, neither UTOPIAN nor iVisClustering were evaluated by end users. The overall differences between the key-term based document clustering systems and the UTOPIAN and the iVisClustering is described in more details in [31].

Interactive Document clustering Systems

An active learning scheme for selecting seed documents to be labeled by the user is presented in [19]. These seed documents will be used as an input to a semi-supervised KMeans algorithm. To facilitate the process of finding seed documents, a visualization is designed, which contains a *Term Cloud View* of each document and cluster, *Document View*, and *Pie Visualization* of clusters. In the *Pie Visualization*, each slice is a cluster and in the middle of the pie, there is a circle which contains all unlabeled documents. The user can assign these unlabeled documents to a desired cluster. *TopicPanorama* has also used a pie visualization for topics. All these topics are sub-topics of a few major topics determined by the user. Topics are linked together based on graph matching techniques in this system. This sub-topic graph structure is inside a hollow circle named *Radial Icicle Plot* in which for each major topic there is a color-coded arc. The user can zoom in on a topic by changing the length of the arc. Subtopics more similar to the major topics are near its arc and the common sub-topics are in the middle of the circle. In our proposed system, we used key-term based interactions, which is easier and more effective than document based interactions [31].

In our proposed system, we focus on key-term based interaction, which has been argued to be more effective and intuitive for the user than document based interaction or hybrid interaction in our previous work [31]. Novel contributions of this paper over [31] are the following: a flexible framework for key-term based document clustering, where users can interact based on key-terms with more than one word (e.g. bigrams); a new key-term based clustering algorithm, which shows how to combine a key-term interaction with document supervision; improved document projection by combining t-SNE with Force-directed placement in order to better visually distinguish groups of documents; and a user study that evaluates both the effectiveness of this type of interface for supervised clustering and the overall interactive document clustering approach.

OVERVIEW OF THE PROPOSED SYSTEM

The proposed system is a web-based user-centered document clustering system that integrates key-term based clustering algorithms with an interactive visualization. First, the user uploads documents, which the system preprocess and provides

an initial fully automatic clustering. Second, the user obtains insight into the document collection by inspecting the visual components including the document projection. Third, the user provides key-term based feedback to the clustering algorithm to guide the result of clustering. In the following, each component of the system is described.

Document Preprocessing

In the first step, we extract the plain text of each documents after removing punctuations and numbers. In the next step, we remove useless terms from the document collection dictionary. Even a few documents can lead to several thousand unique terms. This will cause a negative impact on the quality of document clustering [24] and increases the clustering time which is very crucial for interactive systems. In order to tackle this problem, we used an unsupervised feature selection method based on terms *tf-idf* score. This score shows the discriminative power of a term over each document [28]. Let D be the set of documents and d a document in D, the tf-idf score of the term w is as Equation 1.

$$tf\_idf(w,d,D) = f(w,d) \times log\frac{|D|}{|d \in D : w \in d|} + 1. \quad (1)$$

The $f(w,d)$ is the frequency of term w in document d. Now each document has a vector of terms' tf-idf score. We normalize each document vector based on the *Euclidean (L2) norm*. For each term, the *mean-tf-idf* score is calculated based on Equation 2.

$$mean\_tf\_idf(w,D) = \frac{1}{|D|} \times \sum_{d \in D} tf\_idf(w,d,D). \quad (2)$$

All terms with the mean-tf-idf score above the average mean-tf-idf of all terms are selected to shape the final document-term matrix. Approaches based on term frequency are reported [26, 42] to be as effective as more complicated methods while having linear time complexity for unsupervised feature selection in textual datasets.

Document Clustering

In this system, we used two document key-term based clustering algorithms, LDC and iKMeans. The user can switch between these algorithms in each iteration.

The LDC Algorithm

LDC contains two steps, the first step is term clustering and the second step is using term clusters to create a distilled set of terms to guide the assignment of documents to each term cluster. The *Fuzzy C-means* algorithm [8] is used for term clustering. This algorithm allows a term to belong to more than one cluster (soft clustering). The goal of Fuzzy C-means (FCM) is to optimize the objective function in Eq. 3.

$$FCM = \sum_{i=1}^{|W|} \sum_{j=1}^{k} u_{ij}^{m} cosineSim(w_i, c_j), \quad 1 < m < \infty \quad (3)$$

The w_i ($w \in W$: W=set of terms) is the ith column of the document-term matrix which is a $|D|$ dimensional vector of tf-idf scores. The c_j is a $|D|$ dimensional vector of the jth (k=*number of clusters*) cluster center. The *cosine distance* is used to calculate the similarity between each term and term clusters center. The membership matrix of u is recalculated in every iteration of FCM to optimize the objective function shown in Eq. 3. In the first iteration, all the values in matrix u are assigned randomly. In case there exist user feedback, the matrix u will be initialized based on user's feedback instead of random initialization. LDC distills the term clusters and then assigns documents to the closest term cluster after several steps.

Algorithm 1: *iKMeans* algorithm

input : K=Number of clusters;document-term matrix$^{|D|\times|W|}$;F=User defined *seed term*;Confidence(%)
output : Doc_clusters

if *firstIteration* **then**
  termClusters ← FCM(*document-term matrix*,K,m=1.1,maxIter=50);
  foreach *termCluster* ∈ *termClusters* **do**
    termCluster ← getTopTerms(Default=5)
else
  termClusters ← F
end
termClustersCenter ← CC(termClusters);
for $i \leftarrow 1$ **to** K **do** `// Expand key-terms`
  while $counter_1 < \alpha \times |W| \times 2^{(2-\frac{Confidence}{25})}$ **do**
    $termClusters_i$ += `// Cosine Sim.` nextSimilar(term∈W,$termClustersCenter_i$);
    $counter_1 \leftarrow counter_1 + 1$
  end
end
termToDocCenter ← CC($termClusters^T$);
for $i \leftarrow 1$ **to** K **do** `// Find seed docs`
  while $counter_2 < \beta \times |D| \times 2^{(2-\frac{Confidence}{25})}$ **do**
    $SeedDocs_i$ += `// Cosine Sim.` nextSimilar(doc∈D,$termToDocCenter_i$);
    $counter_2 \leftarrow counter_2 + 1$
  end
end
SeedDocsCenter ← CC(SeedDocs);
Doc_clusters ← KMeans(*document-term matrix*,K,SeedDocsCenter);
Function $CC(M^{I\times J})$**:** `// Calculate Center`
  for $i \leftarrow 1$ **to** I **do**
    $Center_{M_i} \leftarrow \frac{\sum_{j=1}^{|M_i|}(M_{ij}=[m_1,m_2,...,m_n])}{|M_i|}$
  end

iKMeans Algorithm

To demonstrate that our proposed system is independent of the clustering algorithm, we proposed an extension of Seeded-KMeans [6] algorithm based on *seed term* interaction. The main contribution of this algorithm is the way it converts key-term seeds to document seeds because the Seeded-KMeans only designed to work with documents seeds. The same pro-

cedure could be applied to other document supervised clusters and be subject to the same interface as our framework.

The first step of the algorithm is term clustering (lines 2-4 of Algorithm 1). Fuzzy C-means is used to find top terms (the top 5) for each cluster. In term clustering, we cluster the columns of the document-term matrix which represent a term as a vector of documents. This step is only for the first iteration of the algorithm; in the next iterations the top terms are determined by the user (lines 1-7 of Algorithm 1). This property represents the interactivity of the algorithm.

The center of these top term is calculated to extend the list of top terms for each cluster (line 8 of Algorithm 1). The center is the average *entry-wise* sum of each term vector (column of document-term matrix). A term vector is the representation of a term in the vector space defined by the documents. Terms that co-occur often in the same documents will have similar term vectors. As the number of top terms increases, the result of document clustering will be more biased to the top terms indicated by the user. The user can determine his/her confidence percentile in every interaction. The user confidence level regulates the number of top terms to be extended. Based on user confidence, the number of extension of top terms is calculated from the equation in line 10 of Algorithm 1 (Default $\alpha = 0.2$). A term is assigned to the list of top terms of a term cluster if it is more similar to its center according to *Cosine similarity*.

Each term cluster contains a list of terms in which one could imagine that all these terms may belong to a single imaginary document. The average tf-idf score of these terms in all documents is considered as the tf-idf score of terms in this imaginary document (line 11 of Algorithm 1). This imaginary document is now the center of document cluster. Several documents are assigned to each document cluster center based on Cosine similarity. The number of assigned documents for each document center is related to the user confidence level (line 17 of Algorithm 1). The KMeans algorithm uses these seed documents to initialize the document clusters and then assigns all the remaining documents to each cluster. Based on our experiments in Section 4.1, KMeans can produce promising results by having good seed documents.

Document Projection

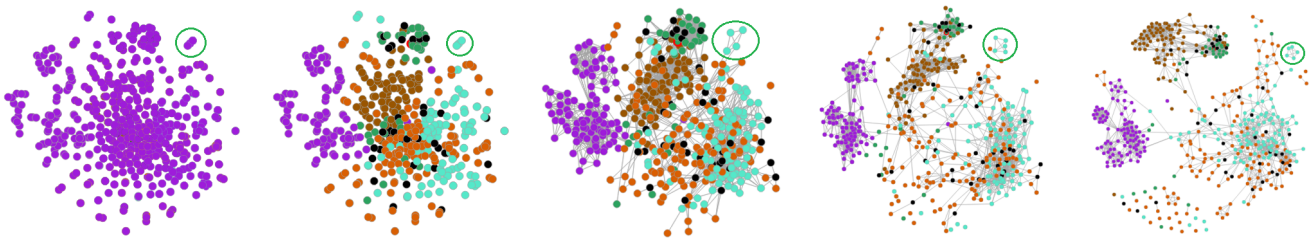
Projecting all documents in a 2D space gives the user a global view of the document clusters with their internal and external relations. Principal Component Analysis (PCA) [40] and t-Distributed Stochastic Neighborhood Embedding (t-SNE) [27] are among the most popular approaches for projection of documents. The t-SNE algorithm demonstrates better performance in visualizing clusters of data points than the PCA [27]. The t-SNE and PCA algorithms use the bag of words representation of documents to calculate the pairwise similarity between documents. In t-SNE, there is a chance that data points are loosely grouped and consequently, there may be many overlaps between data points from different clusters which makes it difficult for the user to fully understand the structure of clusters. The UTOPIAN [14] and TopicLens [21] systems tried to tackle this problem by multiplying the pairwise distance of data points belonging to the same cluster by a particular factor. In this way, data points belonging to the same cluster are grouped together which results in a clearer view of clusters. The problem with this approach is that, by changing the pairwise distance between data points, the positions of nodes are no longer as meaningful and the user cannot easily find similar documents belonging to different clusters on the visualization.

Extended t-SNE: In order to tackle the cluttering problem and avoid the pitfall of perturbing original similarities with bias towards cluster assignment, we combine Force-directed placement [16] with t-SNE. First, t-SNE is calculated in the conventional way for nodes; then, starting from the current location of nodes, Force-directed placement is applied. Force algorithms try to improve proximity of similar data point and increase separation for dissimilar data points. By starting from a favorable position, such as that calculated by t-SNE, new improved positions are obtained that may correct possible imprecise positioning in the embedding. The projection of 489 documents belonging to BBC sport dataset [17] based on original t-SNE is displayed in Fig. 1a. After labeling nodes based on their clusters, it becomes clearer that some nodes overlap (Fig. 1b). After some iterations of force directed algorithm, it is evident that segregation is improved (Figs. 1c, 1e, 1d). As the goal of Force-directed placement is reducing the number of cross-links, before applying the Force-directed forces, we add links to nodes. The links are added according to the cosine similarity between the bag of words vector representation of documents (nodes). The two nodes will have a link if they are similar above a certain threshold. After applying a few Force-directed steps the overlaps between nodes decrease significantly while still preserving the meaning of nodes location (based on t-SNE) (Fig. 1d). If the number of Force-directed iterations increases, the result of extended t-SNE will be similar to the original Force-directed layout (Fig. 1e).

Projecting documents based on Force-directed approach will place documents with similar cluster labels together while placing the isolated nodes far from the center (Fig. 1e). The problem with Force-directed placement is that the location of nodes is not directly related to their pairwise similarity (contrary to t-SNE). This means that two nodes placed near each other are not necessarily similar to each other unless there exists a link between them. In addition to the augmented t-SNE with the Force-directed projection of documents, we also provide the user of the document with the layout based exclusively on Force-directed, with flexible threshold for the links. If the user selects a node in one of these layouts, that node will be highlighted in the other layouts as well.

Visual Components

The implemented web-based user-centered system is depicted in Fig. 2. The system contains several visual components called *views*. The name of each component is given in its header. The size and the location of each view are designed as a result of our user study due to their importance and usage frequency (see Section 4.3). The *Graph view* is the most frequently used component, and views on its right part are more frequently used than the ones on the left. The user has the freedom of resizing and relocating every component of the system. The components are coordinated and changing a



(a) The original t-SNE (b) Cluster labels added (c) 2 Force-directed ticks plus adding edges (d) 10 Force-directed ticks (e) 100 Force-directed ticks

Figure 1. The comparative result of t-SNE (a, b) and the t-SNE combined with Force-directed (c, d, e). Cluster labels are based on clustering algorithm. Even with applying a few Force-directed iterations a clearer projection of documents is producing (d). If the number of Force-directed iterations increases, the result of t-SNE will be similar to the Force-directed layout (e). A subset of 490 randomly chosen documents of *BBC sport* data set is used in these figures.

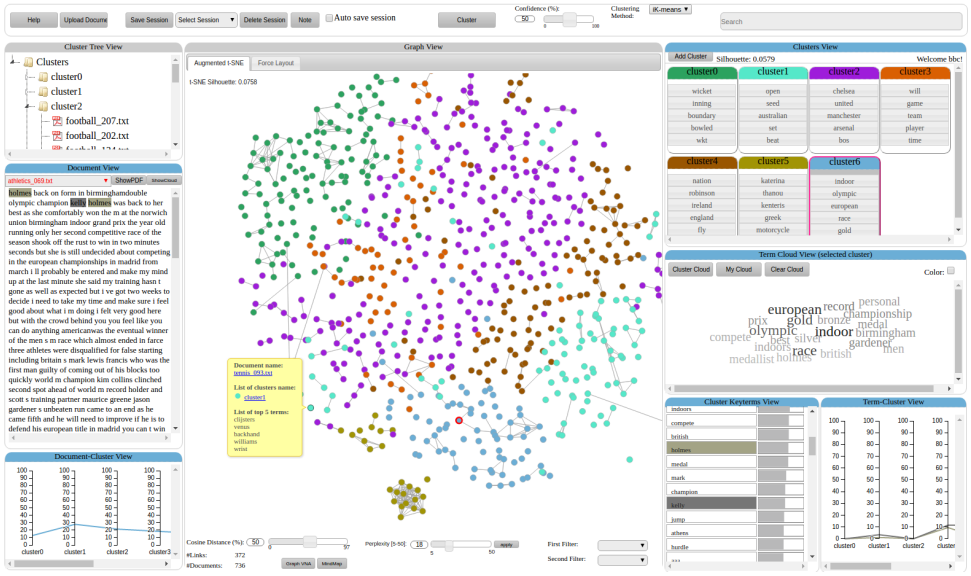


Figure 2. The visual interface of the proposed system. In the middle, the projection of 737 documents of the *BBC Sport* dataset is depicted (*Graph view*). On the left, we see the *Cluster tree view* for a hierarchical display of clusters and documents, the *Document view* for showing the plain text of documents, and the *Document-cluster view* to depict the relatedness of the selected document to each cluster. The name of each visual component is given in its header. On the right, we see the *Clusters view*, which demonstrates top terms of clusters, the *Term cloud view* for highlighting top terms of a selected cluster or set of documents, the *Cluster key-terms view* for listing top terms of a selected cluster with their level of importance (bar charts), and beside it the *Term-cluster view* to depict relatedness of a selected term(s) in *Cluster key-terms view* to each cluster. The views with colored header are all related to the selected cluster in the *Clusters view*. The selected cluster has a red margin and the same header color. The user can add, remove, rename or recolor a cluster or merge two clusters in *Clusters view*. The feedback to the clustering process by changing the number of clusters or adding/removing terms in *Clusters view*, as well as adding or removing cluster. The user can send changes made by pressing the *cluster* button on the top of the *Graph view*.

parameter in one could impact the other components as well. For example, the components with the similar color header are sharing information about the selected cluster with the same color (the one with red color margin in the picture) in the *Cluster view*, in Fig. 2. The user gains insights into the document collection and provides feedback to the clustering algorithm through the views. The feedback is in terms of adding, removing, and reordering of the clusters' top terms or adding a new cluster or removing a cluster (which change the number of clusters) in *Cluster view*. The user can iteratively interact with the clustering algorithm and in each iteration can apply a different clustering algorithm. The clustering result can be saved in each iteration so the user can roll back to previous results, and pursue different clustering directions.

The document's glyph in Graph view (the circles) contains information such as document cluster(s) color, document's name, and a list of top terms based on the documents' tf-idf score. The user can find the location of the selected document in the Document view by the red color stroke of a glyph, or load the textual content of a node in Document view by clicking on the name of a cluster in the node's tool-tip. If a document belongs to more than one cluster, its inner color turns to black. This is more convenient for the user to spot such documents than using Pie Visualization when there are many documents. The user can zoom in and out in the Graph view. The glyphs are connected according to Cosine similarity exceeding a threshold. If the user changes the threshold, links will be added/removed from graph view in real time. By setting the Cosine distance threshold to zero, only duplicate

documents will still have a link. Two documents are duplicate if they share exactly the same textual content with different file names. If the user clicks on a glyph, that glyph and its neighbors will be highlighted in both augmented t-SNE and Force-directed layouts (the second tab in Graph view). It is possible to attach and select more nodes to the selected nodes (*keep function*) or deselect a node from the selected nodes (*un-keep function*). The user can view the summary of selected nodes in *Term cloud view*. Let G be a set of selected glyphs (documents), and $M^{|G|\times|W|}$ a subset of document-term matrix containing selected documents. The score (its level of importance) of a term t belonging to the selected documents is defined as Equation 4.

$$Score(t_i) = \frac{1}{|G|}\sum_{j\in G} M_{ji} \tag{4}$$

In the *Term cloud view*, a term with higher score has a larger font size and darker color. The black and white term representation helps the user to spot the important terms faster than the color version of it. It is possible to switch between black and white to the colored versions, for instance in case of better spotting of *bigram* terms.

Selecting terms in *Cluster key-term view* highlights all documents containing those terms in the *Graph view*, and the Document view. This will help the user to understand the discriminative power of selected terms, which is related to visualizing the tf-idf score of selected terms. We chose the gray-scale color for highlighting the selected terms in the *Cluster key-term view*, and the Document view to differentiate them from the clusters' color. The user can search a term in document collection and directly add it to the list of top terms of a cluster. The document relatedness in Document-cluster view and term relatedness in *Term-cluster view* are calculated by the Chi-squared statistic (χ^2) using assigned clusters labeled by the clustering algorithm as the classes in the typical use of the Chi-squared statistic in supervised feature selection.

EXPERIMENTS

In this section, a quantitative evaluation, a case study and a user study are reported to examine the quality and effectiveness of the proposed system. The back end of the proposed web-based system is implemented in Python, while the front end in JavaScript, jQuery, HTML, and D3 [10]. The result of document clustering can be saved as a Zip file of clustered documents, as a Mind Map [1], or in VNA graph format [2]. The following datasets are used in our experiments.

NG5: This dataset is a subset of 20 *Newsgroups*[2] dataset including 5 categories with 80 randomly chosen documents for each category. *Newsgroups* dataset consists of nearly 20,000 messages of Internet news articles with 20 categories collected by Ken Lang [22].

R8: A subset of *Reuters-21578* dataset containing 8 categories with 2,189 test and 5,485 training documents (7,674 documents together) [12]. Reuters-21578 is a popular dataset for text classification from Reuters newswire in 1987 assembled by David Lewis [35].

[2] www.qwone.com/~jason/20Newsgroups/

Table 1. The comparison result of clustering algorithms with random initialization (The average 200 runs for each algorithm). Adjusted Random Score (ARS), Adjusted Mutual Information (AMI), Homogeneity (H), and Average Silhouette (S) are used as a metric.

Dataset Name	Metric	Fuzzy C-means	LDC	KMeans	iKMeans
NG5 (400 doc. 4 classes)	ARS	0.521	0.501	0.201	0.628
	AMI	0.614	0.577	0.297	0.710
	H	0.619	0.582	0.306	0.714
	S	0.090	0.074	0.034	0.087
R8 (2189 doc. 8 classes)	ARS	0.353	0.440	0.195	0.305
	AMI	0.438	0.481	0.387	0.447
	H	0.637	0.659	0.473	0.592
	S	0.092	0.083	0.077	0.092
WebKB (4199 doc. 4 classes)	ARS	0.310	0.324	0.265	0.320
	AMI	0.337	0.326	0.281	0.311
	H	0.354	0.337	0.283	0.314
	S	0.027	0.025	0.024	0.022

Table 2. The role of seeds (documents) quality in the result of document clustering. The *NG5* dataset is used in this experiment. Bad seeds are assigning 3 seed document with similar label to each cluster. Good seeds are assigning 5 correctly labeled seeds to each of cluster. Semi-bad and -good are between bad and good seeds.

Seed Type	Evaluation metric	Fuzzy C-means	Seeded KMeans
Bad seeds	Adjusted Random Score	0.558	0.006
	Adjusted Mutual Info	0.647	0.006
	Homogeneity	0.651	0.051
	Average Silhouette	0.104	0.010
Semi-bad seeds	Adjusted Random Score	0.486	0.067
	Adjusted Mutual Info	0.588	0.198
	Homogeneity	0.593	0.208
	Average Silhouette	0.102	0.052
Semi-good seeds	Adjusted Random Score	0.548	0.305
	Adjusted Mutual Info	0.641	0.414
	Homogeneity	0.646	0.422
	Average Silhouette	0.103	0.048
Good seeds	Adjusted Random Score	0.545	0.637
	Adjusted Mutual Info	0.639	0.711
	Homogeneity	0.644	0.715
	Average Silhouette	0.103	0.088

NewsSeparate: This dataset contains 381 news feeds manually categorized into 13 categories [33]. In the user study (Section 4.3), we randomly selected 100 documents belonging to 4 categories from this dataset.

WebKB: This dataset is 4199 faculty, student, project, and course web sites crawled from the computer science faculty website of four universities in January 1997 [15].

Yahoo Answers: A collection of 189,467 questions and answers extracted from Yahoo! answers website with 20 top-level and 280 sub-categories [13]. In the user study, we used 6 sub-categories containing general questions about *Computer*, *Education*, *Music*, *Food Receipts*, *General Health*, and *Society*. For each category we randomly selected 100 question and answer pairs.

BBC Sport: Introduced by Greene and Cunningham [17] which contains 737 news articles about 5 sport categories extracted from BBC website from 2004-2005.

Quantitative Evaluation

In our experiments we employed *Adjusted Random Score* (ARS), *Adjusted Mutual Information* (AMI), *Homogeneity* (H), and *Average Silhouette* (S) (the average of all documents' *Silhouette* score). ARS is a measure of similarity between the ground truth clustering and the clustering algorithm [20]. ARS is between 0 and 1, where value near 0 reflects random clustering. ARS is penalized by the number of false positive and false negative predictions. AMI is an adjusted version of *Mutual Information* to reduce the effect of the agreement by chance and it is 0 in random clustering [38]. Homogeneity is between 0 and 1. A clustering has a higher *Homogeneity* score if its clusters contain more documents which are members of a single class [36]. *Silhouette* is an unsupervised metric for evaluation of the document clustering algorithm and helping the user to find the optimum number of clusters [37]. The Silhouette is between -1 and 1, where the higher positive value for a document shows it is more similar to the assigned cluster than the other clusters.

We compared different clustering algorithms implemented within our proposed framework without the user interactions in Table 1. The experiments are a result of using unigrams with mean-tf-idf as feature selection, and algorithms are initialized randomly. First, iKMeans (Algorithm 1) receives the top 5 terms for each cluster from the output of the Fuzzy C-means algorithm, then it provides seed documents for the KMeans algorithm. In all of the datasets iKMeans performed better than KMeans which demonstrates the effectiveness of our proposed framework. Comparing iKMeans with LDC and Fuzzy C-means algorithms shows that there is no clear winner. What algorithm performs best is dependent on the dataset.

In the next experiment, the interactivity of the KMeans algorithm, which is called Seeded KMeans in Table 2 is evaluated. In this experiment, the documents' labels are provided as supervision to the algorithm so instead of random initialization it has been initialized based on these document labels. This is somehow simulating lines 1-21 of Algorithm 1 by directly providing the seed documents to the KMeans clustering algorithm. We provided four different sets of seed documents.

- *Bad Seeds*: In the case of Bad Seeds, 3 seed documents are assigned to each cluster where all of these seed documents have a similar label.
- *Semi Bad Seeds*: If the number of seed documents per cluster is reduced to 1, Semi Bad Seeds situation will occur.
- *Good Seeds*: For clustering with Good Seeds, 5 seed documents are assigned to each cluster, where each cluster's seeds are labeled to a single correct label different from other clusters' seeds.
- *Semi Good Seeds*: Reducing the number of seed documents to 1 results in a Semi Good Seeds setting.

The results in Table 2 demonstrate that by providing a few good labeled documents KMeans performance increases significantly. On the other hand, the Fuzzy C-means performance did not change with this number of documents and more seeds are needed to see a positive impact. From the user's point of view, the algorithm which could be improved with a smaller number of interactions is preferable.

Table 3. The improvement of the Avg. Silhouette after each interaction.

Avg. Silhouette	Interaction 1	Interaction 2	Interaction 3
t-SNE	0.3493	0.3533	0.3544
Clustering	0.1267	0.1382	0.1530

Use Case

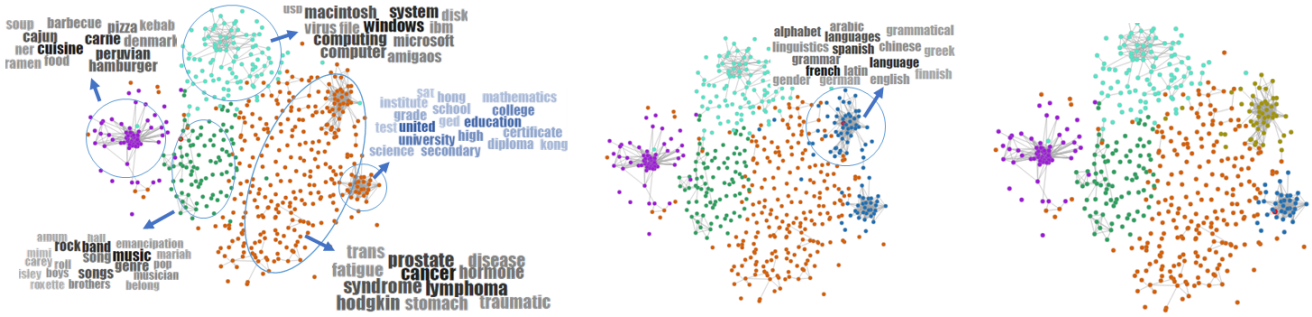
We asked a computer science researcher with good knowledge of document clustering to cluster the *Yahoo Answers* dataset without mentioning the subject of the dataset to her. The use case took 40 minutes including 20 minutes for system introduction. We recorded the user interactions and report the summary of them in this paragraph. First, the user asked the clustering algorithm to generate 4 clusters, shown in Fig. 3a. Second, the user checked the Term Cloud of clusters to know their topic and by looking at the Graph view, she noticed that there were two dense groups of documents in the cluster with the topics *Cancer*, *Syndrome*, and *Lymphoma*. Third, the user selected several nodes related to one of these groups of documents in the Graph view with the help of the Keep function. She noticed that these nodes were about *education* and *university* (the blue Term Cloud in Fig. 3a), so she selected the top three terms from Term Cloud and created the new cluster. The new clustering result after adding a new cluster is shown in Fig. 3b. The new cluster (blue color) now had two dense sets of nodes. Fourth, the user found out that terms such as *Education* were common between these two sets of documents with the help of Term-cluster view. On the other hand, one of the sets of documents was more about *Language*, *French*, and *Spanish* so she decided to create a new cluster. The user also removed the term *Education* from the blue color cluster to help the clustering algorithm separate these two sets of documents. The result of clustering after the third interaction of the user is in Fig. 3c. The improvement in the Silhouette score after each interaction (see Table 3), gave the user more confidence about the usefulness of the feedback to the clustering algorithm.

User Study

We invited 18 participants (9 female) for a user study. The participants were computer science students with at least acceptable knowledge of document clustering and strong English comprehension skills. The study was in an office with a single monitor with 1920x1200 resolution. In this study, we had two research questions: 1) evaluate the impact of users' interactions on the quality of document clustering. 2) determine if the visualization assists the users in obtaining better insight into the document collection and in improving the clustering result. To find the answer to these questions we designed two separate tasks. For each user, 20 minutes were allocated for training and 50 minutes were allocated to finish both tasks of the study.

Task 1. Interactive clustering.

In the first task, we gave each of the 18 participants 30 minutes to cluster the *Yahoo Answers*. Each document in this dataset was renamed to prevent the user from finding the correct cluster label of documents by their name. We did not provide any



(a) Initial clustering result. (b) Second interaction. (c) Third interaction.

Figure 3. The screen-shots of different interaction rounds conducted by the user. The *Yahoo Answers* dataset and the iKMeans algorithm is used in this use case.

Table 4. Statistics about the study. The left table shows users' vote for visualization modules. The middle table is average number of important users' interactions in the first task. The right table highlights the most frequent users' operations and the name of its parent module.

Module name	# Votes
Document-cluster & Document view	4
key-terms view & Term-cluster view	6
Cluster view	5
Term cloud view	6
Graph view	14

Action name	Avg. Count
Re-clustering	7.17 ± 4.78
Add term	25.39 ± 26.42
Remove term	5.61 ± 5.77
Add cluster	4.22 ± 2.05
Remove cluster	0.33 ± 0.77

Operation description	Module name	Percent
Highlight documents containing the selected term	Graph view	11.45%
Click on a cluster to load its information on views	Cluster view	10.95%
Mouse over a node to see its information (tool-tip)	Graph view	7.56%
Creating terms cloud	Term Cloud view	6.86%
Click term in key-term view & load Term-cluster view	key-terms view & Term-cluster view	6.69%

information about the topics of the dataset or its number of clusters. To provide similar conditions for each user, each one started with the result of random initialization of the clustering algorithm with 3 clusters. We recorded every operation that users conducted during this task. The average frequency of five important actions of the term-based interactive clustering that users conducted is in the middle section of Table 4. The following is a summary of findings in the first task.

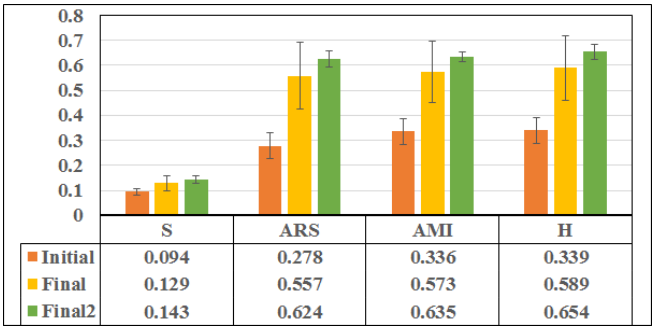
- The high standard deviation for the number of *Add term*, indicates that some users generated a longer list of key terms as the supervision to the clustering algorithm. There is a $+0.42$ (Pearson) correlation between the number of *Add term* and the Homogeneity score of the final clustering result after users' interactions.

- The users categorized this dataset to 5.7 ± 1.3 clusters in average which is close to the actual number of classes in this dataset.

- The most frequent operations users did during this task is summarized in the right section of Table 4.

- The users were asked in the post-task questionnaires to vote for each module of the system (multiple votes were allowed). The level of popularity and the importance of the Graph view by the users is displayed in left part of Table 4.

In order to investigate the impact of users' interactions on the quality of the final clustering result, we compared the initial clustering and the clustering after user interactions in the first task. The evaluation metrics are Adjusted Random Score (ARS), Adjusted Mutual Information (AMI), Homogeneity (H), and Average Silhouette (S) (See Fig. 4). Results demonstrate that user supervision significantly improved the result of clustering. During the study, four participants misunderstood the instructions and were not able to complete the task properly. A higher improvement with lower Standard Deviation was achieved after removing these participants (Final2 bar charts in Fig. 4).



	S	ARS	AMI	H
Initial	0.094	0.278	0.336	0.339
Final	0.129	0.557	0.573	0.589
Final2	0.143	0.624	0.635	0.654

Figure 4. The comparison between the initial clustering and the final clustering after user interactions. The final2 is after removing outliers.

Task 2. Obtaining insight on a document collection.
In the second task, we studied the impact of the user interface on helping the user obtain better insight into the document collection. The *NewsSeparate* dataset with 3 initial clusters is used in this task. The users were unfamiliar with the content of the dataset prior to the experiments. After the user performed the task, a questionnaire was provided to each user, who had 10 minutes to complete it (See Table 5). We asked the user to answer these questions twice (two modes); once with and

Table 5. Comparing the result (number of correct answers) of the system with (Vis.) and without (Base) visualization and their significance differences based on Wilcoxon signed-rank test with $p < 0.05$.

Question	Base	Vis.	P-value
1-Provide a title for each cluster	14/18	16/18	0.2119
2-Give the names of two clusters with most similar topics	4/18	9/18	0.0184
3-For each cluster, provide at least 1 term belong to other clusters as well	2/18	9/18	0.0054
Overall (for all questions)	**20**	**34**	**0.0023**

Table 6. The Post-task questionnaire. Questions 1-10 are from the Software Usability Scale (SUS) questionnaire. All answers are in 5-point Likert scale agreement scores (1: strongly disagree, 2: disagree, 3: neither agree nor disagree, 4: agree, 5: strongly agree).

Statement	Avg.
1-I think that I would like to use this system frequently	4.28±0.83
2-I found the system unnecessarily complex	2.28±0.83
3-I thought the system was easy to use	4.22±0.65
4-I think that I would need the support of a technical person to be able to use this system	2.22±1.06
5-I found the various functions in this system were well integrated	4.61±0.50
6-I found there was too much inconsistency in this system	1.44±0.62
7-I would imagine that most people would learn to use this system very quickly	4.17±0.92
8-I found the system very cumbersome to use	2.17±1.04
9-I felt very confident using the system	4.28±0.75
10-I needed to learn a lot of things before I could get going with this system	2.28±1.13
11-It is more meaningful to use phrases instead of single words to determining the clusters topics	3.89±0.83
12-Term based visualization and term labeling is a useful way in generating desired cluster topics	4.67±0.69
13-The user interface is a useful tool for document clustering in general	4.67±0.59
14-I would like to use the system in the future	4.61±0.70

once without (base mode) the visualization. In the mode without visualization, the ordered list of the top terms of each cluster and the folder of documents related to each cluster was provided. The users used the Windows file explorer and the Notepad++ text editor to dig into the clustering result in this mode. These are common tools for users to inspect automatic clustering results without the visualization. We randomly divided the participants into two halves. For the first half of the participants the clustering results with and for the second half without visualization was provided first. For the first question, the users were able to answer it properly in both modes. We believe the reason is that the ordered list of the top terms for each cluster is a good description of each cluster and it was easily accessible in both modes. For the next two questions, the result of the visualization mode is statistically significantly better, which shows the effectiveness of the visualization. The order of providing the visualization or the base mode at first did not have a statistically significant impact on the quality of users' answers. We did not inform the users that in both modes they were answering the same questions about the same dataset. The overall comparison result in Table 5 demonstrates that the visualization is significantly better than the base mode.

Three more questions were asked from users to test the *ease of use* and the *learnability* of the visualization. The first question asked which cluster has the highest number of documents and 83% of participants answered correctly. The second question was about the number of repetitive documents in the collection and 45% of participants were able to answer this question correctly. For the third question, participants needed to give the number of documents with more than one cluster labels. 67% of the participants answered this question correctly. The correct answer rate for the complicated questions such as the second question indicates that near a half of users were able to learn the very detail of the system only after 20 minutes of training.

The result of the post-task questionnaire is in Table 6. The goal of these questions was to get the users experience and opinion during the user study. The first 10 questions of Table 6 are selected from the Software Usability Scale (SUS) questionnaire [11]. The result of participants answers to the questions of Table 6 demonstrates the effectiveness and usefulness of the proposed system for interactive document clustering.

CONCLUSION AND FUTURE WORK

We introduced a novel solution for interactive document clustering. First, the proposed solution was evaluated in real world datasets by end users, demonstrating significant improvement in the quality of the clusters over fully automatic clustering. We built our system based on key-term interaction because of its intuitiveness for the user. Second, we introduced an interactive version of KMeans called iKMeans. The proposed method for iKMeans could be applied to other clustering algorithms and consequently could be employed in the proposed system. Third, we have combined the t-SNE algorithm with force directed display for improved projection of documents. The evaluation result demonstrates the effectiveness of the proposed system on improving the clustering result.

In the future, we intend to add a visualization module for better depicting the clusters' change during user' interactions. Incorporating the feedback obtained as a result of the user study to extend the functionality and usability of the system for better visualization and clustering of document collection is in the list of our future work. We would like to further extend our system for datasets with temporal aspect, in that case user can interact with the clustering algorithm to improve the result of the clustering for only a specific time frame.

ACKNOWLEDGMENTS

The authors thank all participants in the user study. The research was partially funded by the Natural Sciences and Engineering Research Council of Canada, the Boeing Company (Canada), CNPq and FAPESP (Brazil), and the International Development Research Center, Ottawa, Canada.

REFERENCES

1. Accessed: 2017-10-07. Mind Map file format Description. `http://http://freemind.sourceforge.net`. (Accessed: 2017-10-07).

2. Accessed: 2017-10-07. VNA file format Description. `https://gephi.org/users/supported-graph-formats/netdraw-vna-format/`. (Accessed: 2017-10-07).

3. David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating Domain Knowledge into Topic Modeling via Dirichlet Forest Priors. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, New York, NY, USA, 25–32. DOI:http://dx.doi.org/10.1145/1553374.1553378

4. Pranjal Awasthi, Maria-Florina Balcan, and Konstantin Voevodski. 2014. Local Algorithms for Interactive Clustering. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14)*. JMLR.org, II–550–II–558. http://dl.acm.org/citation.cfm?id=3044805.3044954

5. Maria-Florina Balcan and Avrim Blum. 2008. Clustering with Interactive Feedback. In *Proceedings of the 19th International Conference on Algorithmic Learning Theory (ALT '08)*. Springer-Verlag, Berlin, Heidelberg, 316–328. DOI: http://dx.doi.org/10.1007/978-3-540-87987-9_27

6. Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. 2002. Semi-supervised Clustering by Seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML '02)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 27–34. http://dl.acm.org/citation.cfm?id=645531.656012

7. Ron Bekkerman, Hema Raghavan, James Allan, and Koji Eguchi. 2007. Interactive Clustering of Text Collections According to a User-specified Criterion. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence (IJCAI'07)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 684–689. http://dl.acm.org/citation.cfm?id=1625275.1625385

8. James C. Bezdek. 1981. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

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

10. Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec. 2011), 2301–2309. DOI: http://dx.doi.org/10.1109/TVCG.2011.185

11. John Brooke and others. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

12. Ana Cardoso-Cachopo. 2007. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa. (2007).

13. M. Chang, L. Ratinov, D. Roth, and V. Srikumar. 2008. Importance of Semantic Representation: Dataless Classification. In *AAAI*. http://cogcomp.cs.illinois.edu/papers/CRRS08.pdf

14. Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. 2013. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec. 2013), 1992–2001. DOI: http://dx.doi.org/10.1109/TVCG.2013.212

15. Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Seán Slattery. 1998. Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence (AAAI '98/IAAI '98)*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 509–516. http://dl.acm.org/citation.cfm?id=295240.295725

16. Thomas M. J. Fruchterman and Edward M. Reingold. 1991. Graph drawing by force-directed placement. *Software: Practice and Experience* 21, 11 (1991), 1129–1164. DOI: http://dx.doi.org/10.1002/spe.4380211102

17. Derek Greene and Pádraig Cunningham. 2005. Producing Accurate Interpretable Clusters from High-dimensional Data. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'05)*. Springer-Verlag, Berlin, Heidelberg, 486–494. DOI: http://dx.doi.org/10.1007/11564126_49

18. Yuening Hu, Jordan Boyd-Graber, and Brianna Satinoff. 2011. Interactive Topic Modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (HLT '11)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 248–257. http://dl.acm.org/citation.cfm?id=2002472.2002505

19. Yeming Hu, Evangelos E. Milios, James Blustein, and Shali Liu. 2012. Personalized Document Clustering with Dual Supervision. In *Proceedings of the 2012 ACM Symposium on Document Engineering (DocEng '12)*. ACM, New York, NY, USA, 161–170. DOI: http://dx.doi.org/10.1145/2361354.2361393

20. Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification* 2, 1 (1985), 193–218. DOI:http://dx.doi.org/10.1007/BF01908075

21. M. Kim, K. Kang, D. Park, J. Choo, and N. Elmqvist. 2017. TopicLens: Efficient Multi-Level Visual Topic Exploration of Large-Scale Document Collections. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (Jan 2017), 151–160. DOI: http://dx.doi.org/10.1109/TVCG.2016.2598445

22. Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*. 331–339.

23. Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. 2012. iVisClustering: An Interactive Visual Document Clustering via Topic Modeling. *Computer Graphics Forum* 31, 3pt3 (2012), 1155–1164. DOI: http://dx.doi.org/10.1111/j.1467-8659.2012.03108.x

24. Y. Li, C. Luo, and S. M. Chung. 2008. Text Clustering with Feature Selection by Using Statistical Data. *IEEE Transactions on Knowledge and Data Engineering* 20, 5 (May 2008), 641–652. DOI: http://dx.doi.org/10.1109/TKDE.2007.190740

25. Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2004. Text Classification by Labeling Words. In *Proceedings of the 19th National Conference on Artifical Intelligence (AAAI'04)*. AAAI Press, 425–430. http://dl.acm.org/citation.cfm?id=1597148.1597218

26. Tao Liu, Shengping Liu, Zheng Chen, and Wei-Ying Ma. 2003. An Evaluation on Feature Selection for Text Clustering. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML'03)*. AAAI Press, 488–495. http://dl.acm.org/citation.cfm?id=3041838.3041900

27. Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.

28. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

29. Seyednaser Nourashrafeddin, Evangelos Milios, and Dirk Arnold. 2013. Interactive Text Document Clustering Using Feature Labeling. In *Proceedings of the 2013 ACM Symposium on Document Engineering (DocEng '13)*. ACM, New York, NY, USA, 61–70. DOI: http://dx.doi.org/10.1145/2494266.2494279

30. Seyednaser Nourashrafeddin, Evangelos Milios, and Drik V. Arnold. 2014. An Ensemble Approach for Text Document Clustering Using Wikipedia Concepts. In *Proceedings of the 2014 ACM Symposium on Document Engineering (DocEng '14)*. ACM, New York, NY, USA, 107–116. DOI: http://dx.doi.org/10.1145/2644866.2644868

31. Seyednaser Nourashrafeddin, Ehsan Sherkat, Rosane Minghim, and Evangelos Milios. in press. A Visual Approach for Interactive Keyterm-based Clustering. *ACM Transactions on Interactive Intelligent Systems* (in press).

32. Pentti Paatero and Unto Tapper. 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* 5, 2 (1994), 111–126. DOI: http://dx.doi.org/10.1002/env.3170050203

33. F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. 2008. Least Square Projection: A Fast High-Precision Multidimensional Projection Technique and Its Application to Document Mapping. *IEEE Transactions on Visualization and Computer Graphics* 14, 3 (May 2008), 564–575. DOI: http://dx.doi.org/10.1109/TVCG.2007.70443

34. Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009. Labeled LDA: A Supervised Topic Model for Credit Attribution in Multi-labeled Corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1 (EMNLP '09)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 248–256. http://dl.acm.org/citation.cfm?id=1699510.1699543

35. Tony Rose, Mark Stevenson, and Miles Whitehead. 2002. The Reuters Corpus Volume 1-from Yesterday's News to Tomorrow's Language Resources.. In *LREC*, Vol. 2. 827–832.

36. Andrew Rosenberg and Julia Hirschberg. 2007. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure.. In *EMNLP-CoNLL*, Vol. 7. 410–420.

37. Peter Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *J. Comput. Appl. Math.* 20, 1 (Nov. 1987), 53–65. DOI: http://dx.doi.org/10.1016/0377-0427(87)90125-7

38. Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.* 11 (Dec. 2010), 2837–2854. http://dl.acm.org/citation.cfm?id=1756006.1953024

39. Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained K-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML '01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 577–584. http://dl.acm.org/citation.cfm?id=645530.655669

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

41. Yi Yang, Shimei Pan, Yangqiu Song, Jie Lu, and Mercan Topkara. 2015. User-directed Non-Disruptive Topic Model Update for Effective Exploration of Dynamic Content. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI '15)*. ACM, New York, NY, USA, 158–168. DOI: http://dx.doi.org/10.1145/2678025.2701396

42. Yiming Yang and Jan O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 412–420. http://dl.acm.org/citation.cfm?id=645526.657137