Evaluating the Use of Clustering for Automatically Organising Digital Library Collections

Mark Hall[1,2], Paul Clough[2], and Mark Stevenson[1]

[1] (m.mhall|r.m.stevenson)@sheffield.ac.uk
Department for Computer Science
Sheffield University
Sheffield, UK
[2] p.d.clough@sheffield.ac.uk
Information School
Sheffield University
Sheffield, UK

Abstract. Large digital libraries have become available over the past years through digitisation and aggregation projects. These large collections present a challenge to the new user who wishes to discover what is available in the collections. Subject classification can help in this task, however in large collections it is frequently incomplete or inconsistent. Automatic clustering algorithms provide a solution to this, however the question remains whether they produce clusters that are sufficiently cohesive and distinct for them to be used in supporting discovery and exploration in digital libraries. In this paper we present a novel approach to investigating cluster cohesion that is based on identifying instruders in a cluster. The results from a human-subject experiment show that clustering algorithms produce clusters that are sufficiently cohesive to be used where no (consistent) manual classification exists.

1 Introduction

Large digital libraries have become available over the past years through digitisation and aggregation projects. These large collections present two challenges to the new user [22]. The first is resource discovery: finding the collection in the first place. The second is then discovering what items are present in the collection. In current systems, support for item discovery is mainly through the standard search paradigm [27], which is well suited for professional (or expert) users who are highly familiar with the collections, subject areas, and have specific search goals.

However, for the novice (or non-expert) user exploring, investigating, and learning [16, 21] tend to be more useful search modalities. To support these modalities the items in the collection must be classified according to a relatively consistent schema, which is frequently not the case.

In many domains no standard classification system exists and, even if it does, collections are often classified inconsistently. Additionally, where collections have been formed through aggregation (e.g. in large-scale digital libraries) the items will frequently be classified using different and incompatible classification systems. Manual (re-)classification would be the ideal solution, however the time and expense requirement when dealing with hundreds of thousands or millions of items means that it is not a viable approach.

Automatic clustering techniques provide a potential solution that can be applied to large-scale collections where manual classification is not feasible. The advantage of these techniques is that they automatically derive the cluster structure from the digital library's items. On the other hand the quality of the results can be variable and thus the choice of which clustering technique to employ is central to providing an improved exploration experience.

The research questions posed at the start of this work was: Do automatic clustering techniques produce clusters that are cohesive enough to be used to support the exploration of digital libraries? We define a cohesive cluster as one in which the items in the cluster are similar, while at the same time clearly distinguishable from items in other clusters. Our paper provides two major contributions in this area. Firstly, we propose a novel variant of the intruder detection task [6] that enables the measurement of the cohesion of automatically generated clusters. Secondly, we apply this task to evaluate the cluster model quality of a number of automatic clustering and topic modelling algorithms.

Our results show that the clusters are sufficiently good to be used in digital libraries where manually assigned classifications are not available or not consistent. The remainder of the paper is structured as follows: The next section provides background information on the use of clustering in digital libraries, their evaluation, and the clustering techniques evaluated in this paper. Section 3 describes the methodology used in the evaluation experiment and section 4 the experiment results. Section 5 concludes the paper.

2 Background

The issues large, aggregated digital libraries present to the user were first highlighted in [22] who suggested manual classification by the user and automated clustering as approaches for dealing with the large amounts of information provided by these digital libraries. Since then a number of digital library exploration interfaces based on clustering documents [26, 9, 8] and search results [11, 28] have been proposed. Most of these approaches were evaluated in task-based scenarios and shown to improve task performance, however the cluster quality itself was not evaluated.

2.1 Cluster Evaluation Metrics

Cluster evaluation has traditionally focused on automatic evaluation metrics. They are frequently tested on synthetic or manually pre-classified data [17, 1]

or using statistical methods [29, 13]. However, these do not necessarily capture whether the resulting clusters are cohesive from the user's perspective.

There have been attempts at using human judgments to quantify the cohesion of automatic clustering techniques. Mei et al. [18] evaluate the cohesion of Latent Dirichlet Allocation topics in the context of automatically labelling these topics. The number of changes evaluators make to a clustering has also been used to judge cluster cohesion [24].

Chang et al. [6] devised the "intruder detection" task, where evaluators are shown the top five keywords for an LDA topic to which a keyword from a different topic is added. They then have to identify the added "intruder" keyword and the success at identifying the intruder is used as a proxy to evaluate the topic's cohesion. The more cohesive a topic, the more obvious it is which keyword is the intruder. The results of their work have been compared to a number of automatic similarity algorithms and Pointwise-Mutual-Information (PMI) was identified as a good predictor for the agreement between the evaluators [20].

2.2 Classification Models

This paper investigates three unsupervised clustering algorithms: Latent Dirichlet Allocation (LDA) [5], K-Means clustering [14], and OPTICS clustering [2]. Hierarchical and spectral clustering algorithms were also investigated, but not tested due to them either being too computationally complex for the data-set size or producing only degenerate clusterings.

Latent Dirichlet Allocation (LDA) is a state-of-the-art topic modelling algorithm, that creates a mapping between a set of topics T and a set of items I, where each item $i \in I$ is linked to one or more topics $t \in T$. Each item is input into LDA as a bag-of-words and then represented as a probabilistic mixture of topics. The LDA model consists of a multinomial distribution of items over topics where each topic is itself a multinomial distribution over words. The item-topic and topic-word distributions are learned simultaneously using collapsed Gibbs sampling based on the item - word distributions observed in the source collection [10]. LDA has been used to successfully improve result quality in Information Retrieval [3, 30] tasks and is thus well suited to support exploration in digital libraries. Although LDA provides multiple topics per item, in this paper items will only be assigned to their highest-ranking topic and the topics will be referred to as clusters for consistency with the other algorithms.

***K*-Means Clustering** is a frequently used clustering method [31] that takes one input parameter k and assigns the n input items to k clusters and has been used in IR [12]. Items are assigned to the clusters in order to maximise the intra-cluster similarity while minimising the inter-cluster similarity. Cluster similarity is calculated relative to the cluster's mean value.

K-Means uses random initial cluster centres and then iteratively improves these by assigning items to the most similar cluster and moving the cluster centres to the mean of the items in the cluster.

OPTICS Clustering is a density-based clustering algorithm that does not directly produce a cluster assignment, but instead provides an ordering for the items that can then be used to create clusters with arbitrary density thresholds. The algorithm defines a reachability value for each item which specifies the distance to the next item in the ordering. Large reachability values represent the boundaries between clusters and depending on what reachability threshold is chosen, a larger or smaller number of clusters is generated.

3 Methodology

In this paper we propose a novel version of the "intruder detection" task that evaluates the cohesion of the items in a cluster instead of just the cluster's keywords. To generate this "intruder detection" task, a cluster (the *main* cluster) is chosen at random from the clustering model and four items are chosen at random from the items allocated to that cluster. A second cluster, the *intruder* cluster, is also chosen at random and a random item chosen from that cluster as the *intruder* item. The five items, termed a *unit*, are then shown to the participants and they are asked to identify the *intruder* item. For each of the tested models the evaluation set consisted of 30 such *units*.

3.1 Source Data

The source data used in the experiments is a collection of 28,133 historical images with meta-data provided by the University of St Andrews Library [7]. The majority (around 85%) of images were taken before 1940 and span a range of 160 years (1839-1992). The images mainly cover the United Kingdom, however there are also images taken around the world. Most (89%) of photographs are black and white, although there are some colour photographs. Of the available meta-data fields only the the title, description, and manually annotated subject classification are used in the experiments. On average, each image is assigned to four categories (median=4; mean=4.17; $\sigma = 1.631$) and the items' title and description tend to be relatively short (word-count: median=23; mean=21.66; $\sigma = 9.5857$). Examples are shown in Figures 1 and 2.

The collection was chosen for a two main of reasons: first, the collection has a manually annotated subject classification that provides an evaluation baseline; second, the data provides a realistic test case, as it was taken from an existing library archive (enabling the generalisation of results to other digital libraries), is large enough to make manual classification time-consuming, and at the same time small enough that it can be processed in a reasonable time-frame.

3.2 Data Preparation

Each item's title and description were processed using the NLTK [15] to carry out sentence splitting and tokenization. The resulting bags-of-words are the input into the three clustering algorithms. All processing was performed on an Intel i7 @1.73 GHz with 8GB RAM. Processing times are shown in Tab. 1.

				
Ottery St Mary Church.	Church of Ireland, Killough, Co Down.	Melrose Abbey.	Elgin Cathedral.	**Isle of Arran. Corrie from the Water,**

Fig. 1. Example of a cohesive *unit* taken from the "K-Means TFIDF" model. The intruder is the last image, in **bold font**.

				
Luzern, *Lucerne* Untitled.	Moorland beside Sand Sike (Syke) and River Tees; plant habitat.	**Panorama of the Cairngorm Hills, viewed from Docharn Craig, Strathspey.**	Kobresia caricina, plants in flower, on moorland between Sand Sike (Syke) and River Tees, Teesdale.	*Luzern*, *Lucerne* Untitled.

Fig. 2. Example of a non-cohesive *unit* taken from the "K-Means TFIDF" model. The intruder is the middle image, in **bold font**.

Model	**Wall-clock time**
LDA 300 clusters	00:21:48
LDA 900 clusters	00:42:42
LDA + PMI 300 clusters	05:05:13
LDA + PMI 900 clusters	17:26:08
K-Means - TFIDF	09:37:40
K-Means - LDA	03:49:04
OPTICS - TFIDF	12:42:13
OPTICS - LDA	05:12:49

Table 1. Processing time (wall-clock) for the tested clustering algorithms and initialisation parameters

LDA Two LDA-based clusterings were created using Gensim [23], one with 300 clusters ("LDA 300"), one with 900 clusters ("LDA 900"). The reason for testing two cluster numbers is that 300 clusters is in line with the number of clusters in other work using LDA [30]. At the same time our work on visualising clusters has hinted that clusters with around 30 items work best, which with 28000 items leads to 900 clusters. Although LDA provides a list of topics with probabilities for each item, the items are assigned only to their highest-ranking topic in order to maintain comparability with the clustering results.

Previous work [20] indicates that pointwise mutual information (PMI) acts as a good predictor of cluster cohesion. A modified cluster assignment model designed to increase the cohesiveness of the assigned clusters was developed. This approach is based on repeatedly creating LDA models and only selecting those clusters that have sufficiently high PMI scores.

The algorithm starts by creating an LDA model for all items using n clusters. The clusters are then filtered based on the median PMI score of their keywords $t_1 \dots t_5$ (eq. 2), creating the filtered set T_g of "good" clusters (eq. 3). Items for which the highest ranked cluster $t \in T_g$ are assigned to that cluster. A new LDA model is then calculated using the items for which their highest ranked cluster $t \notin T_g$ using a reduced number of clusters $n - |T_g|$.

$$\operatorname{pmi}(x, y) = \log \frac{p(x, y)}{p(x)\, p(y)} = \log \frac{p(x \mid y)}{p(x)} \tag{1}$$

$$\operatorname{coh}(t) = \operatorname{median}\{\operatorname{pmi}(t_i, t_j) : i, j \in 1 \dots 5 \wedge i \neq j\} \tag{2}$$

$$T_g = \{t \in T : coh(t) > 0\} \tag{3}$$

This process is repeated until either all items have been assigned to a cluster or the LDA model contains no clusters with a $coh(t) > 0$, in which case all items are assigned to their highest ranked cluster and the algorithm terminates. Two models using this algorithm with 300 and 900 clusters were created ("LDA + PMI 300" and "LDA + PMI 900").

***K*-Means** Two k-means classifications were produced, both with 900 clusters. The first used term-frequency / inverse-document-frequency (TFIDF) vectors, calculated from the items' bags-of-words, to define each item ("K-Means TFIDF"). As Tab. 1 shows the time required to create this model was very high, thus a faster k-means clustering was created using the item-topic probabilities from a 900-topic LDA model to define each item ("K-Means LDA").

OPTICS The OPTICS clustering used the same input data as the k-means clustering, creating two models ("OPTICS TFIDF" and "OPTICS LDA"). The reachability threshold required to create a fixed set of clusters was automatically determined for both models using an unsupervised binary search algorithm.

Upper- and Lower-bound Data An upper bound data-set was created based on the manually annotated subject classifications provided in the original meta-data. This classification has a total of 936 distinct clusters from which the 30 tested *units* were selected using the random algorithm described above.

To aid in the interpretation of the results a lower bound was determined statistically as the number of cohesive clusters where the binomial likelihood of seeing that number of correctly identified units out of 30 is less than 5%, resulting in a lower bound of 3 correctly identified *units*.

Control Data To ensure that the participants took the task seriously and did not simply select an answer at random, a set of 10 control *units* were created. These were randomly selected from the manual subject classifications and then manually filtered to ensure that the intruder was as obvious as possible.

3.3 Experimental Set-up

The experiment was constructed using an in-house crowdsourcing interface. In the experiment each *unit* was displayed as a list of five images with their captions, and five radioboxes that the participants used to choose the intruder. Participants were shown five *units* on one page, one of which was always a control *unit*. The four model *units* were randomly sampled from the full list of *units* (the model *units* and upper bound *units*). The sampling took into account the number of judgements already gathered for the *units* to ensure a relatively even distribution of judgments. The experiment was run using a population recruited from staff and students at our university.

A total of 821 people participated in the experiment, producing a total of 10,706 ratings. 121 participants answered less than half of the control questions they saw correctly and have thus been excluded from the analysis, reducing the number of ratings analysed to 8,840, with each *unit* rated between 21 and 30 times, with the median number of ratings at 27. The large variation is due to how the filtered participants' ratings were distributed, but has no impact on the results as the evaluation metric takes the number of samples into account.

3.4 Evaluating Cohesion

The human judgements were analysed to determine which *units* were judged to be cohesive. The metric used to determine cohesiveness is strict. A *unit* is judged to be *cohesive* if the correct intruder is chosen significantly more frequently than by chance and if the answer distribution is significantly different from a uniform distribution (Fig. 1). The first aspect is tested using a binomial distribution and testing whether the likelihood of seeing the observed number of correct intruder judgements relative to the total number of judgements for the *unit* by chance is less than 5%. This does not necessarily guarantee a cohesive cluster, as it does not take into account the distribution of the remaining answers. If these were evenly distributed, then even though the intruder was detected by a significant

number of participants, the remaining participants were evenly split and thus the *unit* cannot be classified as cohesive. A χ^2-test was used to determine whether the answer distribution was significantly different ($p < 0.05$) from the uniform distribution. If both conditions hold then the *unit* and with it the *main* cluster the *unit* was derived from, are said to be cohesive.

In addition, a second metric was used to judge whether *units* were *borderline cohesive*. A *unit* (and its *main* cluster) are defined as *borderline cohesive* if the total number of judgements allocated to two of the five possible answers makes up more than 95% of all judgements for that *unit* and one of the two answers is the intruder. This covers the case where in addition to the intruder item there is a second item that could also be the intruder. The *main* cluster such a *unit* is derived from might not be ideal, but is probably acceptable to the user, especially as the evaluation will show that even manually created clusters can be non-cohesive. All remaining *units* are classified as "non-cohesive" (Fig. 2).

4 Results

Table 2 shows the number of "cohesive", "borderline" and "non-cohesive" clusters per model. The results clearly show that k-means clustering based on TFIDF produces the most cohesive clusters. LDA with a large number of clusters also works well. The impact of filtering by PMI seems to be negligible. OPTICS clustering does not work well on the tested data-set.

Model	**Cohesive**	**Borderline**	**Non-Cohesive**
Upper-bound	27	0	3
Lower-bound	3	0	27
LDA 300 clusters	15	6	9
LDA 900 clusters	20	4	6
LDA + PMI 300 clusters	16	4	10
LDA + PMI 900 clusters	21	2	7
K-Means - TFIDF	24	3	3
K-Means - LDA	20	0	10
OPTICS - TFIDF	14	2	14
OPTICS - LDA	16	0	14

Table 2. Experiment results for the various clustering algorithms and initialisation parameters. "Cohesive" lists the number of clusters were the *intruder* was consistently identified, "borderline" the number of clusters with two potential intruders, and "non-cohesive" the number of clusters that are neither "cohesive" nor "borderline".

Table 1 shows the time required to generate each of the clusterings. The pure LDA models are fastest, while LDA + PMI with 900 clusters is the slowest algorithm. OPTICS and k-means lie between these extremes. Using LDA item-topic distributions for item similarity is faster than using TFIDF vectors.

4.1 Discussion

All models show a clear improvement on the lower bound and can thus be said to provide at least some benefit if no manual classification is available. However, the OPTICS and "LDA 300" models achieve cohesion for only about 50% of the clusters. OPTICS is clearly not a good choice for the type of data tested, as it is either slower or less cohesive than the other techniques.

The upper bound achives a very high score (90% of clusters cohesive), however even here there are three *units* that were not cohesive and these were further investigated (Tab. 3). The non-cohesive *unit* #1 is mis-classified in the original data and the *intruder* item was from the same geographic area as the *main* cluster items, making it impossible to determine which is the *intruder*. That this was picked up by the experiment participants is a good indication that the "intruder detection" task can distinguish cohesive from non-cohesive clusters. Analysis of the other two non-cohesive *units* shows that for both neither the image nor the caption provide sufficient information to identify the *intruder*. However, when the classification label is known, then the *intruder* can be determined. What this implies is that as long as there is some kind of logical link between the items, a certain amount of variation between the items is acceptable to human classifiers.

#	**Main cluster**	**Intruder cluster**
1	Renfrews all views	Isle of May all views
2	Garages - commercial	Colleges - technical
3	Soldiers	Belfries

Table 3. Subject labels attached to the non-cohesive upper-bound *units*

"K-Means TFIDF" is clearly the best of the models, achieving cohesion for 80% of the *units* and including the 3 borderline cohesive *units* pushes the rate up to 90%, matching the manual classification. Post-hoc analysis of the three border line *units* shows that in all three cases the *main* cluster item also identified as an intruder is linked to the other *main* cluster items via the description text, which the participants did not see. This means that those three clusters should also be acceptable when the users have access to all of the items' meta-data. The drawback with k-means is the long processing time rquired to create the model. Using the LDA document-cluster distribution instead of TFIDF leads to a significant reduction in processing time, however the quality of the resulting model suffers and is lower than the pure LDA results. K-Means is thus only viable for smaller collections, although the exact limit depends on what optimisations [25] can be achieved through improved initialisation [4] or parallelisation [32].

Processing speed is the strength of the pure LDA models, with the best-performing "LDA 900" model faster than "K-Means TFIDF" by a factor of almost 14 (Tab. 1). It does not achieve quite as good classification results (66% of *units* cohesive), but for data-sets that are too large to be clustered using k-means it is a viable alternative. Of the four borderline *units* two were of similar

type as the k-means borderline *units*, however in the other two the item from the *main* cluster identified as the intruder had no connection to the other *main* cluster items, leading to a total of 22 (73%) acceptable clusters.

The results of the LDA models also show the necessity for models where the individual clusters do not have too many items. The "LDA 300" and "LDA + PMI 300" models are significantly worse than their 900-cluster counterparts. This also validates the use of 900 clusters in the k-means and OPTICS models.

While [19, 20] show that PMI is a good predictor for the inter-annotator agreement, using it to filter clusters shows only minimal improvement. For both the 300 and 900 cluster models a single additional cohesive *unit* is created. Additionally in the 900 cluster case, PMI reduces the number of borderline units. Considering the increase in processing time this is not a viable method.

5 Conclusion

Large digital libraries have become available over the past years through digitisation and aggregation projects. These large collections present a challenge to the new user who wishes to discover what is available in the collections. Supporting this discovery task would benefit from a consistent classification system, which is frequently not available. Manual re-classification is prohibitively expensive and time consuming. Automatic cluster models provide an alternative method for quickly generating classifications.

This paper investigated whether clustering algorithms can generate cohesive clusters, where a cohesive cluster is one in which the items in the cluster are similar, while at the same time being distinguishably different from items in other clusters. Latent Dirichlet Allocation (LDA), K-Means clustering, and OPTICS clustering were investigated. To enable the comparison we proposed a novel version of the "intruder detection" task, where the experiment participants have to identify an item taken from a cluster and inserted into a set of four items taken from a different cluster. The results show that this task provides a good measurement for the cohesion of cluster models and can successfully identify non-cohesive clusters and mis-classifications.

Using this evaluation metric we showed that k-means clustering on TFIDF vectors produces the highest number of cohesive clusters, but is computationally intensive and thus only viable for smaller collections. LDA-based models with large cluster numbers provide the best cohesion – processing time trade-off, allowing them to be applied to large digital libraries. We believe that both algorithms create models where a sufficiently large number of clusters are cohesive to allow them to be used where no (consistent) classification is available. We intend to investigate if post-processing the clusters can further improve cohesion.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agree-

ment n° 270082. We acknowledge the contribution of all project partners involved in PATHS (see: http://www.paths-project.eu).

References

1. Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval 12* (2009), 461–486. 10.1007/s10791-008-9066-8.
2. Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. Optics: ordering points to identify the clustering structure. *SIGMOD Rec. 28*, 2 (June 1999), 49–60.
3. Azzopardi, L., Girolami, M., and van Rijsbergen, C. Topic based language models for ad hoc information retrieval. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on* (july 2004), vol. 4, pp. 3281 –3286 vol.4.
4. Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. Scalable k-means++. In *VLDB 2012* (2012).
5. Blei, D. M., Griffiths, T., Jordan, M., and Tenenbaum, J. Hierarchical topic models and the nested chinese restaurant process. In *NIPS* (2003).
6. Chang, J., Boyd-Graber, J., Wang, C., Gerrish, S., and Blei, D. M. Reading tea leaves: How humans interpret topic models. In *NIPS* (2009).
7. Clough, P., Sanderson, M., , and Reid, N. The eurovision st andrews collection of photographs. *ACM SIGIR Forum 40*, 1 (2006), 21–30.
8. Eklund, P., Goodall, P., and Wray, T. Cluster-based navigation for a virtual museum. In *Adaptivity, Personalization and Fusion of Heterogeneous Information* (Paris, France, France, 2010), RIAO '10, Le Centre de Hautes Etudes Internationales d'Informatique Documentaire, pp. 211–212.
9. Granitzer, M., Kienreich, W., Sabol, V., Andrews, K., and Klieber, W. Evaluating a system for interactive exploration of large, hierarchically structured document repositories. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on* (0-0 2004), pp. 127 –134.
10. Griffiths, T., and M. Steyvers, M. Finding scientific topics. In *Proceedings of the National Academiy of Science* (2004), vol. 101, pp. 5228–5235.
11. Handl, J., and Meyer, B. Improved ant-based clustering and sorting in a document retrieval interface. In *Parallel Problem Solving from Nature — PPSN VII*, J. Guervós, P. Adamidis, H.-G. Beyer, H.-P. Schwefel, and J.-L. Fernández-Villacañas, Eds., vol. 2439 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2002, pp. 913–923. 10.1007/3-540-45712-7_88.
12. Hassan-Montero, Y., and Herrero-Solana, V. Improving tag-clouds as visual information retrieval interfaces. In *Proceedings InfoSciT* (2006).
13. He, J., Tan, A.-H., Tan, C.-L., and Sun, S.-Y. On quantitative evaluation of clustering systems. In *Information Retrieval and Clustering*. Kluwer Academic Publishers, 2003, pp. 105–133.
14. Lloyd, S. P. Least square quantization in pcm. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137.
15. Loper, E., and Bird, S. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics - Volume 1* (Stroudsburg, PA, USA, 2002), ETMTNLP '02, Association for Computational Linguistics, pp. 63–70.

16. Marchionini, G. Exploratory search: From finding to understanding. *Communications of the ACM 49*, 4 (2006), 41–46.
17. Maulik, U., and Bandyopadhyay, S. Performance evaluation of some clustering algorithms and validity indices. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 12 (dec 2002), 1650 – 1654.
18. Mei, X. S., and Zhai, C. Automatic labeling of multinomial topic models. In *Proceedings of KDD 2007* (2007), pp. 490–499.
19. Newman, D., Karimi, S., and Cavedon, L. External evaluation of topic models. In *Proceedings of teh 14th Australasian Document Computing Symposum* (2009), pp. 11–18.
20. Newman, D., Noh, Y., Talley, E., Karimi, S., and Baldwin, T. Evaluating topic models for digital libraries. In *JCDL 2010* (2010).
21. Pirolli, P. Powers of 10: Modeling complex information-seeking systems at multiple scales. *Computer 42*, 3 (2009), 33–40.
22. Rao, R., Pedersen, J. O., Hearst, M. A., Mackinlay, J. D., Card, S. K., Masinter, L., Halvorsen, P.-K., and Robertson, G. C. Rich interaction in the digital library. *Commun. ACM 38*, 4 (Apr. 1995), 29–39.
23. Řehůřek, R., and Sojka, P. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (Valletta, Malta, May 2010), ELRA, pp. 45–50. http://is.muni.cz/publication/884893/en.
24. Roussinov, D. G., and Chen, H. Document clustering for electronic meetings: an experimental comparison of two techniques. *Decision Support Systems 27*, 1–2 (1999), 67–79.
25. Sculley, D. Web-scale k-means clustering. In *WWW 2010* (2010).
26. Song, M. Bibliomapper: a cluster-based information visualization technique. In *Information Visualization, 1998. Proceedings* (1998), pp. 130–136.
27. Sutcliffe, A., and Ennis, M. Towards a cognitive theory of information retrieval. *Interacting with Computers 10* (1998), 321–351.
28. van Ossenbruggen, J., Amin, A., Hardman, L., Hildebrand, M., van Assem, M., Omelayenko, B., Schreiber, G., Tordai, A., de Boer, V., Wielinga, B., Wielemaker, J., de Niet, M., Taekema, J., van Orsouw, M.-F., and Teesing, A. Searching and annotating virtual heritage collections with semantic-web technologies. In *Museums and the Web 2007* (2007).
29. Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. Evaluation methods for topic models. In *Proceedings of the 26th International Confer- ence on Machine Learning* (2009).
30. Wei, X., and Croft, W. B. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference* (New York, NY, USA, 2006), SIGIR '06, ACM, pp. 178–185.
31. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z.-H., Steinbach, M., Hand, D., and Steinberg, D. Top 10 algorithms in data mining. *Knowledge and Information Systems 14* (2008), 1–37. 10.1007/s10115-007-0114-2.
32. Zhao, W., Ma, H., and He, Q. Parallel k-means clustering based on mapreduce. In *Cloud Computing*, M. Jaatun, G. Zhao, and C. Rong, Eds., vol. 5931 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 674–679. 10.1007978-3-642-10665-1_71.