

# Using Novel IR Measures to Learn Optimal Cluster Structures for Web Information Retrieval

Martin Mehlitz  
 Technical University Berlin  
 DAI-Labor  
 10587 Berlin, Germany  
 martin.mehlitz@dai-labor.de

Jérôme Kunegis  
 Technical University Berlin  
 DAI-Labor  
 10587 Berlin, Germany  
 kunegis@dai-labor.de

Şahin Albayrak  
 Technical University Berlin  
 DAI-Labor  
 10587 Berlin, Germany  
 sahin.albayrak@dai-labor.de

**Abstract**—The Internet is a vast resource of information. Unfortunately, finding and accessing this information is often a very cumbersome task even with existing information platforms. Searching on the WWW suffers from the fact that almost every word is ambiguous to a certain degree in the information-rich environment of the Internet. Clustering search results is a way to solve this problem. This paper demonstrates how to employ novel Information Retrieval measures to derive optimal parametrizations for a cluster algorithm.

## I. INTRODUCTION

In the context of information retrieval (IR), clustering was introduced as a technique to improve retrieval results and the *Cluster Hypothesis* was put forth, stating that documents relevant to a query tend to form clusters [1], [2]. Especially in the area of web IR the amount of available data is so large that often the results of searches yield very low quality. Clustering retrieval results can therefore be regarded as a tool for visualizing and accessing retrieved documents.

A variety of approaches have been introduced for clustering text documents but a major problem of many algorithms is that they require a parametrization that suits the scenario in which they shall be employed. Often though, it is a very difficult task to automatize the process of determining a good parametrization. The DisCover algorithm [3] for clustering web search results, for instance, can produce cluster hierarchies of almost arbitrary depth and width and the task of finding a suitable size for it is not possible with common quality measures: The common measure for evaluating the IR performance of such a cluster structure task is the *optimal cluster search* which selects the cluster with the optimal value with regard to the employed measure [1] or the F-measure for cluster approaches (which employes the optimal cluster search)[4], [5], [6], [7]. These methodologies are popular because they eliminate distortion of evaluation results due to factors such as whether the visualization of the cluster structure is intuitive or not. However, if we want to find a good parametrization for the DisCover algorithm these measures can not be employed because they are monotonously increasing in the number of clusters that a hierarchy contains. Hence they would encourage a large but probably very impractical cluster structure.

In this paper we will employ novel IR quality measures [8] to demonstrate how to obtain a parametrization for a cluster algorithm.

The remainder of this paper is structured as follows: We will briefly summarize common evaluation approaches in IR and in cluster based retrieval. Then, we shortly introduce the novel IR measures and how to apply them to the optimal cluster search methodology. Finally, we demonstrate the usefulness of this method by an experiment with the DisCover algorithm in which we determine optimal cluster hierarchy sizes. A conclusion and an outlook to future work will end this contribution.

## II. BACKGROUND AND RELATED WORK

For a collection  $C$  of documents and a query  $q$ , let  $C_{q,n}$  be a sequence of  $n$  top ranking documents  $d_1, d_2, \dots, d_n$  which are returned by an IR system. Evaluating such a retrieval requires relevance judgements  $\mathcal{J}_q$  of the form

$$\mathcal{J}_q : C \rightarrow \{0, 1\}, \quad (1)$$

stating whether a document is relevant to the query or not. For most test collections the relevance judgments are not complete and usually documents that have not been judged are assumed to be irrelevant.

Evaluation measures for IR System are usually functions

$$\mathcal{M} : \text{Sym}(\mathcal{P}(C)) \times \mathcal{J}_q \rightarrow \mathbb{R}. \quad (2)$$

where  $\text{Sym}(\mathcal{P}(C))$  denotes the set of all permutations of all possible subsets of  $C$ . The measures typically employed to evaluate an IR system are *precision* and *recall*. *Precision* (“How many of the retrieved results are relevant?”) and *recall* (“How many of the relevant results are retrieved?”) are defined as:

$$\mathcal{M}_{\text{Prec}_n}(C_{q,n}, \mathcal{J}_q) = \frac{\text{rel}(C_{q,n})}{n} \quad (3)$$

and

$$\mathcal{M}_{\text{Rec}_n}(C_{q,n}, C, \mathcal{J}_q) = \frac{\text{rel}(C_{q,n})}{\text{rel}(C)}, \quad (4)$$

with

$$\text{rel}(D) = \sum_{d \in D} \mathcal{J}_q(d). \quad (5)$$

The F-measure is the harmonic mean of the precision and recall values:

$$\mathcal{M}_{\text{F}_n}(C_{q,n}, C, \mathcal{J}_q) = 2 \frac{\mathcal{M}_{\text{Prec}_n} \mathcal{M}_{\text{Rec}_n}}{\mathcal{M}_{\text{Prec}_n} + \mathcal{M}_{\text{Rec}_n}}. \quad (6)$$

Often precision and recall are measured at different cut-off values, for instance the precision at the first 10 documents:

$$\mathcal{M}_{\text{Prec}_x}(C_{q,n}, \mathcal{J}_q) = \frac{\text{rel}(C_{q,x})}{x} \quad (7)$$

A common methodology is also to measure and average the values for precision or recall at several fixed cut-off values. Since we will deal with document clusters of arbitrary size we will measure a clusters precision  $\mathcal{M}_{\text{CP}}$  as:

$$\mathcal{M}_{\text{CP}}(C_{q,n}, \mathcal{J}_q) = \frac{1}{n} \sum_{i=0}^n \mathcal{M}_{\text{Prec}_i}(C_{q,n}, \mathcal{J}_q) \quad (8)$$

Values for recall and F-Measure of a cluster can be derived analogously.

An evaluation of evaluation measure stability [9] showed that for most variants of the *precision* and *recall* measures, like mean average precision or precision at a certain document cut-off value, a number of 50 queries can ensure an sufficiently low error rate when comparing IR systems with these measures.

Clustering is the process of grouping items such that items within a group are similar to each other and, simultaneously, dissimilar to the items in other groups [10].

Text clustering for information retrieval can be done either statically on the whole collection  $C$  of documents [1] or in a query-specific manner for  $C_{q,n}$  a small number of top ranking documents [11], [12]. Experiments regarding the effectiveness of text clustering for IR [11], [12] concluded that the latter approach has the potential of improving IR.

A query specific cluster algorithm can be seen as a mapping of the form  $\mathcal{A} : C_q \rightarrow \mathcal{S}$  with  $\mathcal{S} \subset \text{Sym}(\mathcal{P}(C_q))$ . Note that this definition does not account for the hierarchical structure that some algorithms produce, however, for the automatic evaluation process we are concerned with, this definition is sufficient. Note also that this definition allows documents to be members of more than one cluster, which is in line with the notion that forcing documents to appear in only one cluster is artificial [12], [13] since documents can contain more than one topic and a good cluster structure should reflect this.

Various techniques for clustering have been proposed. Examples include suffix tree clustering [12], semi-supervised spectral clustering [14], formal concept analysis [11], or non-negative matrix factorization [15].

Usually, novel document clustering methods for IR are compared to existing methods or to an inverted file search, based on *recall* and *precision*. A common evaluation methodology for such a comparison is the “optimal cluster search” [13], [1], [16], [12]. This methodology evaluates every single cluster and then selects the cluster with the optimal value as a representative for the cluster hierarchy, so the optimal cluster search value is the function

$$\text{ocs}_q(\mathcal{A}, C_n, \mathcal{M}, \mathcal{J}_q) = \max_{s \in \mathcal{A}(C_n)} (\mathcal{M}(s, \mathcal{J}_q)) \quad (9)$$

For cluster algorithms, often a variation of the F-measure is reported if the algorithm is evaluated on a set of documents

with class labels. This measure  $F_{\text{cluster}}$  is computed in an IR fashion for each such class by using the optimal cluster search with the regular F-measure, treating all documents from the regarded class as relevant and all documents of any other class as irrelevant. The  $F_{\text{cluster}}$  measure is then the average over all classes.

### III. EMPLOYING NEW MEASURE FOR THE OPTIMAL CLUSTER SEARCH

Comparing algorithms with the optimal cluster search usually means to compute the *ocs* values (see (9)) for several queries and then make a statement of the form: “Algorithm  $\mathcal{A}_1$  is better than algorithm  $\mathcal{A}_2$  because the *ocs* values for the former one are higher than the values for the other”.

Naturally, a cluster algorithm will not produce all the  $2^n$  possible clusters in a scenario with a collection of  $n$  documents but only a relatively small number of clusters. However, it is interesting to note that, if we supposed a cluster algorithm to indeed produce every single possible cluster, it would achieve the highest possible value for the *ocs* score. Therefore, it is important to consider whether the *ocs* values of a certain cluster algorithm can be attributed to the algorithm itself or only to the rationale of choosing the best cluster to generate this value. In the following we will show to which extend the *ocs* values are attributable to the evaluation methodology. We will derive a value that captures this notion, called  $b_{\text{min}}$ , the *minimum baseline* for the evaluation of a cluster algorithm.

In [8] we introduced novel measures, the *absolute precision* and the *absolute recall*, for the performance of an IR system. These measure are based on the minimum baseline and are defined as:

$$\mathcal{M}_{\text{AP}_n}(C_{q,n}, \mathcal{J}_q) = \frac{\text{rel}(C_{q,n})}{n} - b_{\text{min}} \quad (10)$$

and

$$\mathcal{M}_{\text{AR}_n}(C_{q,n}, C, \mathcal{J}_q) = \frac{\text{rel}(C_{q,n})}{\text{rel}(C)} - b_{\text{min}} \quad (11)$$

Integrating this into the *cluster precision* measure that we defined above, we get the *absolute cluster precision* as:

$$\mathcal{M}_{\text{ACP}}(C_{q,n}, \mathcal{J}_q) = \frac{1}{n} \sum_{i=0}^n \mathcal{M}_{\text{Prec}_i}(C_{q,n}, \mathcal{J}_q) - b_{\text{min}} \quad (12)$$

In the following we will review the mathematical background for computing the expectation of random clusterings as a *minimum baseline* for the optimal cluster search. This expectation value can indicate whether the outcome of an experiment was likely to happen by chance or not.

First, we explain how to compute the value for a random pick of pages from a set of documents. Then we address how a growing number of random picks influences this value.

#### A. A Single Random Pick

The probability distribution for the random pick scenario is a hypergeometric distribution: Given a finite set of items (documents) where some are good (relevant) and some are bad (irrelevant), we draw without replacement. Suppose we

have a set of items consisting of  $g$  good items and  $b$  bad items. If we draw exactly  $d$  items, the probability for drawing a number  $i$  of good items is given by:

$$P(x = i) = \frac{\binom{g}{i} \binom{b}{d-i}}{\binom{g+b}{d}}. \quad (13)$$

The corresponding expected value for this probability amounts to:

$$\mu(d) = \sum_{i=0}^d \left( i \frac{\binom{g}{i} \binom{b}{d-i}}{\binom{g+b}{d}} \right) = \frac{g}{g+b} d. \quad (14)$$

Writing this expectation as a function of  $d$ , the total number of items drawn from the set of available items, shall emphasize that, if  $g$  and  $b$  are fixed, it only depends on how many items are drawn.

### B. Multiple Random Picks

The next step is to analyze to which extent the methodology of first generating multiple clusters and then selecting the best one changes the expected value of good items. Therefore, we need to express the likelihood of an outcome of a certain number of occurrences of relevant pages with regard to the number and size of clusters. Suppose we create  $c$  clusters where each cluster contains  $d$  items. Since the optimal cluster methodology selects the best cluster from a set of clusters, we need to describe the probability that a number of  $i$  relevant pages is the maximum number of relevant pages in the set of  $c$  clusters and that  $i$  really occurs. It is given by:

$$P(x = i, c, d) = \frac{B(x = i, d, c)}{A(d, c)}, \quad (15)$$

where  $A$  is the total number of different outcomes of choosing  $c$  times the number of  $d$  items and where  $B$  is the number of different ways of choosing these items with  $i$  being the maximum number of good items in the set of  $c$  clusters and with  $i$  really occurring.  $A$  is given by:

$$A(d, c) = \binom{g+b}{d}^c, \quad (16)$$

while  $B$  is given by:

$$B(x = i, d, c) = \sum_{j=1}^c \binom{c}{j} C(i, d)^j D(i, d)^{c-j}, \quad (17)$$

where  $j$  denotes the number of clusters that actually contain  $i$  good items. The first factor in the sum denotes the number of ways of distributing these  $j$  good items over the clusters  $c$ .  $C$  is the number of ways of drawing the  $i$  good items, while  $D$  counts the number of ways of drawing less than  $i$  good items.  $C$  and  $D$  are given by:

$$C(i, d) = \binom{g}{i} \binom{b}{d-i} \quad (18)$$

and

$$D(i, d) = \sum_{k=0}^{i-1} \binom{g}{k} \binom{b}{d-k} \quad (19)$$

respectively.

The expectation for multiple random draws is given by:

$$\mu_{all}(d, c) = \sum_{x=0}^d x P(x, c, d) \quad (20)$$

which only depends on the number of clusters and the number of pages in each cluster.

## IV. LEARNING CLUSTER ALGORITHM PARAMETRIZATION

In this section we will first introduce DisCover, a simple cluster algorithm. Then we will discuss the problem of how to parametrize this algorithm and finally we will demonstrate how to derive a good parametrization by employing the introduced measure  $\mathcal{M}_{ACP}$ .

### A. DisCover Algorithm

The DisCover algorithm [3] was designed as a fast algorithm for query-based clustering of web retrieval results. DisCover generates a hierarchy of clusters as follows:

Let  $\mathcal{N}_x$  be any node in a hierarchy of clusters (this could also be an empty root node) and let  $\mathcal{N}_{x,0}, \dots, \mathcal{N}_{x,b}$  be the direct children of this node. Let further  $C_{\mathcal{N}}$  denote the set of documents in  $\mathcal{N}$ .

Suppose every document  $d_i$  in a set of documents  $C_{q,n}$  can be represented by a vector of features  $\vec{f}_i$  with  $\vec{f}_i^T = (f_{i1}, f_{i2}, \dots, f_{im})$ . These features can be terms or phrases, references to other documents, the documents' author and so on. Let  $\mathcal{F}(\mathcal{N})$  be the set of all distinct features in  $\mathcal{N}$  and let  $\text{cover}(f)$  be the set of document with

$$\text{cover}(f) = \left\{ d_i \in C_{q,n} \mid f \in \vec{f}_i \right\}_{i \in 1, \dots, n}. \quad (21)$$

Given a cluster  $\mathcal{N}_x$  in the hierarchy, the DisCover algorithm produces child nodes of the form

$$\mathcal{N}_{x,i} = (f, \text{cover}(f) \cap C_{\mathcal{N}_x}). \quad (22)$$

where the feature  $f \in \mathcal{F}(\mathcal{N}_x)$  is the cluster description and  $C_{\mathcal{N}_{x,i}} = \text{cover}(f) \cap C_{\mathcal{N}_x}$  are those documents of the parent node which contain this feature. The features that are chosen as:

$$f = \text{argmax}_{f \in \mathcal{F}(\mathcal{N}_x)} (\text{discover}(f, \mathcal{N}_x)) \quad (23)$$

This feature scoring function is the following weighted sum:

$$\begin{aligned} \text{discover}(f, \mathcal{N}_x) &= \text{coverage}(f, \mathcal{N}_x) w_1 \\ &\quad + \text{diversity}(f, \mathcal{N}_x) w_2 \end{aligned} \quad (24)$$

where for the  $i$ -th child of  $\mathcal{N}_x$  the summands are:

$$\text{coverage}(f, \mathcal{N}_x) = |(C_{\mathcal{N}_{x,0}} \cup \dots \cup C_{\mathcal{N}_{x,i-1}}) \cap C_{\mathcal{N}_{x,i}}| \quad (25)$$

and

$$\text{diversity}(f, \mathcal{N}_x) = \frac{|(\mathcal{F}(C_{\mathcal{N}_{x,0}}) \cup \dots \cup \mathcal{F}(C_{\mathcal{N}_{x,i-1}})) \cap \mathcal{F}(C_{\mathcal{N}_{x,i}})|}{|\mathcal{F}(C_{\mathcal{N}_{x,i}})|} \quad (26)$$

So the diversity of a feature  $f$  in the context of a given node (and therefore in the context of previously generated

child nodes) is the number of features in the cover of  $f$  that do not appear in any other child node of the given node, while the coverage of the feature is the number of documents in the cover of  $f$  that do not appear in any other child node. The weights  $w_1$  and  $w_2$  quantify the relative importance of coverage and diversity.

One major advantage of the DisCover algorithm is its iterative nature. It is not necessary to produce the whole cluster structure or even all the children of a node at once. For instance, an application can cluster search results by treating the query as root cluster, containing all the returned documents. Then the first  $n$  children are produced. The user can now browse this cluster hierarchy by either looking *deeper* – forcing to generate the first  $n$  children of a particular node that caught her or his attention – or by looking *broadier* – forcing to produce the next  $n$  child nodes for the root node. In this way, there never needs to be more than  $n$  clusters produced at any time. This increases the response time for such an application, which is a very desirable characteristic for a web application.

### B. Parametrization of the DisCover Algorithm

One usability aspect of a web application is the response time. As we mentioned, the DisCover algorithm is suitable for clustering web search results, since in an interactive fashion, only the part of the cluster structure is created which really interests the user. Suppose a user has a very fast connection to the application server the DisCover methodology of generating only that part of the hierarchy that the user really explores can indeed speed up browsing the search results. On the other hand, if for every click in the hierarchy communication to the application server is needed, in the case of a slow connection the user will have to wait everytime he explores new parts of the hierarchy.

Since, in general, the DisCover algorithm is computational not very expensive, it is therefore reasonable to compute a larger part of the hierarchy at once and deliver the result to the user. Figure 1 shows a sample of a cluster structure in the PIA system<sup>1</sup> generated by the discover algorithm with a breadth of five and a depth of three.

What we want to derive now is a good size for a hierarchy. Meaning, how deep and how broad should an initially created hierarchy be. But how do we measure the quality of this structure? The common evaluation methodology to measure IR performance is the already discussed optimal cluster search strategy, but the value of this measure will increase monotonously with increasing size of the hierarchy, therefore we can not learn optimal cluster structures based on these values. However, if we employ the measure that we introduced, *the absolute cluster precision*, we can measure the quality of the cluster hierarchy in relation to its size – enabling us to derive the optimal size for the cluster structure.

## V. EXPERIMENTS

Since an evaluation of web search results is almost impossible to reproduce, because the WWW is constantly

Number of Documents	Optimal Depth	Optimal Breadth
100	2	6
200	2	6
300	3	6
400	3	7
500	3	7

TABLE I

OPTIMAL HIERARCHY SIZES FOR DIFFERENT NUMBERS OF CLUSTERED DOCUMENTS

changing, we will demonstrate how to derive an optimal parametrization for the DisCover algorithm on the collection of approximately 170,000 Wall Street Journal documents of the Tipster corpus<sup>2</sup>.

We use the TREC2 topics of the Ad Hoc Track to measure quality of cluster hierarchies. For these 50 topics  $t_{51}, t_{52}, \dots, t_{100}$ , the relevants judgments  $\mathcal{J}_{t_n}$  are known. For each topic, the topic's title is used to query a lucene [17] index over all the WSJ articles. Then, the top  $n$  ranking documents are retrieved and the 25 terms with the highest term-frequency (TF) values [2] are extracted for each document. These documents and features are the input for the DisCover algorithm.

For different values  $d$  and  $b$ , the depth and breadth respectively, we generate the corresponding cluster hierarchy and compute the *ocs* values with the  $\mathcal{M}_{ACP}$  measure. Figure 2 displays the results of this experiment, Table I shows the optimal values for each number of clustered documents. For smaller numbers of documents ( $n = 100, n = 200$ ) we observe that the performance of a hierarchy of depth  $d = 2$  and  $d = 3$  are not very far from each other. However, since we want to have a structure that is as small as possible, a hierarchy of depth  $d = 2$  is preferable.

Throughout all the experiments results we can see that a depth of  $d = 1$  is not enough to achieve good cluster results. Furthermore, we observe the tendency that the quality of the cluster hierarchy decreases after a certain breadth is reached for values of  $d = 2$  and  $d = 3$ . This is consistent with our expectation that very large cluster structures are not desirable. Note that this also means, that in general the DisCover algorithm produces an optimal cluster in a medium size hierarchy, so enlarging this structure is not necessary.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown how novel IR measures enable us to derive a parametrization for a cluster algorithm that we could not derive with traditional IR measures. Thereby, we increase the usability of an existing algorithm for web search result clustering. Furthermore, we have demonstrated how an evaluation methodology that has some theoretical flaws – the optimal cluster search – can be employed if the appropriate quality measures are used to measure optimality.

<sup>1</sup><http://pia.cs.tu-berlin.de/>

<sup>2</sup>The Wall Street Journal (WSJ) text collection is part of the Tipster corpus and can be obtained from <http://trec.nist.gov/>

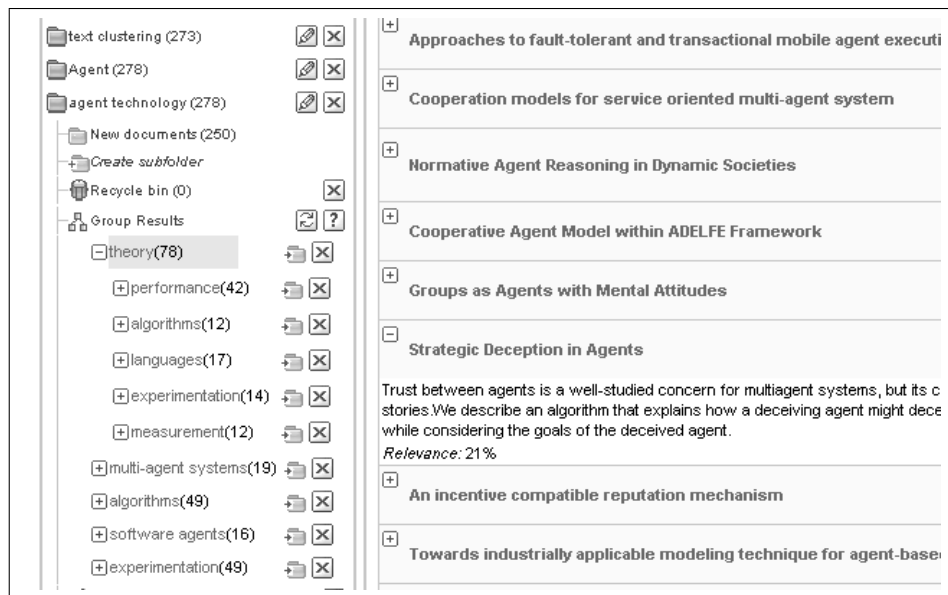


Fig. 1. A sample of document clustering for the query “agent technology” in the PIA System.

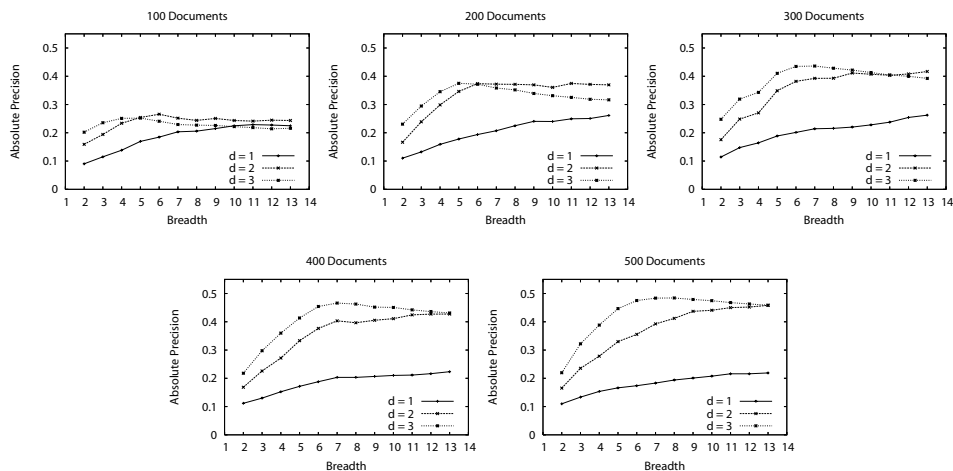


Fig. 2. Absolute optimal cluster search values for cluster structures at different numbers of clustered documents.

## REFERENCES

- [1] N. Jardine and C.J. Van Rijsbergen, “The use of hierarchic clustering in information retrieval,” *J. of Information Storage and Retrieval*, vol. 7, pp. 217–240, 1971.
- [2] G. Salton, *Automatic Information Organization and Retrieval*, McGraw Hill, New York, 1968.
- [3] K. Kumnamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram, “A hierarchical monothetic document clustering algorithm for summarization and browsing search results,” in *Proc. ACM Int. Conf. on World Wide Web*, 2004, pp. 658–665.
- [4] Bjornar Larsen and Chinatsu Aone, “Fast and effective text mining using linear-time document clustering,” in *Proc. ACM KDD*, 1999, pp. 16–22.
- [5] Ying Zhao and George Karypis, “Evaluation of hierarchical clustering algorithms for document datasets,” in *Proc ACM CIKM*, 2002, pp. 515–524.
- [6] Nachiketa Sahoo, Jamie Callan, Ramayya Krishnan, George Duncan, and Rema Padman, “Incremental hierarchical clustering of text documents,” in *Proc ACM CIKM*, 2006, pp. 357–366.
- [7] David Cheng, Ravi Kannan, Santosh Vempala, and Grant Wang, “A divide-and-merge methodology for clustering,” *ACM Trans. Database Syst.*, vol. 31, no. 4, pp. 1499–1525, 2006.
- [8] Martin Mehlitz, Jerome Kunegis, Christian Bauchhage, and Sahin Albayrak, “A new evaluation measure for information retrieval systems,” in *To appear in Proc. IEEE SMC*, 2007.
- [9] Chris Buckley and Ellen M. Voorhees, “Evaluating evaluation measure stability,” in *Proc. ACM SIGIR*, 2000, pp. 33–40.
- [10] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, 1988.
- [11] C. Carpineto and G. Romano, “Exploiting the potential of concept lattices for information retrieval with CREDO,” *J. of Universal Computer Science*, vol. 10, no. 8, pp. 985–1013, 2004.
- [12] O. Zamir and O. Etzioni, “Web document clustering: A feasibility demonstration,” in *Proc. of ACM SIGIR*, 1998, pp. 46–54.
- [13] M.A. Hearst and J.O. Pedersen, “Rexaming the cluster hypothesis: Scatter/gather on retrieval results,” in *Proc. ACM SIGIR*, 1996, pp. 76–84.
- [14] Xiang Ji and Wei Xu, “Document clustering with prior knowledge,” in *Proc. ACM SIGIR*, 2006, pp. 405–412.
- [15] Wei Xu, Xin Liu, and Yihong Gong, “Document clustering based on non-negative matrix factorization,” in *Proc. ACM SIGIR*, 2003, pp. 267–273.
- [16] H. Schutze and C. Silverstein, “Projections for efficient document clustering,” in *Proc. ACM SIGIR*, 1997, pp. 74–81.
- [17] Erik Hatcher and Otis Gospodnetic, *Lucene in Action (In Action Series)*, Manning Publications Co., 2004.