Author(s): Nurminen, Miika; Honkaranta, Anne; Kärkkäinen, Tommi

Title: ExtMiner : Combining multiple ranking and clustering algorithms for structured document retrieval

Year: 2005

Version:

Please cite the original version:

Nurminen, M., Honkaranta, A., & Kärkkäinen, T. (2005). ExtMiner : Combining multiple ranking and clustering algorithms for structured document retrieval. In Sixteenth International Workshop on Database and Expert Systems Applications (DEXA). 22-26 Aug. 2005, Copenhagen, Denmark. (pp. 1036-1040). IEEE Computer Society. Integrating Data Mining, Databases and Information Retrieval (IDDI) '05 Workshop.. https://doi.org/10.1109/DEXA.2005.91

ExtMiner: Combining Multiple Ranking and Clustering Algorithms for Structured Document Retrieval

Miika Nurminen, Anne Honkaranta, Tommi Kärkkäinen
Faculty of Information Technology
University of Jyväskylä, Finland
minurmin@cc.jyu.fi, anne.honkaranta@it.jyu.fi, tka@mit.jyu.fi

Abstract

This paper introduces ExtMiner, a platform and potential tool for information management in SMEs (Small& Medium-size Enterprise), or for organizational workgroups. ExtMiner supports interactive and iterative clustering of documents. It provides users with a visual cluster- and list views at the same time, supporting iterative search process. ExtMiner may also be applied as a platform for research on retrieval fusion, since it combines search, clustering and visualization algorithms. ExtMiner was evaluated with three document collections. Although the findings were encouraging the user interface and performance with large document repositories need further development.

1. Introduction

Organizations are nowadays provided with overwhelming amount of digital information which has a potential of supporting everyday business and knowledge work. As a side effect of this information overflow, new ways for retrieving, filtering, and managing information in organizations are needed. Search for information on digital repositories pose multiple challenges for information seekers: the retrieved sets of documents rarely match the information needs of people. For one thing, people find it difficult to express their information needs as index terms and keywords used for describing the document content. Same term may have differing connotations for human users, and users may not be able to select the "right keywords" on the number of potential ones.

Heterogeneous document collections cannot be sufficiently described, nor searched for when merely index terms are applied. It also seems that the XML language becomes a "lingua franca" as a format for organizational documents. Hence, enhancements for index-term-based document retrieval, especially for structured documents are required.

Text mining techniques – such as document classification and keyword-based association analysis – provide potential means for retrieving documents in large collections [8]. In this paper, we focus on text mining technique called as clustering, i.e. defining groups of documents with similar content features, which can be applied in SMEs and organizational departments with structured digital document repositories. Document clustering has been used in information retrieval for cluster-based search where the query is matched to a pre-clustered model [15], cluster-based browsing [4] and search results grouping [18].

In this paper, we introduce ExtMiner, a platform and a proof-of-concept for research for combining multiple ranking and clustering algorithms for structured document retrieval. ExtMiner integrates many of the features previously implemented in separate systems. It adopts extended vector model proposed by Fox *et al.* [7] and supports both cluster-based browsing and search results clustering. Either a flat or hierarchical cluster model may be produced. Extended vector model is based on the classical vector model developed by Salton *et al.* [13]. The vector model has become popular, especially among web search engines. Vector model is easy to implement and fast to use, but restricted by its ability to express document structures and links.

There are several document retrieval systems that have some features in common with ExtMiner. Scatter/Gather-system introduced a continuous searching model, where access to a document collection may be a narrowly specified search for a particular document, such as searching a document by its title, or an ad-hoc browsing session with no well defined goal at all or anything between these search tactics [4]. LightHouse is an interface system for a typical web search engine with tight integration between the ranked list and visualization of clusters [11]. Crouch *et al.* [3] have applied extended vector model for XML retrieval. Vivísimo (`http://vivisimo.com/`) is a contemporary metasearch engine supporting clustering of search results.

Retrieval fusion approach, also supported by ExtMiner, investigates various ways of combining different retrieval

strategies [18]. MSEEC search engine [9] presented an architecture for combining multiple clustering algorithms. A retrieval system by Ben-Aharon *et al.* [1] combined various rankers for content- and structure-based XML search. CiteSeer (`http://citeseer.ist.psu.edu/`) uses both text and citation information for searching and similarity assessment for scientific publications on the web.

The paper is organized as follows. In section 2 we introduce the overall structure of the ExtMiner application. Section 3 describes the findings of evaluating ExtMiner with three different document collections. Section 4 concludes the paper and discusses use scenarios for ExtMiner.

2. ExtMiner

ExtMiner uses one index file as a base for document clustering as well as for document retrieval. The index may contain information of textual content, links, structure and metadata of documents in the collection. When documents are searched, the documents' index values are matched against the retrieval index terms given by the user. Documents with term similarities are clustered together.

ExtMiner is able to process multiple kinds of documents, such as text, PDF, and XML documents. Documents in the collection, however, should adhere to the same schema or at least they should be transformable to a comparable representation with same content fields. For example, if the majority of the source documents are on text format, the remaining PDF and XML documents should be transformed to text before including them in the document collection.

2.1. Architecture

The underlying objective of ExtMiner is to provide flexible and modular platform for document clustering and retrieval. We emphasized two properties for the flexibility of the application platform. The first is the ability to process collections having documents of differing types, such as plain-text, semi-structured, and structured documents, as well as Adobe PDF, and the second one is the flexibility provided by the easy plug-in and switching of clustering, visualization, and indexing algorithms on top of the platform.

For modularity, the ExtMiner architecture is divided into three layers: the user interface, the application logic, and the document index. Hence, enhancements on the user interface configuration do not interfer the functionality of the two other layers. Figure 1 illustrates the components of ExtMiner architecture as rectangles.

ExtMiner has been programmed with the Java language, and licensed as an open source application. The application consists of both self-developed as well as open source software components.

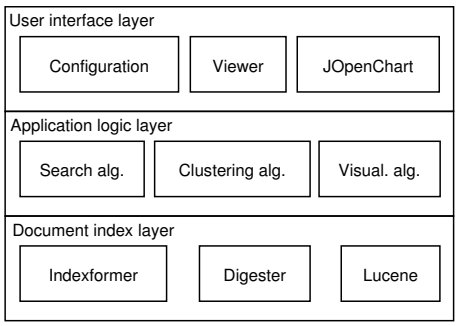

Figure 1. ExtMiner architecture.

The open source components used in ExtMiner include the following. The document retrieval is based on the **Lucene** (`http://lucene.apache.org/`) search engine, developed by the Jakarta project. Lucene utilizes a field-based index structure and supports $tf \times idf$-style term weighting scheme. ExtMiner retrieves most of the information needed in document clustering from Lucene. **Digester** (`http://jakarta.apache.org/commons/digester/`) parses the XML documents. Graphical grid view is based on the **JOpenChart** (`http://jopenchart.sf.net/`) visualization component by S. Müller.

The self-developed components of ExtMiner are the following. **Search algorithm** stores the search query given by the user, and returns a list of matched documents organized by their estimated relevance. **Indexformer** takes care of indexing new documents while they are added to the collection. **Clustering algorithm** takes a list of documents and provides either a flat or hierarchical cluster model of them. **Configuration** defines a filename filter for documents to be indexed, and Java class names for Indexformer and Viewer. **Viewer** is responsible for opening the full-text document in a custom user interface. **Visualization algorithm** projects selected documents into a two-dimensional space.

2.2. Integrated Model for Searching and Clustering

The retrieval model used by ExtMiner integrates ranked lists and cluster-based browsing and searching. Moreover, the results of the individual search and clustering algorithms can be assembled from several components. The general framework for combining document features is the extended vector model [7]. The similarity measure or retrieval function is assembled from various feature vectors. ExtMiner facilitates document retrieval by incorporating an iterative search and clustering process, interactive cluster model and simultaneous views for lists and clusters (see figure 2).

Iterative search and clustering process. Term-based search and document clustering are considered as primitive operations that can be performed iteratively and focused to

an appropriate subset of the collection. When the system is started the user is provided with the global cluster model. The search can be directed to a single cluster. Alternatively a new model can be computed based on search results or any collection subset selected by the user.

Interactive cluster model. The user can select the documents using any of the views provided by the application: ranked lists, cluster tree, or a visual projection representing the document collection. The user is presented with a condensed representation of the document including its most highly weighted keywords and metadata fields. Cluster tree is interactive: a cluster can be marked as "noise", and subclusters of a single cluster can be merged.

Simultaneous view for lists and clusters. Ranked lists and clusters support different search objectives. The user has the most control and flexibility to perform the search, since both views are available. If only ranked lists were used, relevant documents might be scattered around the list because of the ambiguous terms, synonyms or imprecisely formulated queries. Users prefer cluster-based interface to a list interface for the interactive retrieval, although the clusters do not seem to improve the search quality as such [17].

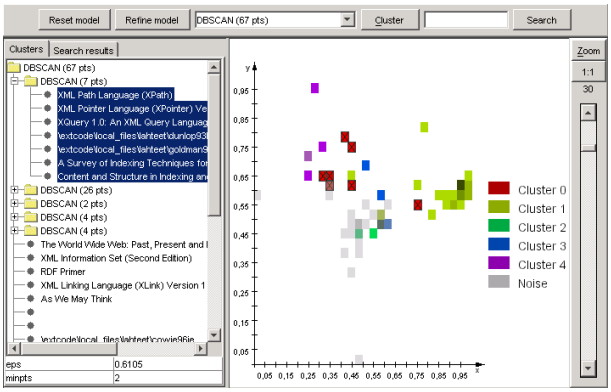

Figure 2. ExtMiner user interface.

2.3. Indexing and Searching

A collection must be downloaded to the local file system for indexing. Indexing is started from a user-specified root directory and continued recursively to subdirectories. The directory path can be interpreted as a web site and converted to a URL for online document viewing. Stopword removal and stemming is performed by Lucene. Indexformer component is responsible for parsing the documents, and delivering the index data to Lucene and the internal indices, such as the similarity matrix. When indexing XML documents, parsing rules provided by Digester can be used. If another document format, such as text files is used, a custom Indexformer must be implemented.

The internal representation of documents and queries is based on a field-based index. Index terms, links, metadata and other custom elements such as headers are represented as vectors and indexed as fields using Lucene. $tf \times idf$-weighting is applied on index terms. Each document collection is represented by a distinctive set of fields that must be determined during the indexing stage. The fields can vary based on document characteristics. For example, a text document has no links or custom elements at all, but may be extended by a rich set of external metadata fields, if the document is originally stored in a document management system.

Let $d \in \mathbf{D}$ be a document and q a query, both represented as tuples of n vectors. The relevance estimate R is computed as follows: $R(d,q) = \sum_{k=1}^{n} w_k sim_k(r_k(d), r_k(q))$, where r_k denotes the restriction that extracts the k-th vector from the vector tuples. sim_k is a similarity measure or retrieval function with values scaled to $[0,1]$. For term-based search, similarity is computed using normalized inner vector product (e.g. cosine measure). PageRank [2] could provide a general link-based relevance estimate. Vector w consists of field weights such that the component sum is 1. The weights are evenly matched or manually specified by the user [18].

2.4. Clustering Algorithms

Vector model is the contemporary way of representing documents for a clustering algorithm. However, structured documents can not be naturally represented in vector space, because of their tree- or graph-like data model. ExtMiner applies metric clustering, because it is independent of document representations, requiring only document-to-document similarities as an input. The similarities can be computed using different domain-dependent measures, such as the cosine measure for text-based similarity, or co-citation and bibliographic coupling [10] for link-based similarity. Advanced structural matching schemes, such as the s-term model proposed by Schlieder & Meuss [14] could also be used, if a path or tree-based index is available. After the computation of similarity (or distance) matrix, any metric clustering algorithm can be used.

DBSCAN [5] was selected as the primary clustering algorithm used by ExtMiner, because it produces a flat clustering model that is more convenient for end users compared to hierarchical clustering. Also, many density-based clustering algorithms are based on DBSCAN so it can be used as a basis for further development. The clustering quality was adequate, but the ϵ and $minpts$ parameters required manual adjustments. Parameters were selected using an approximate manual scan. Group average hierarchical clustering was selected as the second clustering algorithm. The advantage of the method is that it requires no input parame-

ters and produces clusters of superior quality compared to the minimum link method [16]. Both algorithms are known to be inefficient for clustering large document collections. This can be alleviated by incorporating new clustering algorithms to the system, depending on the collection.

2.5. User Interface

ExtMiner provides both ranked lists and the cluster model for browsing and searching the document collection. Individual documents, clusters or any subset of the collection can be selected. A new cluster model or a refined search can be based on selected documents. If a single document is selected, the system highlights its index terms, links and any custom fields that are assigned to the document during the indexing stage. Documents can be opened using a custom viewer, such as a standard web browser or a XML tree view. In addition to ranked list and cluster tree, the user interface features a visual view of the document collection.

The FastMap [6] projection algorithm produces the visual view for ExtMiner. The algorithm maps data from metric space to vector space in linear time such that document-to-document similarities are preserved, thus implementing multidimensional scaling. ExtMiner uses the implementation provided by XXL-library (`http://dbs.mathematik.uni-marburg.de/research/projects/xxl/`). The view facilitates the interpretation of clusters and search results. The granularity of the grid can be scaled and the view is zoomable. The visual model is computed using the same similarity matrix as in clustering. Document clusters are illustrated with different colors, but visualization itself is independent of search and clustering. Figure 2 shows an example of the visualization.

3. Evaluation

The functionality of ExtMiner was evaluated using three document collections: a collection of over 1000 course essays (The essay collection), a collection of 300 HTML-documents of a paper technology learning site (The KnowPap collection), and a collection of references (The thesis reference document collection) from one of the authors' thesis work [12], consisting of 145 source documents. The evaluation was carried out by five persons; three course lecturers, a contact person from one of the paper technology learning site developer organization, and one of the authors of this paper.

Introduction to Software Engineering course was carried out in Fall 2004 in the Faculty of Information technology at the University of Jyväskylä. As part of the course, each student was assigned to produce 13 essays, one for each lecture. Over 200 students signed up to the course, and the amount of essays rose up to over 1000 at the end of the course. Managing the collection of essays manually would have been an overwhelming task for the lecturers.

ExtMiner was utilized for checking up and comparing the essays written by students. Each essay was stored as text. The collection consisted of an index of metadata values about the students and essays, such as students' names and majoring studies. The name of each essay was retrieved automatically from the first text line of each document. As the collection was indexed, the lecturer could retrieve essays from the collection by using each of the metadata values as a search key. The essay collection was also clustered according to subject-matter terms, which allowed the lecturers to cross-inspect each cluster of essays pertaining to certain lecture or subject matter in relation to each other.

KnowPap is an e-Learning application for paper production technologies. It contains a collection of HTML-documents, pictures, video clips, and other material, used for paper technology education. For evaluating ExtMiner, we imported a portion of 300 documents from KnowPap e-Learning web site. In addition to indexing metadata and content of HTML documents the links to multimedia files were also indexed. As a result, we could use ExtMiner as a proof-of-concept for a multimedia database, allowing document and multimedia retrieval, and subject-based clustering. Paper technology trainers may use the ExtMiner as a tool for organizing, browsing, and retrieving training material components for novel training content.

One author of this paper also evaluated ExtMiner with the **collection of references** for his thesis. The collection of references consisted of 145 HTML and PDF documents. To form a homogeneous collection of well-formed HTML documents, the source documents were pre-processed with Pdftohtml and HTML Tidy applications. Pre-processing also consisted of elimination of stop-words and stemming of terms. The reference documents that were not able to pass the automatic pre-processing were excluded. The remaining reference document collection consisted of 69 documents and 16000 terms. The collection was clustered with DBSCAN algorithm and the Group average method. The DBSCAN parameters were adjusted by manual scanning through parameter values, and the numerous clusters produced by the hierarchical clustering algorithm were manually combined into larger document clusters.

According to DBSCAN clusters, the reference documents for the thesis consisted of five main subject areas: **The XML cluster** consisted of references dealing with XML language in overall, **The main cluster** consisted of a dearth of articles and theses focused mainly on clustering, **The LSI cluster** contained the references dealing with Latent Semantic Indexing method, **The data mining cluster** contained articles on data mining, and **The XML indexing cluster** contained articles related to retrieval of XML documents. In addition, the DBSCAN clustering algorithm

classified 24 references as noise.

Group average method produced two new clusters and dropped out one compared to DBSCAN clustering. The two new clusters were **The link cluster**, containing mainly articles on link analysis. The second new cluster consisted of **general articles** on subject area, which DBSCAN had classified as noise. There was no cluster corresponding with the LSI cluster in the DBSCAN model. The XML indexing cluster contained some documents that were ranked into the main XML cluster by the DBSCAN algorithm.

4. Conclusions and Future Work

ExtMiner supports iterative and interactive search of documents within organizational collections. It may be used for text, PDF, or XML documents, given that input documents are transformed to a comparable representation while indexing. As an enhancement for index-based document retrieval, ExtMiner adopts the extended vector model and document clustering as additional means for document search. ExtMiner shows potential to become a tool for information management in SMEs, or for organizational workgroups. It may also be used as a flexible platform for further document retrieval research.

ExtMiner was evaluated with three document collections. Although the findings were encouraging, the user interface needs further development. The essay collection showed that performance of ExtMiner with large repositories is not efficient enough using current clustering algorithms. A usability test on user interface should be conducted with a group of users with varying computing skills. Use of ExtMiner requires manual work for preprocessing heterogeneous source documents. The system should be enhanced with validation functionality for evaluating search and clustering quality using standard test collections.

The parameters for DBSCAN are adjusted manually. The algorithm classified many of the documents in test collections as noise or in one large cluster. With hierarchical clustering, use of cluster model directly in ExtMiner user interface was found somewhat laborious, because the visual view distinguishes all clusters and subclusters as separate entities. The model becomes useful only after manually merging most of the subclusters. The merged clusters appear to be of comparable or superior quality compared to DBSCAN clustering. Automatic tree pruning and parameter estimation may improve the usability of ExtMiner with large document collections.

References

[1] Y. Ben-Aharon, S. Cohen, Y. Grumbach, Y. Kanza, J. Mamou, Y. Sagiv, B. Sznajder, and E. Twito. Searching in an XML corpus using content and structure. In *INEX 2003 Workshop Proceedings*, pages 46–52. INEX, 2003.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In P. H. Enslow, Jr. and A. Ellis, editors, *Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117. Elsevier, 1998.

[3] C. Crouch, S. Apte, and H. Bapat. An approach to structured retrieval based on the extended vector model. In *INEX 2003 Workshop Proceedings*, pages 89–93. INEX, 2003.

[4] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: a cluster-based approach to browsing large document collections. In *Proc. of the 15th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 318–329. ACM Press, 1992.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231. AAAI Press, 1996.

[6] C. Faloutsos and K.-I. Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *SIGMOD Rec.*, 24(2):163–174, 1995.

[7] E. A. Fox, G. L. Nunn, and W. C. Lee. Coefficients of combining concept classes in a collection. In *Proc. of the 11th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 291–307. ACM Press, 1988.

[8] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[9] P. Hannappel, R. Klapsing, A. Krug, and G. Neumann. MSEEC - a multi search engine with multiple clustering. In *Proc. of the 10th Information Resources Management Association Int. Conf.* Idea Group Publishing, 1999.

[10] T. R. Kochtanek. Bibliographic compilation using reference and citation links. *Information Processing & Management*, 18(1):33–39, 1982.

[11] A. Leuski and J. Allan. Lighthouse: Showing the way to relevant information. In *INFOVIS '00: Proceedings of the IEEE Symposium on Information Vizualization 2000*, pages 125–129. IEEE Computer Society, 2000.

[12] M. Nurminen. Tiedonlouhinta rakenteisista dokumenteista *(in Finnish)*. Master's thesis, University of Jyväskylä, 2005.

[13] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[14] T. Schlieder and H. Meuss. Querying and ranking XML documents. *Journal of the American Society for Information Science and Technology*, 53(6):489–503, 2002.

[15] C. J. van Rijsbergen. *Information Retrieval*. London: Butterworths, 1979.

[16] P. Willett. Recent trends in hierarchic document clustering: A critical review. *Information Processing & Management*, 24(5):577–597, 1988.

[17] M. Wu, M. Fuller, and R. Wilkinson. Using clustering and classification approaches in interactive retrieval. *Information Processing & Management*, 37(3):459–484, 2001.

[18] K. Yang. *Combining Text-, Link-, and Classification-based Retrieval Methods to Enhance Information Discovery on the Web*. PhD thesis, University of North Carolina, 2002.