Chapter 1 : The Text Mining Handbook : Ronen Feldman :

Text mining refers to text categorization, information extraction, and terms extraction. Mining such information from textual sources is an emerging research area, following the popularity of data mining that often refers to extracting specific parameters from numeric data.

Our capabilities of both generating and collecting data have been increasing rapidly in the last several decades. Contributing factors include the widespread use of bar codes for most commercial products, the computerization of many business, scientific and government transactions and managements, a Contributing factors include the widespread use of bar codes for most commercial products, the computerization of many business, scientific and government transactions and managements, and advances in data collection tools ranging from scanned texture and image platforms, to on-line instrumentation in manufacturing and shopping, and to satellite remote sensing systems. In addition, popular use of the World Wide Web as a global information system has flooded us with a tremendous amount of data and information. This explosive growth in stored data has generated an urgent need for new techniques and automated tools that can intelligently assist us in transforming the vast amounts of data into useful information and knowledge. This book explores the concepts and techniques of data mining, a promising and flourishing frontier in database systems and new database applications. Data mining, also popularly referred to as knowledge discovery in databases KDD , is the automated or convenient extraction of patterns representing knowledge implicitly stored in large databases, data warehouses, and other massive information repositories. Data mining is a multidisciplinary field, drawing work from areas including database technology, artificial intelligence, machine learning, neural networks, statistics, pattern recognition, knowledge based systems, knowledge acquisition, information retrieval, high performance computing, and data visualization. We present the material in Data Mining: Mining information and knowledge from large databases has been recognized by many researchers as a key research topic in database systems and machine learning, and by many industrial companies as an important area with an opportunity of major revenues. Researchers in many different fields have sh Researchers in many different fields have shown great interest in data mining. Several emerging applications in information providing services, such as data warehousing and on-line services over the Internet, also call for various data mining techniques to better understand user behavior, to improve the service provided, and to increase the business opportunities. A classification of the available data mining techniques is provided and a comparative study of such techniques is presented. Discovery of association rules is an important database mining problem. We present new algorithms that reduce the data We present new algorithms that reduce the database activity considerably. Theidea is to pick a random sample, to ndusingthis sample all association rules that probably hold in the whole database, and then to verify the results with the restofthe database. The algorithms thus produce exact association rules, not approximations based on a sample. The approach is, however, probabilistic, and inthose rare cases where our sampling method does not produce all association rules, the missing rules can be found inasecond pass. Our experiments show that the proposed algorithms can nd association rules very e ciently in only onedatabase pass. Cheung, Jiawei Han, Vincent T. Wong , " An incremental updating technique is developed for maintenance of the association rules discovered by database mining. There have been many studies on efficient discovery of association rules in large databases. However, it is nontrivial to maintain such discovered rules in large databases because a However, it is nontrivial to maintain such discovered rules in large databases because a database may allow frequent or occasional updates and such updates may not only invalidate some existing strong association rules but also turn some weak rules into strong ones. In this study, an incremental updating technique is proposed for efficient maintenance of discovered association rules when new transaction data are added to a transaction database. The emerging data mining tools and systems lead naturally to the demand of a powerful data mining query language, on top of which many interactive and flexible graphical user interfaces can be developed. This motivates us to design a data mining query language, DMQL, for mining different kinds of kn This motivates us to design a data mining query language, DMQL, for mining different kinds of knowledge in relational databases. Portions of the proposed DMQL language have been implemented in our DBMiner system for interactive mining of multiple-level knowledge in relational databases. Show Context Citation Context Since the tasks and applications of data mining are broad and diverse, it is expected that various kinds of exible, interactive user interfaces for data mining will emerge. We believe that the devel Abstractâ€"Spatial data mining is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. To this end, this paper has three main contributions. One of the methods considered, called the IR-approximation, is very efficient in clustering convex and nonconvex polygon objects. Third, building on top of CLARANS, we develop two spatial data mining algorithms that aim to discover relationships between spatial and nonspatial attributes. Both algorithms can discover knowledge that is difficult to find with existing spatial data mining algorithms. Index Termsâ€"Spatial data mining, clustering algorithms, randomized search, computational geometry. Fu, Yongjian Fu , " Many sequential algorithms have been proposed for mining of association rules. However, very little work has been done in mining association rules in distributed databases. A direct application of sequential algorithms to distributed databases is not effective, because it requires a large amount of A direct application of sequential algorithms to distributed databases is not effective, because it requires a large amount of communication overhead. In this study, an efficient algorithm, DMA, is proposed. It generates a small number of candidate sets and requires only O n messages for support count exchange for each candidate set, where n is the number of sites in a distributed database. The algorithm has been implemented on an experimental test bed and its performance is studied. The results show that DMA has superior performance when comparing with the direct application of a popular sequential algorithm in distributed databases. A lattice conceptual clustering system and its application to browsing retrieval by Claudio Carpineto, Giovanni Romano - Machine Learning , " The theory of concept or Galois lattices provides a simple and formal approach to conceptual clustering. The algorithm utilized by GALOIS to build a concept lattice is incremental and efficient, each update being done in time at most quadratic in the number of objects in the lattice. Also, the algorithm may incorporate background information into the lattice, and through clustering, extend the scope of the theory. The application we present is concerned with information retrieval via browsing, for which we argue that concept lattices may represent major support structures. We describe a prototype user interface for browsing through the concept lattice of a document-term relation, possibly enriched with a thesaurus of terms. An experimental evaluation of the system performed on a medium-sized bibliographic database shows good retrieval performance and a significant improvement after the introduction of background knowledge. Concept hierarchies organize data and concepts in hierarchical forms or in certain partial order, which helps expressing knowledge and data relationships in databases in concise, high level terms, and thus, plays an important role in knowledge discovery processes. Concept hierarchies could be prov Concept hierarchies could be provided by knowledge engineers, domain experts or users, or embedded in some data relations. However, it is sometimes desirable to automatically generate some concept hierarchies or adjust some given hierarchies for particular learning tasks. In this paper, the issues of dynamic generation and refinement of concept hierarchies are studied. The study leads to some algorithms for automatic generation of concept hierarchies for numerical attributes based on data distributions and for dynamic refinement of a given or generated concept hierarchy based on a learning request, the relevant set of data and database statistics. These algorithms have been implemented in the DBkearn knowledge discovery system and tested against large relational databases. The experimental results show that the algorithms are efficient and effective for knowledge discovery in large databases. There are different philosophical considerations on knowledge discovery in databases KDD [5, 23], which may lead to different methodologies in the development of KDD techniques [5, 11, 6, 18, 1, Hypothesis generation, a crucial initial step for making scientific discoveries, relies on prior knowledge, experience and intuition. Chance connections made between seemingly distinct subareas sometimes turn out to be fruitful. The goal in text mining is to assist in this process by automaticall

DOWNLOAD PDF PRESENTATION-LAYER CONSIDERATIONS FOR BROWSING AND QUERY REFINEMENT

The goal in text mining is to assist in this process by automatically discovering a small set of interesting hypotheses from a suitable text collection. But instead of mining a collection of well-structured data, text mining operates off text collections that are at best semi-structured. In both cases, the knowledge discovered is essent

Chapter 2 : Developing a Custom Search Engine for an Expert Chat System - Developer Blog

THE TEXT MINING HANDBOOK Presentation-Layer Considerations for Browsing and Query Refinement IX. 1 Browsing

In this post I will highlight some of the issues that need to be considered before doing the implementation so that you could use it as a reference for planning the Migration of Fast Search for SharePoint For more information please check the references section below. Search alerts By using search alerts in SharePoint Server , users can receive daily or weekly updates on all the new or changed documents for a specific query. FAST Search Server for SharePoint does not keep information about when a document was first indexed or when it was modified, and therefore does not provide this feature. However, similar alerting functionality can be achieved via partners or Microsoft Services. A possible approach would be to save the queries, run them as batches, and then use an external mechanism to track results that have or have not been presented before. Social tagging and its effect on relevance ranking Social tagging is provided in SharePoint Server enterprise search. The feature is available through information that is stored in and extracted from a separate user information database. This does not mean that social tagging is removed completely, only its effect on search results and the possibility to present or refine on social tags in search results. If the information still exists in the user database, it is possible to tag content and have other users see the tags. Social tagging and relevance ranking can then be implemented through custom code created by a partner, Microsoft Services, or on-site developers. Definition extraction SharePoint Server extracts meanings of definitions from indexed text. Definition extraction occurs during a crawl and is a process of associating terms with one or more descriptions through the recognition of cue phrases. Crawl rules for document formats SharePoint Server specifies the document formats to be crawled by defining a set of File Types to include in the content index. For more information, see Operations â€” Add a file type to the content index http: Instead of defining what to include, you specify the file types to be excluded from a crawl. For more information, see Operations â€” Include a file type in the content index http: Mirroring indexes across multiple data centers, backup and recovery functionality FAST Search Server for SharePoint does not offer mirroring and synchronization of indexes across multiple data centers. Index redundancy or index fault tolerance can be achieved by various ways of indexer backup deployments. In most large deployments, the index will be spread over several columns, and it is usual to have several rows with copies of the index for redundancy. In a fault-tolerant indexer setup of FAST Search Server for SharePoint, the data index is automatically copied to the backup indexers, which reside on the backup indexer row. If one indexer server fails with unrecoverable disk errors, you can either recover the server farm from the latest backup, or manually enable a backup indexer to act as the new primary indexer. For more information about redundancy and availability, see Planning and architecture - FAST Search Server farm redundancy and availability http: For more information about backup and recovery strategy, see Operations â€” Plan the system backup and recovery strategy http: This difference has the following effects on size footprint: Therefore, the number of servers and the configuration of the servers will be different. The total disk footprint is typically 2. For more information about the different query syntaxes, see Building Search Queries http: If you have a custom search application that is running on SharePoint Server , you should be aware that some advanced query features that are available through the Federation Object Model or the Query Web service will behave differently after you upgrade to FAST Search Server for SharePoint. Custom security trimming SharePoint Server provides support for custom security trimming of search results through a SecurityTrimmer interface ISecurityTrimmer2. All security trimming is performed as part of the query matching, based on ACL information that is stored in the index. Because FAST Search Server for SharePoint provides query refinement based on all items that match a query, this ensures that refinement counts only reflect the items that the user is entitled to see. Another option is to write custom crawlers that provide custom ACL information to the index. I used the following references in my post:

Chapter 3 : Add search capabilities to your apps for SharePoint â€“ Apps for Office and SharePoint blog

Introduction to text mining --Core text mining operations --Text mining preprocessing techniques --Categorization --Clustering --Information extraction --Probabilistic models for information extraction --Preprocessing applications using probabilistic and hybrid approaches --Presentation-layer considerations for browsing and query refinement.

Abstract Cluster analysis of multidimensional data is widely used in many research areas including financial, economical, sociological, and biological analyses. Finding natural subclasses in a data set not only reveals interesting patterns but also serves as a basis for further analyses. One of the troubles with cluster analysis is that evaluating how interesting a clustering result is to researchers is subjective, application-dependent, and even difficult to measure. This problem generally gets worse as dimensionality and the number of items grows. The remedy is to enable researchers to apply domain knowledge to facilitate insight about the significance of the clustering result. This article presents a way to better understand a clustering result by combining insights from two interactively coordinated visual displays of domain knowledge. The first is a parallel coordinates view powered by a direct-manipulation search. The second is a domain knowledge view containing a well-understood and meaningful tabular or hierarchical information for the same data set. Our examples depend on hierarchical clustering of gene expression data, coordinated with a parallel coordinates view and with the gene annotation and gene ontology. Introduction Cluster analysis is used in numerous research domains, including business, economical, sociological, and biological analyses. A cluster is a group of data items that are similar to others within the same group and are different from items in other groups. Clustering enables researchers to see overall distribution patterns, and identify interesting unusual patterns, and spot potential outliers. Moreover, clusters can serve as effective inputs to other analysis method such as classification. Researchers in various areas are still developing their own clustering algorithms even though there already exist a large number of general-purpose clustering algorithms. One reason is that it is difficult to understand a clustering algorithm well enough to apply it to a new data set. A more important reason is that it is difficult for researchers to validate or understand the clustering results in their own way or in terms of their knowledge of the data set. Current visual cluster analysis tools can be improved by allowing researchers to incorporate their domain knowledge into visual displays that are well coordinated with the clustering result view. This paper describes additions to our interactive visual cluster analysis tool, the Hierarchical Clustering Explorer [3]. We first briefly explain the interactive exploration of clustering results using our current version, HCE 3. In section 3, the design considerations for the direct-manipulation search tool and the dynamic queries are explained in detail. Section 4 presents a tabular view showing gene annotation and the gene ontology browser and section 5 covers some implementation issues. Other clustering algorithms automatically determine the number of clusters, but users may not be convinced of the result since they had little or no control over the clustering process. A hierarchical clustering result is generally represented as a binary tree called dendrogram whose subtrees are clusters. Considering that the lower a subtree is, the tighter the cluster is, we implemented two dynamic controls, minimum similarity bar and detail cutoff bar, which are shown over the dendrogram display. Users can control the number of clusters by using the minimum similarity bar whose y-coordinate determines the minimum similarity threshold. As users pull down the minimum similarity bar, they get tighter clusters lower subtrees that satisfy the current minimum similarity threshold. Users can control the level of detail by using the detail cutoff bar. All the subtrees below the detail cutoff bar are rendered using the average intensity of items in the subtree so that we can see the overall patterns of clusters without distraction by too much detail. Overall layout of HCE 3. Minimum similarity bar was pulled down to get 55 clusters in the Dendrogram View. A cluster of genes is selected in the dendrogram view and they are highlighted in scatterplots, detail view, and parallel coordinates view tab window see section 3. Users can select a tab among the seven tab windows at the bottom pane to investigate the data set coordinating with different views. Users can see the names of the selected genes and the actual expression values in the detail views. Since we get a different clustering result as a different linkage method or similarity measure is used in hierarchical clustering, we need some mechanisms to evaluate clustering results. HCE implements 3 different evaluation mechanisms. Two dendrograms are shown face to face, and when users double-click on a cluster of a dendrogram, they can see the lines connecting items in the cluster and the same items in the other dendrogram. When users click on a cluster in the dendrogram view, the items in the cluster are also highlighted in the k-means clustering result view the last tab in Figure 1 so that users can see if the two clustering results are consistent. Through these three mechanisms, HCE 3. We proposed a general method of using HCE 3. In section 3 and 4, we will use this data set to demonstrate how HCE 3. Parallel coordinates view Many microarray experiments measure gene expression over time [5][9]. Researchers would like to group genes with similar expression profiles or find interesting time-varying patterns in the data set by performing cluster analysis. Another way to identify genes with profiles similar to known genes is to directly search for the genes by specifying the expected pattern of a known gene. When researchers have some domain knowledge such as the expected pattern of a previously characterized gene, researchers can try to find genes similar to the expected pattern. Since it is not easy to specify the expected pattern at a single try, they have to conduct a series of searches for the expression profiles similar to the expected pattern. Therefore, they need an interactive visual analysis tool that allows easy modification of the expected pattern and rapid update of the search result. Clustering and direct profile search can complement each other. Since there is no perfect clustering algorithm right for all data sets and applications, direct profile search could be used to validate the clustering result by projecting the search result onto the clustering result view. Conversely, a clustering result could be used to validate the profile search by projecting the cluster result on the profile view. Therefore, coordination between a clustering result and a direct search result make the identification process more valid and effective. The built-in profile editor makes it possible to edit the search pattern, but the editor view is separate from the profile chart view where all matching profiles are shown, so users need to switch between two views to try a series of queries. The modification of master profile in the profile editor view is interactive, but search results are not updated dynamically as the master profile changes. TimeSearcher [7] supports interactive querying and exploration of time-series data. Users can specify interactive timeboxes over the time-varying patterns, and get back the profiles that pass though all the timeboxes. Users can drag and drop an item from the data set into the query window to create a query with a separate timebox for each time point over the item in the data set. Each timebox at each time point can be modified to change the query. Key design concepts are: Users can submit their queries simply by mouse drags over the search space rather than using a separate query specification window. This enables users to refine their query results, which follows the process of general problem solving. The parallel coordinates view consists of three parts Figure 2: Users specify a search pattern by simple mouse drags. As they drag the mouse over the information space, the intersection points of mouse cursor and vertical time lines define control points. A search pattern is a set of line segments connecting the contiguous control points specified. Users choose a search method and a similarity measure on the control panel. They can change the current search pattern by moving a control point a rectangular point on the search pattern , by moving a line segment vertically or horizontally, or by adding or removing control points. All of these modifications are done by mouse clicks or drags, and the results are updated instantaneously. Layout of the parallel coordinates view and an example of model-based query on the mouse muscle regeneration data. The data silhouette the gray shadow represents the coverage of all expression profiles. Thin regular solid lines are the result of the current query that satisfies the given similarity threshold more than The data set shown is a temporal gene expression profile on the mouse muscle regeneration [9]. Incremental query processing enables rapid updates within ms so that dynamic query control is possible for most microarray data sets. The easy and fast search for interesting patterns enables researchers to attempt multiple queries in a short period of time to get important insights into the underlying data set. In the parallel coordinates view, users can submit a new query over the current query result. If users click on a cluster in the dendrogram view, all items in the cluster are shown in the parallel coordinates view. By pinning this result, users can limit the search to the cluster to isolate more specific patterns in the cluster. Genes included in the search result are highlighted in the dendrogram view. Conversely, if users click on a cluster in the dendrogram view, profiles of the genes in the cluster are shown in the parallel coordinates view so that users can see the patterns of genes in a different view other than color mosaic. Through the coordination between the parallel coordinates view and the dendrogram view, users can easily see the representative patterns of clusters and compare patterns between clusters. Since queries done in the parallel coordinates view identify genes with a similar profile, the search results should be consistent with clustering results, if the same similarity function is used. In this regard, the parallel coordinates view helps researchers to validate the clustering results by applying their domain knowledge through direct-manipulation searches. In the parallel coordinates view, users can run a text search called search-by-name query by typing in a text string to find items whose name or description contains the string. Moreover, two different types of direct-manipulation queries are possible in the parallel coordinates view: The first measure is useful when the up-down trends of profiles are more important than the magnitudes, while the second and the third measures are useful when the actual magnitudes are more important. When users know the name of a biologically relevant gene, they can perform a text-based search first by entering a name or a description of the gene Figure 4. Finally, they adjust the similarity thresholds to get the satisfying results and project them onto other views including the dendrogram view. Ceilings and floors are novel visual metaphors to specify satisfactory value ranges using direct manipulation. A ceiling imposes upper bounds and a floor imposes lower bounds on the corresponding time points. Users can define ceilings and floors on the information space so that only the profiles between ceilings and floors are shown as a result Figure 3. Users can specify a ceiling by dragging with the left mouse button depressed, and a floor by dragging with the right mouse button depressed. They can change ceilings and floors with mouse actions in the same way as they did for changing search patterns in model-based queries. This type of query is useful when users know the up-down patterns and the appropriate value ranges at the corresponding time points of the target profiles. Compared to model-based queries, ceiling-and-floor queries allow users to specify separate bounds for each control point. An example of the Ceiling-and-Floor query. Bold line segments above the profiles define ceilings, and bold line segments below profiles define floors. Profiles below ceilings and above floors at the time points where ceilings or floors are defined are shown as a result. Users can move a line segment or a control point of ceilings or floors to modify current query. Researchers generated in vivo murine muscle regeneration expression profiling data using Affymetrix U74Av2 12, probe sets chips. They measured expression levels at 27 time points to find genes that are biologically relevant to the muscle regeneration process. They already have domain knowledge that MyoD is one of genes that are the most relevant to muscle regeneration.

Chapter 4 : CiteSeerX â€” Citation Query Introduction to Cataloging and Classification. Libraries Unlimited

Text mining is a new and exciting area of computer science research that tries to solve the crisis of information overload by combining techniques from data mining, machine learning, natural language processing, information retrieval, and knowledge management.

Show Context Citation Context To reduce information overload, the user interface supports pre- and post-query refinement capabilities. In , our paper that describes the design and implementation of Delaunay has received the best paper award Navigating query results is a highly volatile task, usually requiring much effort from the users, while not providing a firm reference point for further queries or refinement. Many users accessing traditional search engines are confronted with lengthy pages of hypertext links, through which relevant Many users accessing traditional search engines are confronted with lengthy pages of hypertext links, through which relevant information must be found. Accessing each link can bring a user closer or farther from the information they seek; in either case, the context between the query results and the current web page can be lost. Delaunay MM addresses these issues by creating virtual documents through which all query results are displayed in a meaningful and organized fashion; and that reference the originating web page, thus providing semantic browsing and hypermedia navigation without loss of context. In a real-world decision support application, users often want to search data from various sources according to some criteria, build a visualization based on the data being retrieved, and use the visualization to explore the data. With our approach, these activities are supported within the same wor With our approach, these activities are supported within the same workspace. The resulting views are then arranged into a larger coordinated view. In our layered architecture, data flows through the layers becoming encapsulated inside of metadata that describes the visual attributes being added. This metadata determines both the individual views and the dynamic interactions within a coordinated view. Dynamic interactions are implemented using a mediated notification services architecture. Delaunay View shares with Isabel F. The Personal Digital Historian PDH is an ongoing research project aimed at allowing groups of people to casually browse, embellish, and explore large collections of their personal data, such as pictures, video, or more business-related items such as spreadsheets or PowerPoint slides. Our initial prototype system is designed for a tabletop display and to be used while people are talking to each other. In this paper, we focus exclusively on describing those aspects of our project which provide a visual interface to support exploration of a database of personal data. The interface allows people to organize their images along the four questions essential to storytelling: Users are provided with a wide variety of flexible interaction methods, including region of interest query specification with in-place freeform stroke input, image-based book marking, suggestion generation via automatic query relaxation, and output summarization. With this interface, the users can enjoy their conversation while having the photos at their finger tips, rather than being distracted by the effort of formulating queries. Keywords Multi-person interactive visual interface, table top display, visual navigation, personal databases. We present a multi-layered framework for the visual presentation of information that results from the fusion of multimedia databases. The visual presentation allows for the dynamic interaction among the views that form that presentation. By manipulating elements in a view, the other views may change By manipulating elements in a view, the other views may change dynamically, thus allowing for interactively querying and browsing related database objects. This dynamic interaction is established by the semantic relationships among the objects in the multimedia databases, which originate from the fusion layer and percolate to the presentation layer. Data objects, visual objects, and relationships in each layer of the framework are stored in XML format. We have built a prototype that embodies these principles. A view can be for example a bar chart, a bipartite graph, or a tree. A view is bound to a data Facilitating information retrieval in the vastly growing realm of digital media has become increasingly difficult. Delaunay MM seeks to assist all users in finding relevant information through an interactive interface that supports pre- and post-query refinement, and a customizable multimedia inform Delaunay MM seeks to assist all users in finding relevant information through an interactive interface that supports pre- and post-query refinement, and a customizable multimedia information display. This project leverages the strengths of visual query languages with a resourceful framework to provide users with a single intuitive interface. The interface and its supporting framework are described in this paper. All other users performed the study over the course of a week for 30 to 45 minutes at their leisure. The overall results of the study were very positive.

Chapter 5 : CiteSeerX â€” Citation Query Knowledge Discovery in Databases

Text mining is a new and exciting area of computer science that tries to solve the crisis of information overload by combining techniques from data mining, machine learning, natural language processing, information retrieval, and knowledge management. The Text Mining Handbook presents a.

Fast traversal of specialized publications, customer support knowledge bases or document repositories allows enterprises to deliver service efficiently and effectively. Instead, enterprises can deliver a custom search experience that saves their clients time and provides them better service through a question and answer format. We share our learnings, process, and custom code in this code story. Finally, measuring retrieval performance is key to optimizing the quality of the experience, as managing the quality of the consumer search engine experience is an ongoing task. Few guidelines exist to provide developers with a comprehensive view of processes and best practices to design, optimize, and improve custom search. Moreover, there are few tools that aid developers in the process of measuring how well their custom search engine performs at retrieving what the user intended to retrieve. From text pre-processing and enrichment to interactive querying and testing, each step could benefit from a process road map, how-to guidelines, and better tools. Enterprises have questions such as: Which techniques should be used at what time? What is the performance impact of different optimization choices on retrieval quality? Which set of optimizations performs the best? We leveraged Azure Search and Cognitive Services and we share our custom code for iterative testing, measurement and indexer redeployment. These design considerations will help you create an enterprise search experience that rivals the best consumer search engines. The first step is to understand the custom search life cycle, which involves designing the search experience, collecting and processing content, preparing the content for serving, serving and monitoring, and finally collecting feedback. In Web search engines, for instance, the user intent falls into one of three categories: Surfing directly to a specific website e. Completing a specific task e. Browsing for general information about a topic using free-form queries e. Answering these ten key questions will give you a high-level set of requirements for your end-to-end custom search design. Which user intents will be supported? Is the content available to answer the user queries? Is there any data acquisition or collection that is required to assemble the necessary pieces of content? What type of content will be served: How will the content be served for each intent or sub-intent? How will the user interface work? Which delivery interface s will be supported e. Will the experience include content from more than one source? Which user signals will be automatically captured for analysis? What type of user feedback will be solicited? How will it be solicited: What success metrics are there? Are they objective, subjective or both? How will they be computed? How do you compare alternative experiences? How will you decide which experience is better in the potential situation of conflicting metrics? Define Success Measures and Feedback Define your desired objective success metrics. Is success displaying the best answer in the top five responses, the top three responses, or only in the first response? The success measures will be used in the optimization of the search experience, as well as for ongoing management. Consider measures you will need to optimize for launch and for ongoing performance management. Also consider your approach to experimentation. Search relevance how to define and measure search success including objective and subjective success metrics.

Chapter 6 : Table of contents for The text mining handbook

The Challenge. Querying specific content areas quickly and easily is a common enterprise need. Fast traversal of specialized publications, customer support knowledge bases or document repositories allows enterprises to deliver service efficiently and effectively.

Chapter 7 : Migration to Fast Search for SharePoint â€“ General Considerations â€“ MEA SI Blog

DOWNLOAD PDF PRESENTATION-LAYER CONSIDERATIONS FOR BROWSING AND QUERY REFINEMENT

Preface I. Introduction to Text Mining I.1 Defining Text Mining I.2 General Architecture of Text Mining Systems II. Core Text Mining Operations II.1 Core Text Mining Operations II.2 Using Background Knowledge for Text Mining II.3 Text Mining Query Languages III.