MULTI-DIMENSIONAL FUZZY SEARCH FOR PERSONAL INFORMATION MANAGEMENT, SYSTEMSPROCESSING, MULTIMEDIA INFORMATION PROCESSING, IN THE CONTEXT OF SOCIAL NETWORKING

M V Ramana Murthy
HOD, Department of Mathematics, Osmania Unviersity, Hyderabad
Gopal Menon & Jeelani
Hyderabad central university, Gachhibowli, India

Abstract

Explosion of information is driving data users for complex search tools to have access to heterogeneous data in a simple and efficient manner which is the critical need of the hour. As users store and collect large amounts of personal data it is becoming increasingly important to quickly retrieve files in personal information systems. Numerous search tools have been developed to locate personal information stored in file systems.Existing tools often allow IR-style ranking on the textual part of the query, but consider structure (e.g., file directory) and metadata (e.g., date, file type) as filtering conditions.

In this paper we propose a novel multi-dimensional approach to semi-structured and heterogeneous data searches in personal information management systems.The paper highlights the suite of techniques that allow search tools to effectively and efficiently evaluate query conditions.The paper identifies how to score fuzzy conditions in each query dimension (content, structure, and metadata) in the search,providing a complex query interface. Furthermore, the paper proposes to integrate the three dimension scores into a meaningful unified score, such that users can specify a single query and that can be evaluated at once across file boundaries. Finally, algorithms and data structures are designed to support efficient processing of the multi-dimensional as well as unified queries. We perform a thorough experimental evaluation of our approach and show that the proposed system has the potential to significantly improve ranking accuracy.In addition, we show that scoring efficiently evaluates fuzzy multi-dimensional queries and also show that query processing strategies perform and scale well, making our fuzzy search approach practical for every day usage.

Keywords: Personal Information Management Systems, Fuzzy Search, IR Ranking, Information Retrieval

Introduction

The evolution of memory technology shows a permanent increase of capacity in the last few years which is accompanied by a decrease in size and price (Williams, 2011)[11], (Parrenson, 2011)[10], (Gantz and Reinsel, 2011)[5]. So users are able to store large amount of heterogeneous data (structured and unstructured) on their personal computers. The general study of personal aspects of everyday document management is known as personal information management (PIM).The study of how individuals find and use the information they collect has been an area of focused study for information scientists.The key to the study of personal information management is the idea of an ‘individual’ or ‘unique’ way of finding,

storing, and working with information in more private spaces.Overtime, the definition of PIM has become more extensive. Jones and Teevan [6], they define PIM asboth the practice and the study of the activities people perform to acquire, organize,maintain, retrieve, use, and control the distribution of information items such asdocuments (paper-based and digital), Web pages, and email messages for everyday use tocomplete tasks (work-related or not) and to fulfill a person's various roles.Researchers studying PIM include information scientists, psychologists, computerscientists, and lately, even domain knowledge experts, such as engineers (Lansdale [9]; Jones [7]; Hicks, B.J., Dong, A., Palmer, R, & Mcalpine, H.C [4]).

Interest in the study of PIM has increased in recent years with the growing new applications, new gadgets, and with the overall complexity of PIM. The amount of data stored in personal information management systems is rapidly increasing, following the relentless growth in capacity and dropping prices of storage. This data explosion is driving a critical need for search tools to retrieve heterogeneous data in a simple and efficient manner. PIM systems are becoming more ubiquitous and present a need for improving and enhancing information search results using wide range of tools and procedures to store, manage, retrieve and show information. Numerous search tools have been developed to perform keyword searches and locate personal information stored in file systems, through commercial tools like Google Desktop Search and Spotlight.

Existing tools typically index text content, but only consider structure and metadata as filtering conditions. These tools usually support some form of ranking for the textual part of the querybut only consider structure (e.g., file directory) and metadata (e.g., date, file type) as filtering conditions. Thus, it is too rigid to use this information only as filtering conditions since any mistake in the query will lead to relevant files being missed.Because of the structural and data heterogeneity in PIMs, we believe it is critical to support approximate matches on both the structure and content components of queries and to allow for query conditions to be evaluated across file boundaries.Keyword-only searches do not exploit the rich structural information typically available in Personal Information Management Systems.

The scope of the project is a novel approach that allows users to efficiently perform fuzzy searches across three different dimensions: content, metadata and structure. We propose a framework to combine individual dimension scores into a unified multi-dimensional score. We adapt existing top-k query processing algorithms and propose optimizations to improve access to the structure dimension index. Later, we evaluate scoring framework experimentally and show that approach has the potential to significantly improve search accuracy over current filtering approaches. We empirically demonstrate the effect of our optimizations on query processing time and show that our optimizations drastically improve query efficiency and result in good scalability. While our work could be extended to a variety of data space applications and queries, we focus on a file search scenario in this paper. Of course, our techniques could be extended to a more flexible query model where pieces of data within files could be returned as results.

This paper is divided into 5 sections. Thepaperisstructuredasfollows: In Sec.1we present the overview of Personal Information Management (PIM). We also examine the PIM concepts followed by a more in-depth examination of various PIM research conclusions about organization; In Sec. 2we present the proposed multi-dimensional scoring framework; Sec. 3 describes the overall architecture of the system and the algorithms we use to aggregate scores and return the best answers to the queries. In Sec. 4 we present our experimental results. This literature review concludes by introducing the concept of personal organization. in Sec. 5.

Multi-dimensional scoring framework

Information retrieval (IR) is finding data of an unstructured nature satisfying information needs within large data collections. IR is becoming the dominant form of information access. Fuzzy set theory techniques have been proposed to improve the effectiveness and flexibility of search engines. Modern web search engines are based on boolean keyword-based formulation of queries, and a bag-of-words representation of documents.

Retrieval models (Bordogna and Pasi [1]) (Kerre, Zenner, and Caluwe [8]) in which documents are represented as fuzzy sets have also been proposed. Conceptually, fuzzy IR models are similar to that of the vector-space model. In fuzzy IR model the relevance of a document is calculated using fuzzy logic connectives, measuring the degree to which a document 'implies' a query term, and subsequently combining these degrees using flexible alternatives for the operations of boolean conjunction or disjunction. The main advantage of fuzzy IR models is in the flexibility they give users to specify their queries.

Top-k processing is a crucial requirement in interactive environments that involves massive amounts of data. The main objective is to return the K highest ranked answers quickly and efficiently.One common way to identify the top-k objects is scoring all objects based on some scoring function. Data objects are usually evaluated by multiple scoring predicates that contribute to the total object score. A score function is therefore usually defined as an aggregation over partial scores. Top-k processing techniques are classified based on the restrictions they impose on the scoring function. Most proposed techniques assume monotone scoring functions.

Inverse Document Frequency (IDF)is a popular measure to quantify words importance. Documents are represented as weighted collections of terms. Document can formally be modeled as a vector in a multidimensional space, with one dimension for each term occurring in the document.IDF is believed to measure a words ability to discriminate between the documents. IDF invariably appears in a host of heuristic used in Information Retrieval (Salton and McGill [3]). DF (Sparck Jones [7]) is defined as the logarithm of the ratio of number of documents in a collection to the number of number of documents containing the given word. Inverse document frequency (IDF) is defined as

$$\mathrm{idf}_t = \log\left(\frac{N}{\mathrm{df}_t}\right)$$

Thus the idf of a rare term is high, whereas the idf of a frequent term is likely to be low.

Tf-idf weightingscheme assigns to term t a weight in document d given by

$$\mathrm{tf} - \mathrm{idf}_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

The weight of a component of a document vector is calculated based on the frequency of the occurrence of the corresponding term in the document (term frequency), and on the number of documents of the collection in which this term appears (inverse document frequency). This vector form will prove to be crucial to scoring and ranking. As a first step, we introduce the overlap score measure: the score of a document d is the sum of overall query terms, of the number of times each of the query terms occur in d. We can refine this idea so that we add up not the number of occurrences of each query term, but instead the tf-idf weight of each term in

$$\mathrm{Score}\,(q, d) = \sum_{t \in q} \mathrm{tf} - \mathrm{idf}_{t,d}$$

The query of the user, which is provided as a list of keywords, can also be represented as a vector, by treating it as a document. A common approach to estimate the relevance of a document to a query calculates the cosine of the angle between the corresponding vectors.

Vector-space model of information retrieval (Salton, Wong and Yang [12]) has traditionally been the most popular approach to information retrieval, and is still considered state-of-the-art approach. However, state-of-the art performance in the vector-space model is obtained for variants of the aforementioned cosine-similarity which are difficult to interpret intuitively and rely on careful tweaking of the parameters involved (Zobel and Moffat [14]).

Our model allows users to query both the content of files (using a standard keyword-based model) as well as their structure (internal and/or external). A query over unified data model is a combination of structural patterns and content terms. Our query model allows for approximation in both the structure and content dimensions, as well as across the two dimensionsto avoid discarding relevant information because of mistakes in the query.We decompose a query into component path queries. Similar to many popular search approaches, we focus on a ranked query model where only the k best matches are returned to the user.We score individual paths of a given query using TF_IDF approach.Our strategy is to compute scores for answers based on how close they match the original query conditions.The score of a match is defined by a scoring function.Content closeness is based on the frequency of keywords in the query condition. For structure, we use query relaxations that make queries more general. IDF scores for relaxed queries and TF scores for matching files are computed. Individual path scores are then combined together into an overall score representing the answer's relevance to the query to produce a unified score. This computation can be viewed as a multidimensional scoring problem, with each component path query representing a distinct scoring dimension.The lowest matching node in the path query is identified as a match point. A file is an answer if its structure and content contain one or more match points. We argue that allowing flexible conditions on structure and metadata can significantly increase the quality and usefulness of search results in many search scenarios. We adapt the existing and popular Threshold Algorithm (TA) to efficiently solve multidimensional scoring problem.

Desktop search facilities can search across different forms of information providing a tremendous potential to support a more integrative access to information. Some of this potential has already been realized in available facilities such as Google Desktop, Spotlight, Longhorn, SIS etc.A variety of desktop search programs are now available. Few of the desktop search programs (Noda and Helwig [13]) available in the market as free test version include Microsoft Windows 7 search, Copernic Desktop Search, Google Desktop Search, Hulbee Desktop, xFriend personal Desktop Search, Archivarius 3000, Find and Run Robot (FARR) etc.For an effective retrieval of heterogeneous data a desktop search engine (DSE) is used more and more often. The goal of desktop search is to find some particular item. These tools check the contents of user's PC to find information relating to web browser histories, e-mail archives, text documents, sound files, images, videos etc. Historically, desktop search was initiated by Apple Computer'sAdvanced Technology Group in early 1990s as AppleSearch technology. Most desktop search engines build and maintain an index database to achieve reasonable performance when searching several gigabytes of data. Indexing usually takes place when the computer is idle and most search applications can be set to suspend it if a portable computer is running on batteries, in order to save power. Their disadvantage is that they can only feasibly search a certain folder tree, not the entire computer. In desktop search, the semantics of keyword queries are often context-aware. Untapped productivity and security are the two major concerns for large firms in implementing Desktop search. Nonetheless, TF-IDF scores do not always work well, for the following two reasons:

- TF-IDF scores only consider the frequency of the appearance of the keyword in the documents.

- Document with high TF-IDF score often does not mean it is the particular item that the user is trying to find.

The use of fuzzy search algorithms in real search engines is closely related to the phonetic algorithms, lexical stemming algorithms, which extract base part from different forms of the same word, statistic-based ranking or the use of some complex sophisticated metrics.

Architecture of the system

Before developing the tool it is necessary to determine the time factor, economy and company strength.The feasibility study of the proposed approach has been carried out. It is to ensure that the proposed approach is not a burden to the company/ organization. Three key considerations involved in the feasibility analysis include: Economic feasibility, Technical feasibility, Social feasibility.

Economic Feasibility: This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the R & D of the system is limited. Thus the proposed system is well within the budget and this is possible because most of the technologies used are freely available. Only the customized products have to be purchases.

Technical Feasibility: This study is carried out to check the technical requirements of the system. The proposed system has modest requirements, null changes or only minimal changes are required for implementing this system.

Social Feasibility:The aspect of the study is to check the level of acceptance of the system by the user. It included the study of the process of training the user to use the system efficiently.

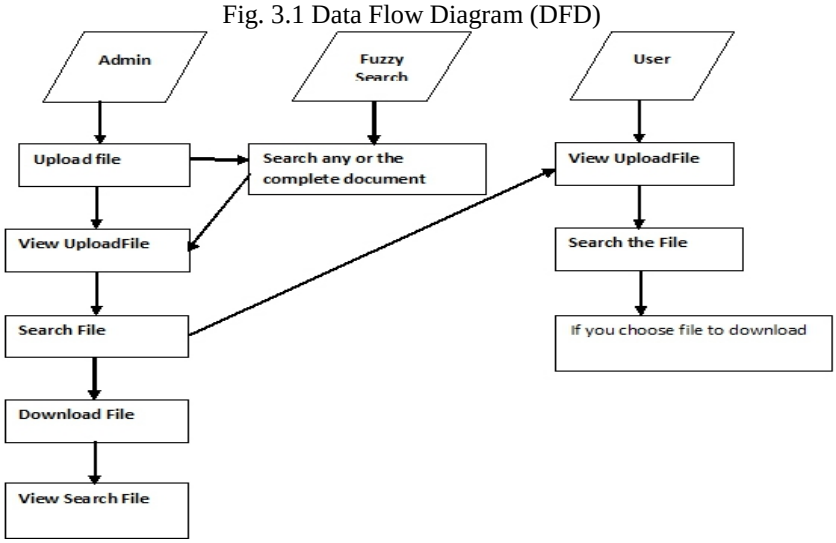
Process

In this approach we are creating a metadata for all the system files. In addition to saving all file names in a database it also saves some information from the text file in this module. While searching, the user enters the text to be searched for in the required file. It starts search from the database based on the filename. Later, it checks for some related file name. It then collects some text of the file and makes another search. Finally, it produces search results for corresponding text of the user.

Design

In Fig. 3.1 a simple graphical formalism of the system is representedwith the terms of the input data, various processing activities carried out, and the output data that is generated.

Fig. 3.1 Data Flow Diagram (DFD)



In Fig. 3.2 a detailed process of information flowis depicted in the form of Activity diagrams, sequence diagrams and class diagrams (UML Diagrams).

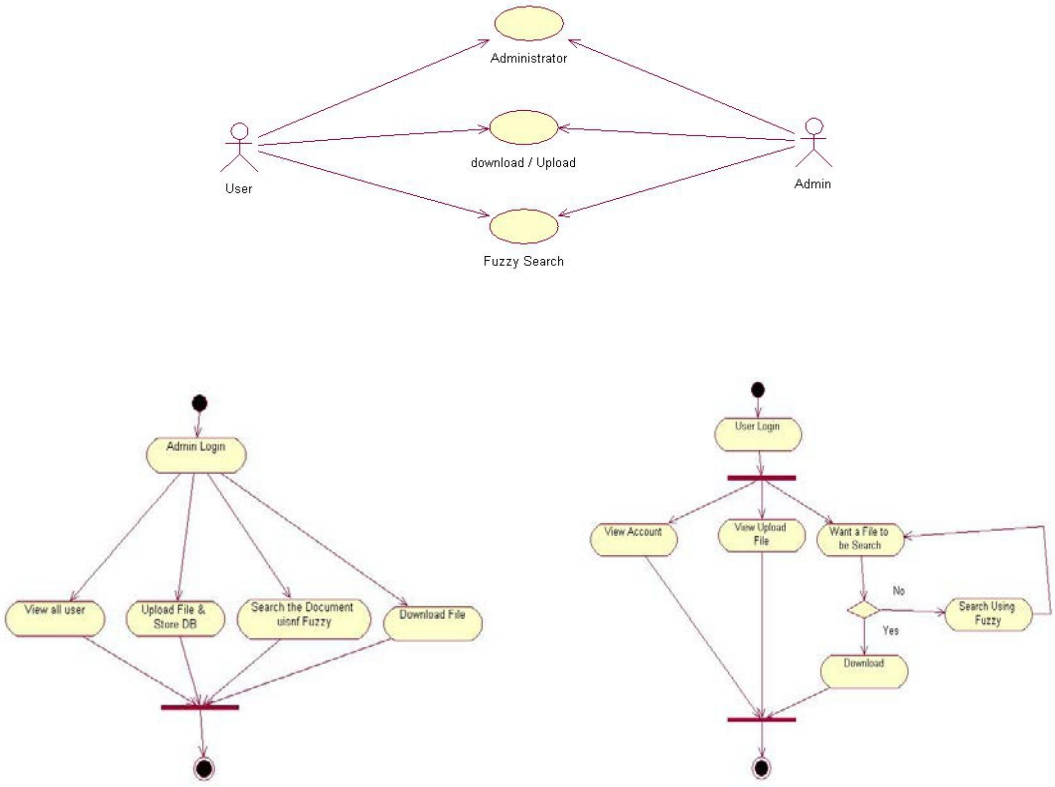


Fig. 3.2UML Diagrams

Experimental results

Once the feasibility study is completed, which operating system and programming language should be used needs to be determined. We now experimentally study and evaluate our unifiedsearch approach. Prototype of unified searchtool isimplemented using Tomcat 6.0.18 as Application Server; HTML, Java, JSP for designing front end applications; Scripting language used is Javascript, andMysql 5.0 is used to store all indexes using JDBC Database connectivity. This toolis run on a PC with a 2.4GHz Pentium IV, 512 MB RAM, and 40GB Hard disk capacity. Windows 95/98/2000/XP Operating System is used to run this tool effectively. Reported query processing times are averages of 40runs, after 40 warm-up runs to avoid measuring JIT effects.

Testing is the process of exercising software with the intent of ensuring the software system to meets its functional requirements and user expectations discovering every conceivable fault or weakness in a work product. Unit testing ensured that each unique path of the business process accurately performed to the documented specifications. Functional testing demonstrated systematic functioning against the business and technical requirements. In integration testing the components of the software system and the software applications were checked. System testing, white box testing, black box testing, unit testing and acceptance testing were also conducted to validate the tool designed. The tool designed has passedthe tests successfully. No defects were encountered.

Conclusion

We presented a unified scoring framework for multi-dimensional query processing over both content and structure in personal information systems. Based on the proposed approach we specificallydefined structure and metadata relaxations and proposed IDF-based

scoring approaches for content, metadata, and structure query conditions. Our experimental evaluation shows that our unified approach improves search accuracy by leveraging information from structure, content as well as relationships between the terms.We implemented and evaluated our scoring framework and query processing techniques. We have designed indexing structures and dynamic index construction and query processing optimizations to support efficient evaluation of multi-dimensional queries making multi-dimensional searches efficient enough for practical everyday usage.Our work shows that multi-dimensional score aggregation technique preserves the properties of individual dimension scores and has the potential to significantly improve ranking accuracyresulting in good overall query performance and scalability.Our evaluation shows the importance of structural query approximation in personal information queries and opens important research directions for efficient and high-quality search tools. In this paper, we have focused on files as the result unit. In the future, we will relax this restriction to allow for logical units of data to be returned.

References:

Bordogna, G., Pasi, G.: A fuzzy linguistic approach generalizing boolean information retrieval: A model and its evaluation. Journal of the American Society for Information Science 44(2), 70–82 (1993).

Electronic files: a mechanical engineer's perspective. ACM Transactions on Information, Ergonomics. 19 (1) 55-66.

G. Salton and M. J. McGill. 1983. Introduction to Modern Information Retrieval. McGraw-Hill.

Hicks, B.J., Dong, A., Palmer, R, & Mcalpine, H.C. (2008) Organizing and managing personalInformation Management. London: University of Washington Press.

J. Gantz, D. Reinsel, The Digital Universe Decade – Are You Ready?May 2010, http://www.emc.com/collateral/demos/microsites/idc-digital-universe/iview.htm (Accessed 2011-04-10).

Jones, W. & Teevan, J. (2007). Introduction. In W. Jones and J. Teevan (EDs.), Personal information management. Annual Review of Information.

K. Sparck Jones. 1973. Index term weighting. Information Storage and Retrieval, 9:619-633.

Kerre, E., Zenner, R., De Caluwe, R.: The use of fuzzy set theory in information retrieval and databases: A survey. Journal of the American Society for Information Science 37(5), 341–341 (1986).

Lansdale, M. (1988). The psychology of personal information management.

M. Parrenson, The hard drive turns 50. In: PCWorld, September 2006, http://www.pcworld.com/article/127104-2/the_hard_drive_turns_50.html (Accessed 2011-04-10).

M. Williams, Toshiba claims data storage break-through, San Francisco, 2010, http://www.infoworld.com/d/storage/toshiba-claimsdata-storage-breakthrough-278 (Accessed 2011-04-10) , of the ACM 18(11), 613–620 (1975).

Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. CommunicationsScienceand TechnologySystems, 26, (4-23).

T. Noda, S. Helwig, Benchmark Study of Desktop Search Tools –There`s more to search than Google & Yahoo! - An Evaluation of 12 Leading Desktop Search Tools. Wisconsin, 2005. http://www.uwebi.org/reports/desktop_search.pdf (Accessed 2011-04-10).

Zobel, J.,Moffat, A.: Exploring the similarity space. SIGIR Forum 32(1), 18–34 (1998).