clearyk6 / CA4009 Public

Code Issues Pull requests Actions Projects Security Insights

master

CA4009 / Notes / Chapter7.md

clearyk6 Notes on Enterprise Search and Recommender Systems History

1 contributor

133 lines (88 sloc) | 6.79 KB

Enterprise Search

Search for corporate or enterprise content, the application of Search Tech to info within an organisation

Forecast to be the biggest growth industry

Something like 80% of enterprise content is n *unstructured form*, not in a structured database

Must sort first

Find relevant info within the records of the organisations involved after.

Used by **employees** or other **authorities** seeking info held internally in a variety of formats and (potentially) at a number of locations. Potentially to legal agencies in the case of legal actions involving the organisation.

Users:

- class 1: Members of an organisation, familiar with the information/documents they searching for
- class 2: Members of the organisation, looking for info within the organisation but do not know where it may be found
- class 3: third parties who are looking for info relating to a topic of interest. Anything from extensive to no knowledge on info held within org

Keep the difference in users and vocab to be used, knowledge of the subject

Constraints:

- considerations of security
- inability to index specialised content (multimedia)
- difficulty integrating structured and non-structured content
- Cost, time and difficulty required to incorporate the diverse content repositories held within the organisation

Major challenge is the need to index content from a wide range of sources within an organisation, then search it effectively & efficiently.

Another challenge is the *metadata* associated with info from different sources in an enterprise may have different formats and/or have different metadata fields (eg email, meeting minutes etc) - *difficult to index in a consistent way within search index file*

Can be difficult to achieve and maintain coverage and freshness of the contents of an enterprise search system

- presence of multiple copies of the same content in diff places
- near duplicate detection
- difficulty in determining content changed recently
- network bandwidth issues
- difficulty of link extraction from JScript and Flash

Often there is *no link-structre* to assist with document ranking, cannot use PageRank type algorithms. Metrics such as popularity, recency of update, spam score of email, doc type, source repo may be used in query-independant weighting of docs.

Enterprise Search can include tools such as the following to bridge gap between terms occuring in queries/documents

- stemming
- thesaurus expansion
- relevance feedback
- query suggestion (of previously used queries)

Database Search

Typically use a complex language form such as SQL

- Requirement to know how the database is strucured
- difficult to pose similar queries to many differing databases
- no ranking of output from each DB
- no meaningful way to integrate the output from multiple DBs
- Where should the user start looking among retrieved content

Database search is like **boolean search** in IR

Improvements by using modern best match info retrieval search

- queries can be posed in simple language
- no requirement to know database structure
- same query can be posed to any doc collection
- meaningful ranking of retrieved docs
- ranked lists can be integrated into one (reliable merged ranking is difficult)
- user can start at the top of the ranked list
- No training required for any database technology
- Faster response tim (subsecond)
- Search engines can work with peak loads, db systems cannot

Database offloading:

- Queries are applied to a database to probe what it contains
- Each line of results retrieved from the database in response to a query becomes a psuedo 'document' which is indexed in a search engine
- Standard unstructured queries can then be applied to the indexed contents of the database to find out if it contains relevant info.

*The database still exists, but the database itselt is *not the main access source* to the info it contains

Best-Match search queries can be used to search db in 2 ways

1

- Connect directly to the db and transform the search request into SQL or the appropriate native database query language
- Apply transformed query to the db
- collect the output
    - this means the retrieved docs will not be ranked

2

- Use a search engine spider to crawl the pages generated from the db using database offloading
- Index the generated pages into a standard Text Retrieval system
- Apply Best-Match unstructured queries to the data indexed in teh information retrieval system
- Collect the ranked output

Access Control

Key aspect, right of access - diff people have diff restrictions. Egs, only the human resources office authorised to access personnel files; only financial office has access to financial data etc.

Early binding security: access control attributes are stored when the doc is indexed

Late binding security: each entry in the results list is checked at display time.

If data is v fresh, there may be no opp to set controls at index time..

Legislation in many counries requires organisations to store much of its info in case it is needed in legal disputes

eDiscovery is discovery in civil litigation which deals with the exchange of info in electronic format. Usually (but not always) a digital forensics analysis is performed to recover evidence for use in legal cases.

- Data is identified as relevant by lawyers, evidence is then extracted and analysed using digital forensic procedures.

Only need to hand over info required by law - search can be used to cleanse content of other info which is not legally required. AKA *litigation support*

UIs

Finding info within an enterprise should be fast and efficient, should be done through a single interface. Typically give users a much richer user interface functionality. No ad space!

User Needs in Enterprise Search

Often looking for a single known item. This is often better facilitated through *exploration* of the available content - **exploratory search**

Supported through methods including:

- Document clustering
- Faceted Search

(These more complex interfaces may require the user to be trained in the use of the sstem & its interfaces)

Document Clustering: based on metadata fields (from same email sender, same date etc) or based on their contents where docs with similar content are placed into clusters as potentially related items. Algos from content-based clustering beyond this module's scope.

Faceted Search

A technique for accessing information by filetering items based on facets of the info

- Each facet typically corresponds to the possible values of a property common to all objects eg author, lang, format, date etc
- Faceted search is useful if the searcher remembers one or more details of the item they are looking for, even if they can't remember enough for a meaningful search query.
- Faceted search can combine text search choices in facet dimensions eg first narrow by sender, then by date etc...