XML and Content Management

Lecture 13: Enterprise search

Maciej Ogrodniczuk

MIMUW, 10 January 2011

Enterprise search

Enterprise search is not about searching the Web (i.e. "Google syntax") or database search, but about indexing, querying and presenting of company documents to authorized users.

Motto: search engine is the most important component of the document management system.

Differences, compared with Internet search:

- limited scope (usually intranet, familiar access rights etc., ...)
- possibility to build on company standards (e.g. set of common metadata),
- document ranking not that important,
- need for searching various sources, various formats,
- better relevance of the result list,
- no garbage!

Search process

User's point of view:

- target: „find me the best answer for a given question",
- enter the query,
- wait for result list.

System's point of view:

- analyze the query,
- get access to data,
- analyze data,
- create the result list, apply user permissions,
- order results,
- present the list to the user.

But what is really happening when we search?

Naive idea:

1. browse documents one by one,
2. if a document contains the element user was searching for (metadata, word, phrase, pattern) – inform the user,
3. if results are to be sorted, collect them in some temporary data structure and display only the matching ones after checking all documents.

Works for 100 documents. But what for 100,000?

Huge amounts of data cannot be searched effectively without **indexing** the content first.

Index is a data structure comparable to what we can find in (good) books, optimized for search and typically containing information:

- on occurrence of a word in a document,
- usually also about the exact place where it occurred.

Index properties

Important issues:

- the index must be up-to-date since it is a primary source of information returned to the user,
- frequency and method of synchronization of the index with indexed documents depends on application and technical constraints:
  - when incremental update is not feasible, the whole index must be rebuilt,
  - when systematic update is not possible, cyclical update must be performed,
- the process of indexing can generate additional data useful for result display (e.g. document summaries).

Should index contain all words?

Stopwords are frequent words which separately do not convey any meaning, e.g. definite/indefinite articles, conjunctions, prepositions, pronouns, demonstratives, possessives etc.

Should we search stopwords?

- independently – surely not, since they appear in (almost) any document,
+ why Google does?

Should we index stopwords?

- index will be smaller without them, and search simpler (after removing them from the query),
+ searching „Take That" or „The The" will not be possible without them...

A piece of advice: would be nice to defined your own stopword list in the search module.

What do we mean by "words"?

Splitting the text into some basic units (e.g. to put them in the index) is called **text segmentation** (tokenization).

Text segments can be shorter than orthographic words ("from space to space").

Tokenization problems:

- knowing language of text is important (e.g. Asian languages do not contain spaces),
- character encoding is important (to separate whitespace from "the real text" – see `U+00A0`),
- which character should the text be split at? (→ *can't*, *długośmy*, *przyszedłby*, *doń*, *polsko-niemiecki*, *pięćdziesięciozłotowy*),
- does space always separate segments? (*twenty five*),
- what happens to dates? numbers? currency expressions?
- abbreviations (prof. = professor?)
- ...

What happens to inflected words?

The problem: we need to know the rules of *"playing roulette"* – we enter the query into the search field... and it is not found since the only document had *"how to play roulette"*.

Placebo: wildcards in the query: *"play* roulette"*.

Panacea: inflection search (stemming, lemmatization).

Index can contain base forms of the words (e.g. nouns in the nominative, verbs in the infinitive etc.) and the query is normalized in the same way.

The problem is not trivial, especially in Polish: *„Dudek, obciąć pensję"* vs. *„Real obetnie pensję Dudkowi"*.

Note: changes in the index can make words indistinguishable ($\rightarrow$ *bore* = nudziarz vs. *bear* = niedźwiedź, pol. *zęby* vs. *żeby*).

Best lemmatizer for Polish: zainteresowanych: `http://www.sgjp.pl/morfeusz/`

Multilingual lemmatization

Obviously: we need modules for every language (within our interest) for:

- lemmatizing documents,
- lemmatizing the query,
- matching the query against document content.

Less obviously: which module should be used for stemming?

Document language can be detected automatically (using statistical methods, with almost 100% chances of success):

- searching for words characteristic for a language,
- searching for characteristic n-grams (segments of letters),
- comparing how text compresses with a referential chart,
- ...

Last but not least: query language must be set (manually, assign by default etc.) which can be difficult for short query texts (→ *stare forty*).

Spelling hints

We all know how it works:

- a user enters the query,
- the system "guesses" what user is about – as he/she types or just after first results are returned (e.g. only when no results are retrieved).

Two methods of query correction:

- automated, by changing the misspelled word into its correct version,
- half-automated, by showing the correct spelling to the user and letting him execute the query again using the right version.

Spelling hints

Spelling control can be usually configured by changing:

1. similarity between two words which must be maintained to let the system accept that one word is a misspelled version of another one,
2. similarity which should result in automated correction of the misspelled version.

What are the hints really coming from?

- the list of words from indexed texts,
- a separate (orthographic) dictionary.

Golden rule: don't suggest anything which cannot be found.

Indexing heterogeneuos sources and text extraction

Text sources are not always files, but also:

- online content,
- CLOB database field content (get text, use column names for metadata extraction),
- e-mails,
- ...

which makes access to data as important as the search engine.

Extraction problems:

- extraction is simple for plain text files,
- but much complex for other text formats (e.g. remove HTML tags, use `alt` attribute content and `<meta>` data; don't spoil Polish diacritics in PDFs),
- requires dedicated converters for other formats.

Sorting the results

What does it mean that the documents "matches" the query?

The place of the result on the list results from many parameters – sometimes not very obvious:

- occurrence of a word from the query in a document (appeared in the lead = very important),
- occurrence of a word in metadata, link texts (*PageRank*),
- any advance on that?
- ...

Most popular model for representing documents and queries: *vector space* made by all indexed words (each making a separate dimension).

Query metrics

A document (bag of words) and a query can be modelled as multidimensional vectors storing information on occurrence of words in document/query contents (occurrence results in a non-zero value).

Quality of the result is a function of distance between *the document vector* and *query vector* according to a defined metrics.

Two most important metrics:

- binary – vector contains 1 for every word in the document and 0 for all other words,
- TF/IDF – the ratio of the *term frequency* to *inverse document frequency* (indicating how "important" words are – the lower, the more documents contain a given word.

Similarly a distance between document vectors can be calculated to create a list of *similar documents*.

When full-text search is not enough

Full-text search works best when a user can "predict" which words were used in search documents (which is not always the case):

- queries can contain synonyms of words used in documents („azure cardigan"),
- documents can contain detailed terms without indicating the general class („KDL-19S5730E does not play CD") while queries can be general („underground in Poland"),
- users don't always know what they are searching for („TV not working").

Solutions:

- synonym-enabled search,
- dialog-controlled search,
- decision trees,

which generally makes it a search with a (knowledge) model.

The simplest model: metadata, dictionaries, taxonomies

Text metadata can contain arbitrary content (e.g. article title) or a value from restricted set (newspaper section name, country etc.) – a dictionary.

Metadata values can be set in two ways:

- "manually", i.e. basing on document/form fields etc.
- automatically – basing on document content.

The metadata set makes a simple data model which can be used to:

- display the user interface (advanced search),
- dialogue with the user to fine-tune query criteria,
- filter results,
- ...

Dialog-controlled search

When some important search criteria (to be assigned to metadata/ attributes in the model) are missing from the query and number of results is high, the system can automatically generate some additional questions to the user.

The site of a used car dealer's:

- *car make* – Audi, Fiat, ...
- *model* – make-dependent: A4, A6, A8, TT, ...
- *production year*, *price*, *mileage*, *colour*, ...

A query: *Audi for less than 10,000 EUR*.

System help: a form showing additional criteria basing on indexed documents:

- which model? A4, A6, A8? (no TT at that price),
- which year?
- which mileage? less than 100K, 100K-200K, over 200K?

Filtering results

Similarly filters for narrowing result list can be made available after the query has executed.

Benefits as compared to repeated form filling:

- dynamic, up-to-date information on the number of results in each class,
- eliminating empty answers – no more guessing.

Decision trees

Decision trees facilitate interactive solving of difficult problems.

The tree is a graph of choices resulting in solving the problem:

- every node contains a question and a set of answers,
- answers can direct to other nodes (with questions) or, finally, solutions.

How does it work?

- answering the question sets values of certain attributes of the query (as with advanced search),
- already set values can be used as parameters to further queries,
- if dialog does not lead to the solution, "standard" search can be used.

Decision trees – applications and further capabilities

For diagnostic tests:

- in call centres and contact centres,
- service centres,
- other specialized self-service applications.

More about decision trees:

- graphical tree-building IDEs,
- multilinguality in the node,
- paths depending on user/group role,
- default answers.

Decision trees – a GUI example

start.dt | recording.dt | *power.dt | playback.dt | clock.dt

Selection
Marquee
Components
Question
Antwort
Solution
Tree Reference
Pipe Call

power_general

Who is the producer of your appliance?

Sony | Samsung | Grundig | JVC | Panasonic

Philips
Sony
Toshiba
Panasonic
Philips

Switch off the POWER SAVE MODE! Does it work now?

Yes | No

Problem solved!

Did you just install or move the recorder?

No | Yes

Hand the recorder to your local dealer ...

The recorder was autom. switche...

Translation

	de	en
Question Who...		
Question	Von welchem Hersteller ist ihr Gerät?	Who is the producer of your appliance?
Process		
Role		
Comment		
Attribute	Att_Manufacturer	
Display		

Search user interface

Tips and tricks from designer's notebook:

1. **keep it simple, stupid**:
   - no sophisticated help system can fight the intuition of user who would like "to start searching immediately",
   - graphical design is worth investing in,
   - usability tests are more than necessary (for search form and result list).
2. **anyone should be happy**:
   - most queries are no longer than 3 words and only 5% uses operators,
   - but: advanced users can need more, so "advanced search" is still needed.
3. **metadata**:
   - use the most important ones (e.g. document type/format) even in the simplest search form,
   - don't overload the single result with metadata.

More designer's notebook hints

4. **result list**:
   - add header information: terms what were searched for, number of results, spelling hints etc.,
   - limit the basic layout to most important metadata,
   - show document sizes (to alert users to large documents); if multimedia content is found, add some visual player,
   - show the search context with query terms highlighted to make it clear why a certain result is not the list,
   - use sorting, paging, grouping, filtering results.

More designer's notebook hints

5. **less means more**:
   - present data from different perspectives right on the first page by categorizing them,
   - allow search in returned results,
   - use tabs and filters to group the results,
   - maintaining taxonomies is costly, but having them is appreciated by users,
6. **be careful about bells and whistles**:
   - a hyperbolic tree is good for a demo, but does it really help in daily work?
   - one new component showing unobvious relations between data (an interesting metadata filter, context links etc.) will always pay for itself,
7. **forget the grumblers** – user must learn to use even for best-tested and most intuitive interface.

Search performance and scalability

Two basic methods of dealing with extensive usage of a computer system:

1. parallelizing installations,
2. modularizing the system (e.g. separating index update from result retrieval, splitting the index into parts).

Note:

- 100% availability of a system is not possible in practice and improving it 10 times (e.g. from 99% to 99,9%) generates 10 times higher costs,
- a ratio of costs of improving availability to losses related to keeping it untouched should be always considered.

Another important rule: avoid reacting to unrealistic threats.

Search and security

Two methods of controlling search result display:

1. indexing content with access permissions (*early binding*) which automatically excludes protected documents from the result list,
2. verification of permissions at resource access: (*late binding*):
   a. showing all documents on the list, checking permissions at access,
   b. removing restricted documents from the result list even before displaying it.
3. or both!

Data protection

Results:

1. every change of access rights influences the index (updating it may be time-consuming and therefore done offline!), so it can happen that user can access documents which he/she should not (since permission has already been taken back, but not yet propagated to the index),
2. permissions controlled in real time, indexer independent on authorization, but checking access rights first could limit the part of the index to be browsed,
   a. seems safest, but should we let Mr. Smith to find the document `contract-termination-smith-1-feb.doc`?
   b. good idea, but lists of related documents should also be updated, number of results recalculated, hints specifically processed...

Typical search problems (and solutions)

Advice to users:

- is the spelling correct?
- did I use synonym?
- isn't the query too general?
- is the query too specific?
- is there any language problem?
- do I have permissions do display results?

Advice to administrator:

- analyze logs:
  - which queries were most frequent?
  - which errors were more frequent?
  - which queries did not return any results?
- add modules which may help users.

Selecting the best search system

3 basic types of search systems:

- software search systems,
- search appliances,
- remote search services.

Which functions do we need?

- which search methods?
- flexible interface, easily integrating into the company portal?
- just the form and result list – or we also need API?
- export of results to XML?
- which modules? can new ones be easily implemented?
- which document sources can be indexed?
- performance, scalability, load-balancing methods, caching...

How to compare search systems?

- index representative data sources,
- compare the lists of indexed documents,
- compare results lists for a given test query set.

Deployment of the search system

- gradual approach: index the main server first, leave "the whole intranet" for later,
- monitor, what users are doing and broaden the scope,
- think what to index – everything since users can have different needs,
- reasonable minimum:
  - filesystems, network drives, mail archives,
  - wiki servers, bug tracking systems (bugzilla etc.),
  - company site, text content in the database,
  - specialized programs, if they can still be integrated,
- think about index update strategy – every 5 minutes or in the night? sending data to indexing processes (*push*) or letting it to be downloaded independently (*pull*)?

Maintenance of the search system

Important issues:

- is it possible to remove content from index? how?
- how is the content indexed? how is index updated?
- how are technical problems taken care of (e.g. when unavailable external source is accessed)?
- how taxonomy changes influence the performance and validity of the system?
- is it possible (and how?) to add new metadata? new document types? does it require index rebuild? interface change?

Note: changes in the model are potentially costly – to include new attributes and values in the search, the whole repository must be reindexed!

Scaling search

Important issues:

- does it really concern us?
- how fast are the number of our documents growing?
- how fast do we expect the number of request grow?
- which methods of scalability are offered by the search system?
  - buying new hardware only?
  - splitting modules over different physical machines?
  - balancing load of modules by running their multiple instances on different machines – which processes can be run this way (indexing, search)?

Case study – Centre for Eastern Studies

CES is a Polish think tank monitoring political, economic and social situation in Eastern countries.

Main assumptions for the search system:

- searching multilingual document base (at least PL, EN, DE, RU, UA),
- common availability of heterogeneous sources (own documents, news services, thematic portals),
- answers in a few seconds,
- using linguistic tools and advanced search methods,
- search interface available from a browser with portlet GUI,
- personalization,
- permission system.

Case study – Centre for Eastern Studies

Result:

- integration of many sources in a single repository,
- two methods of adding documents:
  - manual, as documents are created by analysts, by means of a Web form,
  - automated, to cyclically import documents from Polish, Russian and Ukrainian news services.
- data model:
  - specific document types,
  - multilingual transcripts of names (e.g. searching documents with name returns documents in which it is written as Medvedev or Miedwiediew),
  - multilingual synonyms (searching for „bezrobocie w Polsce" returns also „unemployment in Poland"),
- full-text inflection search: integrated stemmers for Polish, English, Russian and German,

Case study – Centre for Eastern Studies

Result:

- queries:
  - in natural language ("Google-like syntax"),
  - by metadata form,
- filtering result list to restrict results to selected regions, languages, dates and some other important document properties,
- viewing documents in HTML format: automated converter from e.g. 200 popular formats.

System interface

Wyszukiwanie dokumentów

Zapytanie: Merkel Rosja [Szukaj] [Wyczyść]

Szukaj: wszystkich wyrazów / któregokolwiek z wyrazów — Język zapytania: PL

Teksty własne | Teksty własne tajne | Biogramy | Bazy działowe | Wiadomości | Subskrypcje | Agencje informacyjne

Sortowanie wg: **trafności** | daty — Zaznacz wszystkie wyniki na stronie

Liczba wyników: 1032 — Zaznaczonych: 0

1 2 3 4 5 6 7 8 9 10 — 1 - 10

Niemcy/Merkel: Rosja solidnym partnerem

5.3.Berlin (PAP) - Kanclerz Niemiec Angela **Merkel** uważa **Rosję** za solidnego partnera Unii Europejskiej. "Stawiamy na to, że **Rosja** jest solidnym partnerem. Dlatego popieram zasadniczo rosyjskie zaangażowanie gospodarcze w Europie" - powiedziała **Merkel** w wywiadzie opublikowanym w poniedziałek na łamach dziennika "Sueddeutsche Zeitung". Dodała, że Europejczycy są z kolei zainteresowani...

98% Agencje informacyjne, PAP/Zagranica TEXT 1 KB Data mod.: 2007-03-05 10:48

Rosja/ Merkel: problem polskiego mięsa wciąż nierozwiązany

21.1.Moskwa (PAP/AFP) - Kanclerz Niemiec Angela **Merkel** przyznała w niedzielę, że problem zakazu importu polskiego mięsa do **Rosji** jest wciąż nierozwiązany". Wyraziła jednak nadzieję, że rozmowy w sprawie porozumienia o partnerstwie UE-**Rosja** rozpoczną się przed końcem pierwszego półrocza tego roku. (PAP) mw/ ap/ 3847 18:18 07/01/21

97% Agencje informacyjne, PAP/Zagranica TEXT 386 B Data mod.: 2007-01-21 18:23

Niemcy/ Merkel zaniepokojona rozwojem demokracji w Rosji

7.1.Berlin (PAP/AP) - Kanclerz Niemiec Angela **Merkel** w rozmowie z tygodnikiem "Der Spiegel" wyraziła zaniepokojenie rozwojem demokracji w **Rosji**. Wywiad ukaże się w poniedziałek. "Nie możemy (...) przenosić naszej koncepcji demokracji do **Rosji** - powiedziała **Merkel**. - Jednocześnie, przyznaję, dochodzi (w **Rosji**) do zdarzeń, które postrzegam jako powód zatroskania, na przykład...

96% Agencje informacyjne, PAP/Zagranica TEXT 1 KB Data mod.: 2006-01-07 19:15

Rosja/ Merkel o energetyce i kolejnym szczycie G-8

17.7.Petersburg (PAP/Reuters) - Kanclerz Niemiec Angela **Merkel** powiedziała w poniedziałek na swej konferencji prasowej po szczycie G-8, że Niemcy nie obawiają się zależności energetycznej od **Rosji**. Dodała, że mimo zaufania do **Rosji** jako dostawcy, Niemcy rozważają dywersyfikację swoich źródeł energii. Kanclerz **Merkel** zapowiedziała, że wśród tematów kolejnego...

96% Agencje informacyjne, PAP/Zagranica TEXT 777 B Data mod.: 2006-07-17 17:15

Premier rozmawiał z Angelą Merkel

19.5.Warszawa (PAP) - O stosunkach Unii z **Rosją** rozmawiał w sobotę po południu premier Jarosław Kaczyński z kanclerz Niemiec Angelą **Merkel** - poinformował PAP rzecznik rządu Jan Dziedziczak. "Premier odbył ponad pół godzinną rozmowę telefoniczną z niemiecką kanclerz Angelą **Merkel**. Przedmiotem rozmowy były najbardziej aktualne kwestie związane z relacjami UE...

96% Agencje informacyjne, PAP/Kraj TEXT 955 B Data mod.: 2007-05-19 14:19

Rosja/ Merkel chce niezawodności dostaw rosyjskich nośników

energii do Europy (krótka2) 21.1.Moskwa (PAP) - Kanclerz Niemiec Angela **Merkel** powiedziała w niedzielę na konferencji prasowej w Soczi po spotkaniu z prezydentem Władimirem Putinem, że rozumie dążenie **Rosji** do wprowadzenie zasad rynkowych w handlu ropą naftową i gazem ziemnym z Ukrainą i Białorusią. Jednocześnie **Merkel** podkreśliła, że Niemcy opowiadają się za niezawodnością rosyjskich dostaw nośników energii do Europy. Niemiecka kanclerz dodała, że miała z Putinem "dobrą rozmowę" o tym, jak Europa i **Rosja** powinny lepiej się komunikować. W jej opinii, Europa i **Rosja** są sobie wzajemnie potrzebne. (PAP) mal/ ap/ 17:29 07/01/21

96% Agencje informacyjne, PAP/Zagranica TEXT 705 B Data mod.: 2007-01-21 17:33

Angela Merkel pojedzie do Moskwy na początku 2006 r.

25.11.Berlin (PAP/AFP,dpa) - Kanclerz Niemiec Angela **Merkel** uda się do Moskwy na początku 2006 r., aby spotkać się z prezydentem **Rosji** Władimirem Putinem - ogłosił w piątek rząd niemiecki. W piątkowej rozmowie telefonicznej Angela **Merkel** i Władimir Putin "potwierdzili, że doskonałe stosunki niemiecko-rosyjskie powinny być kontynuowane i rozwijać się" - dodała kancelaria kanclerska. Prezydent Putin był jednym z pierwszych szefów państw, który przesłał nowej kanclerz gratulacje, wyrażając "wiarę" w "pogłębienie partnerstwa strategicznego" między dwoma krajami.(PAP) kd/ ro/ 3453 15:03 05/11/25

96% Agencje informacyjne, PAP/Zagranica TEXT 695 B Data mod.: 2005-11-25 15:10

1 2 3 4 5 6 7 8 9 10 — 1 - 10

Filtry

Wyszukiwanie we wszystkich dokumentach.

Bazy

- Bazy
  - Teksty własne
  - Biogramy
  - Wiadomości [49]
  - Bazy działowe
  - Subskrypcje i agencje [983]
    - PAP [982]
    - Interfax [1]

Regiony

- Świat
  - Azja [318]
    - Azja Centralna [68]
      - Kazachstan [38]
      - Kirgistan [10]
      - Tadżykistan [5]
      - Turkmenistan [8]
      - Uzbekistan [7]
    - Kaukaz [113]
    - Turcja [93]
      - Ankara [24]

Daty

- Daty
  - 2005 [241]
  - 2006 [381]
  - 2007 [410]
    - 01 [96]
    - 02 [39]
    - 03 [56]
    - 04 [26]
    - 05 [82]
    - 06 [62]
    - 07 [24]

Języki

PL [1031] EN [0] DE [0] RU [1] inny [0]

Was supposed not to be about Google, but...

10 Google tips for enterprise search:

1. Keep your search users happy – if your search engine helps doing their work done, they will not question their cost. Ask users what else you can do for them.
2. Keep search speedy – easiness of querying and quick result retrieval encourage people to use search.
3. Make search ubiquitous – put the search field on every page and also on the result list, encouraging users to use the search engine and fine-tuning their queries.

Was supposed not to be about Google, but...

4. Keep your search pages clean and simple – keep the home/ main page simple. Move additional options to another page, limit the search unrelated navigation components on the result page.
5. Crawl as much content as possible – users tend not to find an important document since it has not been included into index.
6. Don't forget your non-HTML content – apart from HTMLs index also PDFs, MS Office documents etc.
7. Publish, publish, publish – encourage users to store their documents in indexed repositories (or at least on the network drive and not their computers.

Was supposed not to be about Google, but...

8. Don't be afraid to crawl secure content – use authentication but try to show users what kind of information is available.
9. Run test queries against your content – try to search important/representative. Encourage users to test the search engine.
10. Measure the benefit and look to the future – monitor some predefined criteria (average number of searches per user or per day to track changes in user's attitude to the search tool) and make the selected solution scalable.