Az Oracle Text további lehetőségei

Szekció szerinti keresés

Amint a könyvben már említettük , az Oracle Textben lehetőség van a keresést a dokumentum valamely meghatározott szekciójára (SECTION) korlátozni. A rendszer több különböző szekcióhatár-értelmezést támogat, ebből a felhasználónak kell a megfelelő értelmezést kiválasztania. A szekcióhatárt a szekciócsoport típusa határozza meg, amelyek az alábbiak lehetnek:

- null_section_group: csak mondat vagy bekezdés szekciók lesznek;
- basic_section_group: a szekció határokat az <A> és </A> tagok határolják;
- html_section_group: HTML-forrásban értelmezett szekciók;
- xml_section_group: XML-forrásban értelmezett szekciók;
- auto_section_group: XML-ben automatikus szekció kijelölés;
- news_section_group: NEWSML-ben értelmezett határolás.

A szekciócsoporton belül több különböző szerepű szekciót lehet értelmezni, melyek az alábbiak:

- normálzóna (zone);
- mező (field);
- végjel (stop);
- metaadat (mdata);
- attribútum (attribute);
- speciális szekciók mint pl. mondat, bekezdés (sentence, paragraph).

A *normálzóna-szekció*k rendszerint a szöveg törzsét jelentik. A határoló elemeket a CTX_DDL csomag ADD_ZONE_SECTION eljárásával definiálhatjuk. *Mezőszekció*kat az ADD_FIELD_SECTION eljárással hozhatunk létre. Ezek hasonlóak a zónaszekciókhoz, azonban itt a kiemelt fontosságú adatokra fókuszálunk: a szekcióba eső részeket a rendszer külön indexeli, és ezen adatokhoz gyorsabb hozzáférést biztosít. Hátránya, hogy ez a szekciótípus nem ágyazható egymásba, és ilyen szekciók nem lehetnek átlapolóak. Automatikus szekciócsoportok esetén *végjelszekció*val tudunk egyes részeket kivonni az indexelés hatálya alól. A *metaadat-szekció*ba a normáladatokhoz tartozó leíró-információkat helyezhetjük le. Ezáltal egyazon lekérdezési operátorban szerepelhet az adat és leírója. A szekciókra történő szűkítésnél a WITHIN taggal lehet a vizsgálandó szekciót kijelölni.

Tezaurusz alapú keresés

A dokumentumokból történő keresés során a kulcsszavas keresés hátránya az, hogy a felhasználó nem ismeri a dokumentumok pontos szókészletét, így előfordulhat, hogy nem a keresett szó, hanem annak valamely rokon értelmű kifejezése szerepel a dokumentumokban. Ekkor a standard keresés sikertelen lesz, mivel a kereső rendszer nem talál illeszkedő dokumentumot. Ezt a problémát a tezaurusszal támogatott kereséssel lehet kiküszöbölni. A tezaurusz (fogalomtár) kezelése a CTX_THES csomag segítségével történik. A csomag tartalmazza a hierarchia felépítéséhez, ill. módosításához szükséges eljárásokat. Fontosabb elemei:

- tezaurusz létrehozása: CREATE_THESAURUS;
- új bejegyzés felvitele: CREATE_PHRASE(tezaurusz, fogalom);
- kapcsolat létesítése az elemek között: CREATE_RELATION(tezaurusz, forrásfogalom, kapcsolat, célfogalom), ahol a kapcsolat típusa lehet:
  - NT: specializáció,
  - BT: általánosítás,
  - RT: reláció,
  - SYN: szinonima,
  - Nyelv: fordítás;
- elemek, kapcsolatok megszüntetése: DROP_PHRASE, DROP_RELATION, DROP_THESAURUS;
- információlekérdezés, melynek elemei:
  - BT(fogalom, szint, tezaurusz): a fogalom megadott szintű általánosításait adja vissza,
  - TT(fogalom, tezaurusz): a fogalom gyökér fogalmait adja vissza
  - NT(fogalom, szint, tezaurusz): a fogalom megadott szintű specializációit adja vissza
  - SYN(fogalom, tezaurusz): a fogalom szinonimáit adja vissza.

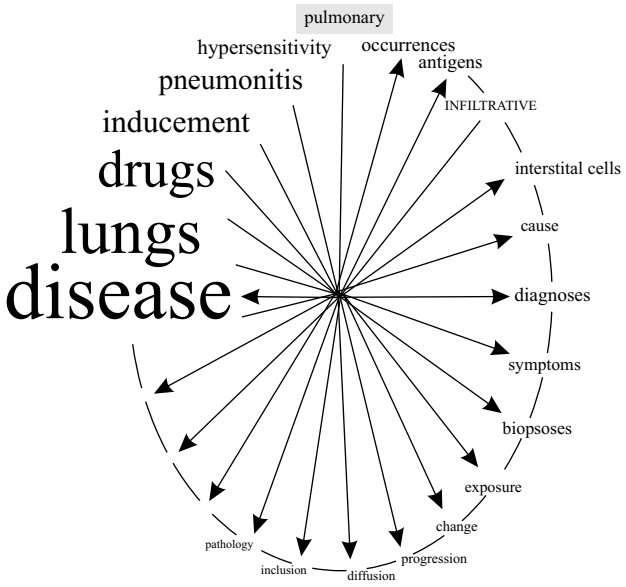
Grafikus megjelenítés

A felhasználóknak az eredmény vizuális megjelenítése gyakran többet mond, mint a puszta szöveg, ezért igen hasznos kiegészítő szolgáltatása az Oracle Text rendszerének a grafikus eredmény megjelenítését végző komponens. Az adatok grafikus megjelenítését CSS és Java programok végzik.

A dokumentum témaköreit leíró THEMES rutin alapesetben egy listát ad eredményül, melyben a kulcsszavak relevanciaértékükkel együtt szerepelnek. Ebből

a listából készít egy tématérképet a CSS-ben implementált ThemeMap modul. A megjelenítésnél a legfontosabb kulcsszavak a térkép közepén, nagyobb betűvel jelennek meg. Egy adott szó kiválasztása után egy részletesebb leírás jelenik meg.

Hasonló megjelenítési célt szolgál az Oracle Interactive Viewer modul is, melyben Java csomag fogja össze a különböző megjelenítő rutinokat. Az Interactive Viewer modul Java futtató környezet mellett használható. A csomag ThemeStar appletje az előzőben említett témakörlistát csillag alakú elrendezésben illusztrálja. A legfontosabb fogalmak ennél a megjelenítési módnál is nagyobb betűvel íródnak ki és a középponttól balra helyezkednek el (ld. 1. ábra). Ez a módszer akkor előnyös, amikor viszonylag kisebb számú kulcsszó található az eredményben. A következő ábra ezt a megjelenítési módot illusztrálja.



1. ábra. Oracle Text ThemeStar megjelenítési mód

A hierarchikus adatok megjelenítéséhez ad segítséget a SearchViewer Java modul. A megjelenítő képernyőn együtt látszik a hierarchia globális nézete és az éppen kijelölt rész részletes tartalma.

Az Oracle Text az angol nyelvű téma alapú kereséshez egy beépített témahierarchiát, alaptezauruszt is tartalmaz. A témakörök kijelölése igen nehéz feladat, hiszen sok szubjektív elemet tartalmaznak, és maga a témaszótár, ill. a témakörök időben dinamikusan változnak.

Mintapélda

A szövegkezelés jellegét egy egyszerűbb mintapéldával szemléltetjük. A feladat egy rugalmas, tezauruszon alapuló keresési módszer megvalósítása. A mintapélda működésének előfeltétele, hogy a CTXSYS-felhasználó már létezzen az adatbázisban a hozzá kapcsolódó jogokkal és adminisztrációs táblákkal együtt. Ha az Oracle Text rendszere már működik, akkor az alkalmazás felépítése a következő lépésekből áll össze.

1. Elsőként létrehozzuk a dokumentumokat tartalmazó táblát:

   ```
   CREATE TABLE doksik (kod NUMBER PRIMARY KEY, szoveg VARCHAR2(200));
   ```

2. Ezután megalkotjuk a táblához tartozó speciális indexet. Mivel most a szövegkeresést kívánjuk bemutatni, egy CONTEXT típusú indexre van szükség:

   ```
   CREATE INDEX idx_docs ON doksik(szoveg) INDEXTYPE IS CTXSYS.CONTEXT;
   ```

3. Ezután következhet a dokumentumtábla feltöltése adatokkal:

   ```
   INSERT INTO doksik VALUES(1,'I have a nice dog');
   ```

   A dokumentumtábla tartalmának módosítása után el kell végezni a kapcsolódó index frissítését is:

   ```
   EXEC CTX_DDL.SYNC_INDEX('idx_docs','2M');
   ```

4. Az index aktualizálása után a szövegkeresési alapfunkciók már használhatók. Például, a dog szót tartalmazó dokumentumok listája a

   ```
   SELECT szoveg FROM doksik WHERE CONTAINS(szoveg,'dog') > 0;
   ```

   paranccsal kérdezhető le. Ha a dokumentum illeszkedési súlyát is tudni szeretnénk, akkor a SCORE értéket is ki kell iratni:

   ```
   SELECT szoveg, SCORE(1) FROM doksik WHERE
   CONTAINS(szoveg,'dog',1) > 0;
   ```

5. A következő lépésben egy tezauruszt hozunk létre. Ehhez előbb megalkotjuk a témaorientált indexelést beállító paramétert:

   ```
   BEGIN
     CTX_DDL.CREATE_PREFERENCE('mylex','BASIC_LEXER');
     CTX_DDL.SET_ATTRIBUTE('mylex','INDEX_THEMES','YES');
   END;
   ```

   Az eljárás lefuttatásával létrejött a 'mylex' Lexer leíró objektum. Ezt adjuk most át az indexünknek:

   ```
   ALTER INDEX idx_docs REBUILD PARAMETERS('REPLACE LEXER mylex');
   ```

6. A tezaurusz létrehozását is több lépésben valósítjuk meg. Elsőként egy üres tezauruszt állítunk elő:

   ```
   EXEC CTX_THES.CREATE_THESAURUS('sajattz',FALSE);
   ```

7. Ezután egyenként feltöltjük előbb a fogalmakkal:

   ```
   EXEC CTX_THES.CREATE_PHRASE('sajattz','animal');
   ```

8. Majd a fogalmak feltöltése után megadjuk a köztük értelmezett kapcsolatokat:

   ```
   EXEC CTX_THES.CREATE_RELATION('sajattz','dog','BT','animal');
   ```

   Az előző paranccsal azt adtuk meg, hogy a dog fogalomnak egy kibővítése, általánosítása az animal fogalom. A szinonimák megadása is hasonló módon végezhető el:

   ```
   EXEC CTX_THES.CREATE_RELATION('sajattz','dog','SYN','pet');
   ```

9. Az elkészült tezaurusz tartalmát szöveges állományba exportálhatjuk a

   ```
   CTXLOAD -USER nev/pwd -THESDUMP -NAME sajattz -FILE ki.txt
   ```

   operációs rendszerbeli paranccsal. A tezauruszon alapuló keresésre lehet példa az animal fogalom specializációit tartalmazó dokumentumok lekérdezése:

   ```
   SELECT szoveg FROM doksik WHERE
     CONTAINS(szoveg,'NT(ANIMAL,3,SAJATTZ)') > 0;
   ```

   illetve a dog és a vele szinonim szavakat tartalmazó dokumentumokat visszaadó lekérdezés:

   ```
   SELECT szoveg FROM doksik WHERE
     CONTAINS(szoveg,'SYN(DOG,SAJATTZ)') > 0;
   ```

   A mintapéldában angol szavak szerepeltek, mivel a rendszer nem rendelkezik a magyar nyelvtan szabályait figyelembe vevő szóillesztési mechanizmussal.