Sukijan asennus- ja käyttöohje

Sukija on Javalla kirjoitettu ohjelma suomenkielisten tekstien indeksointiin.

Sukija analysoi sanat morfologisesti, muuttaa sanat perusmuotoon (joka on sanakirjoissa) ja indeksoi perusmuodot, jotta sanan kaikki taivutusmuodot löytyvät vain perusmuotoa etsimällä.

Sukija tallentaa perusmuodot Solr:n tietokantaan, josta niitä voi etsiä Solr:n käyttöliittymän kautta.

Sukija osaa indeksoida kaikkia niitä tiedostomuotoja, joita Apache Tika (http://tika.apache.org/) osaa lukea.

Mitä tarvitaan ja mistä ne saa?

- Sukija
  https://github.com/ahomansikka/sukija
  Koska luet tätä tekstiä, olet jo imuroinut tämän. (-:
- Suomi-Malaga
  Se on corevoikossa (https://github.com/voikko/corevoikko) hakemistossa suomimalaga.
- Apache Solr 3.6.1
  http://lucene.apache.org/solr/
- Ubuntun paketit libmalaga7 ja maven.

Lisäksi Sukija tarvitsee erinäisiä jar-tiedostoja, mutta Maven imuroi ne verkosta automaagisesti.

Sukijaa voi käyttää myös Voikon Java-version kanssa. Tällöin tarvitaan myös Ubuntun paketti libvoikko1.

Tämä asennusohje olettaaa, että corevoikko ja apache-solr ovat hakemistoissa
`$HOME/Lataukset/corevoikko/` ja
`$HOME/Lataukset/apache-solr-3.6.1`

Ohjelman rakenne

Sukijassa on neljä osaa:

- sukija-core Java-luokkia, joita muut ohjelman osat tarvitsevat.
- sukija-indexer Ohjelma, joka lukee sanoja tiedostoista ja lähettää ne Solr:iin indeksoitavaksi.
- sukija-malaga Solr:n liitännäinen, joka käyttää Suomi-Malagan Sukija-versiota muuttamaan sanat perusmuotoon.
- sukija-voikko Solr:n liitännäinen, joka käyttää Voikkoa muuttamaan sanat perusmuotoon.

Suomi-Malagan asentaminen

Suomi-Malagasta on kaksi versiota, Voikko-versio on tarkoitettu oikolukuun ja Sukija tiedostojen indeksointiin. Sukija-versio käännetään komennolla

```
cd $HOME/Lataukset/corevoikko/suomimalaga
make sukija
```

Tee alihakemisto `$HOME/.sukija` ja kopioi sinne tiedostot
`suomimalaga/sukija/{suomi.*_l,suomi.pro}`

Myös Voikko-versiota voi käyttää indeksointiin, kun sen kääntää ja asentaa komennoilla

```
cd $HOME/Lataukset/corevoikko/suomimalaga
make voikko-sukija
make voikko-install DESTDIR=~/.voikko
```

`DESTDIR` voi olla myös joitan muuta kuin `~/.voikko`.

Sukijan kääntäminen ja asentaminen, Solr:n konfigurointi

Ensin käännetään sukija komennolla

`mvn package`

Komento imuroi netistä tarvitsemansa Javan jar-paketit eli ensimmäinen kääntäminen saattaa kestää kauan. Erityisen kauan se kestää, jos et ole aiemmin käyttänyt mavenia.

Toisessa vaiheessa asetetaan Solr:n konfigurointitiedostoon `schema.xml` saneistajaluokka (ulkomaankielellä tokenizer), joka lukee sanat tiedostoista, ja morfologialuokka, joka muuttaa sanat perusmuotoon. Komennolla `make ____-schema` on viisi eri vaihtoehtoa:

Komento	Morfologialuokka
`make malaga-schema`	MalagaMorphologyFilterFactory
`make malaga-suggestion-schema`	MalagaMorphologySuggestionFilterFactory
`make voikko-schema`	VoikkoMorphologyFilterFactory
`make voikko-suggestion-schema`	VoikkoMorphologySuggestionFilterFactory
`make debug-schema`	

Komentoa `make debug-schema` käytetään vain Sukijan kehittämiseen.

Saneistajan oletusarvo on FinnishTokenizerFactory, joka tulee Sukijan mukana, mutta sen voi vaihtaa muuttujalla TOKENIZER_FACTORY esimerkiksi näin:

```
make voikko-schema TOKENIZER_FACTORY=solr.StandardTokenizerFactory
```

`____FilterFactory` ja `____SuggestionFilterFactory` eroavat toisistaan siten, että jos morfologialuokka ei tunnista sanaa, `Suggestion`-luokissa sanaan tehdään muutoksia (esimerkiksi muutetaan w v:ksi) ja tunnistusta yritetään uudestaan. Tämä ei ole sama asia kuin Voikon oikeinkirjoituksen korjausehdotukset!

`Suggestion`-luokat konfiguroidaan tiedostossa `suggestion.txt`. Katso sivu 4.

Kolmas ja viimeinen komento on `make install`, joka kopioi hakemistossa `conf` olevat Solr:n ja Sukijan tarvitsemat tiedostot oikeisiin paikkoihin.

Tiedostot `logging.properties`, `suggestion.txt` ja `synonyms.txt` kopioidaan hakemistoon `$HOME/.sukija` ja tiedostot `schema.xml`, `solrconfig.xml`, `sukija-context.xml` ja `sukija.xsl` niihin hakemistoihin, joista Solr lukee ne.

Sukijan konfigurointi

Sukija konfiguroidaan kahden Javan ympäristömuuttujan avulla. Esimerkiksi

```
sukija.ignore.files = (?u)(?i).*[.](jpg|jpeg)$
sukija.solr.url = http://localhost:8983/solr
```

`sukija.ignore.files` on säännöllinen lauseke, joka kuvaa ne tiedostot, joita ei indeksoida. Yllä olevassa esimerkissä ei indeksoida JPG-tiedostoja. Säännöllisen lausekkeen alku `(?u)(?i)` kertoo, että tiedostojen nimissä voi olla Unicode-merkkejä ja että kirjaimen koolla (isot tai pienet) ei ole väliä.

`sukija.solr.url` on Solr:n verkko-osoite.

Näiden muuttujien oletusarvot ovat tiedostossa `Indexer.properties`, ja niitä voidaan muuttaa Javan argumentilla `-D`.

Varsinainen indeksointi konfiguroidaan sitten Solr:n kautta. Valitettavasti Solr:n konfigurointi on oma taiteenlajinsa :-).

Sukijan käyttö

Ensin pitää käynnistää Solr:

```
cd $HOME/Lataukset/apache-solr-3.6.1/example
java -jar example.jar
```

Jos Solr valittaa jna:sta, käynnistyskomento on

```
java -Djna.nosys=true -jar example.jar
```

Solr:n lokia voi konfiguroida lisäämällä Solr:n käynnistykseen:

```
-Djava.util.logging.config.file=~/.sukija/logging.properties
```

Myös Javan lokin (java.util.logging) konfigurointi on oma taiteenlajinsa, ja jotta asia ei olisi liian yksinkertainen, Solr ja niin ollen myös Sukijan Solr-liitännäiset käytävät SLF4J:tä (www.slf4j.org), joka puolestaan käyttää Javan lokisysteemiä.

Kun Solr on saatu käyntiin, voidaan ruveta indeksoimaan! Sitä varten Sukijassa on bash-ohjelma, joka käynnistetään näin:

```
./sukija.sh tiedosto ...
```

Ohjelma asettaa Javan classpath-muuttujan ja rupeaa sitten indeksoimaan. Jos tiedosto onkin hakemisto, indeksoidaan kaikki hakemiston ja sen alihakemistojen tiedostot.

Jos classpath on asetettu jollain muulla tavalla, indeksoinnin voi käynnistää näin:

```
java peltomaa.sukija.indexer.Indexer tiedosto ...
```

Jos muuttujien `sukija.ignore.files` ja `sukija.solr.url` oletusarvot eivät kelpaa, ne voidaan asettaa näin:

```
java -Dsukija.ignore.files=... peltomaa.sukija.indexer.Indexer
```

Tiedoston suggestion.txt konfigurointi

Tiedosto `suggestion.txt` pitää konfiguroida erikseen Sukijalle ja Voikolle. Nykyinen konfiguraatio on tehty Sukijalle.

Konfiguraatiossa on neljä komentoa.

`Apostrophe` Poistaa sanasta heittomerkin ja yrittää tunnistaa sanan sen jälkeen. Jos tunnistaminen ei onnistu, poistaa sanasta heittomerkin ja kaikki sen jälkeiset merkit ja palauttaa jäljelle jääneet merkit sanan perusmuotona. Esimerkiksi `Bordeaux'iin` yritetään tunnistaa muodossa `Bordeauxiin`. Jos sitä ei tunnisteta, palauttaa merkkijonon `Bordeaux`.

`Char` Muuttaa sanassa olevan yhden merkin toiseksi. Esimerkiksi

`Char "w" "v"`

muuttaa w:n v:ksi (”wanha” => ”vanha”).

`CharCombination` Muuttaa yhden tai usemman merkin kaikki kombinaatiot. Esimerkiksi

`CharCombination "pt" "bd"`

(1) muuttaa p:t b:iksi, jättää t:t ennalleen, (2) muuttaa t:t d:iksi, jättää p:t ennalleen, sekä (3) muuttaa p:t b:iksi ja d:t t:iksi.

`Length3` poistaa kolmesta peräkkäisestä samasta kirjaimesta yhden. Esimerkiksi ”kauttta” => ”kautta”.

`Regex` muuttaa säännöllisen lausekkeen. Esimerkiksi

`Regex "ai(j)[eou]" ""`

poistaa j-kirjaimen muun muassa sanoista ”aijemmin”, ”aijomme” ja ”kaijutin”.

Säännöllisessä lausekkeessa voi käyttää kirjainta `C` tarkoittamaan konsonantteja ja kirjainta `V` tarkoittamaan vokaaleja. Esimerkiksi

`Regex  "C[ae](hi)C"  "i"`

poistaa h-kirjaimet muun muassa sanoista ”ainahinen” ja ”etehinen”.

Näitä komentoja voi antaa mielivaltaisen paljon missä tahansa järjestyksessä, ja `____SuggestionFilterFactory` palauttaa perusmuotona ensimmäisen tunnistamansa muodon. Jos mitään ehdotusta ei tunnisteta, `____SuggestionFilterFactory` palauttaa alkuperäisen merkkijonon.

Tiedostossa `suggestion.txt` voi olla tyhjiä rivejä. Kommentti alkaa merkillä `#` ja jatkuu rivin loppuun.