CSE 435/535
Information Retrieval

Recitation 2 : Customizing Solr

Agenda

- Moving beyond schema less mode:
  - Defining a new field
  - Setting up copy fields
  - Understanding field analysis
- Other features
  - Language identification
  - De-duplication

Use case

- At the end of whats e covered in class, we have a Solr core running as “gettingstarted”
- We have two types of requirements now:
  - A. Rename fields as needed (lang to tweet_lang for example) that are being analyzed as needed
  - B. Define new fields with modified analysis chains (text_ko, text_tr, etc.)
- So let’s go ahead and do that!
- Note : Simply showing multiple ways of achieving the same thing!

Use case 1 : Rename field + using same instance

- If we use the same instance, we can achieve a rename in two ways:
  - Define a new field and rename data input (search / replace)
  - Define a new field and set it up as a copy
  - Side note : why does pushing the same data multiple times work? Why does it not create multiple copies?

Use case 1 : Copy fields

- Navigate to the schema page (Solr dashboard > Core selector - gettingstarted > Schema)
- Click on “Add Field” and fill in required values:
    - name : tweet_lang, field_type : string, etc.
    - Click on “Add Field”
- Click on “Add Copy Field”
    - source: lang, destination : tweet_lang
    - Click on “Add Copy Field”
- Now simply re-index and the data would be populated!

Use case 2 : Define new fields

- You can do one amongst two things here : modify the schema on the existing core OR define a new core with the modified schema

- Let's do the first one here and show show the latter can be done

- Stop your solr instance and find the file called managed_schema ($SOLR_HOME/example/schemaless/solr/gettingstarted/conf/)

- Rename it to schema.xml

Use case 2 : Example field (text_es)

- For the text_es field, we need to define a new analysis chain
  - All of the existing chains are set up for English.
  - We need a new analysis mechanism for spanish text that uses customized spanish analysis
  - We have a defined type present, text_es and we will use it here.
  - We could make some changes in the pre-defined type, save the file and reload the core
  - Solr will parse the file, affect changes if any and rename the file back to managed_schema

Use case 2 : Defining a new core

- Currently the solr_home is set to $SOLR_HOME/example/schemaless/solr
- If you start Solr without specifying the -e flag, it will look under $SOLR_HOME/server/solr
- To define a new core, say IRF16P1, under solr_home create the following directories:
    - IRF16P1, IRF16P1/conf, IRF16P1/data
- Copy all files and subdirectories from solr_home/gettingstarted/conf to solr_home/IRF16P1/conf/.
- Rename managed_schema to schema.xml and make your changes as before.
- Navigate to Core Admin (Solr dashboard > Core Admin)
    - Click on “Add Core”
    - Set both name and instanceDir to IRF16P1 and click on “Add Core”.
- You can now push data to this core with -c IRF16P1 flag.

Other features : LI

- Language identification : Solr can guess the language based upon a given field and take actions
  - So you could ignore the lang provided by Twitter
  - Set up lang identification to work off of “text” field
  - Define mapped field as “text” => will copy data to text_en, text_es, text_ko, etc. based on determined language
- Remember : we cannot and will not validate if the identified language is correct, so this could be a neat way to handle language specific changes

Other features : Deduplication

- Set up solr to identify duplicates off of text field
- If multiple users simply re-tweet a given tweet, most likely the text would be the same
- Can automatically filter out re-tweets and duplicate tweets