CSE 435/535 Information Retrieval
Fall 2015
Project : A multilingual search system for tweets

Due Date: September 24 (Part A)

Overview:

The goal of this project is to gain hands-on experience in building a complete search solution using Solr (http://lucene.apache.org/solr/). The focus will be on ingesting, indexing and searching twitter (www.twitter.com) data in three different languages, reflecting multiple topics. The system will be built in three stages consisting of: (A) ingesting content and indexing the data using Solr, (B) experimenting with various retrieval models that perform well on the target set of queries, and (C) extending the basic search system by tagging the content with named entities, corresponding Wikipedia links, etc. By the end of the semester, you would have developed a complete search-based solution for multilingual twitter data. Some of the final projects will be selected for in-class presentation. Though the emphasis is more on the appropriate configuration and use of existing tools to develop a complete solution, some programming is required.

Part A

The first part of the project involves getting familiar with both Solr and the twitter API. The focus here is on tokenization and indexing, requiring you to configure Solr based on class lectures. Even though a basic Solr system is easy to setup, some work will be required in figuring out optimal configurations for both the nature and language of the data being indexed.

Part A will be individual projects; the subsequent projects will involve teams. We will use a cloud based VM hosted on Koding.com for setting up Solr. More details will be provided soon.

The following sections describe the various tasks involved, evaluation criteria and submission guidelines.

Project tasks

Overall, you are required to perform two tasks as enumerated below:

1. Crawl twitter: The first task involves collecting topic specific tweets subject to the requirements below:
   - The tweets languages should be restricted to English, Russian and German.
   - For the total tweets collected, the volume for any language should not be less than 25%.
   - You should collect at least 2,000 tweets with data collected over at least 5 different days. Ideally, you should crawl about a 100 tweets minimum on a daily basis.
   - The tweets should focus on one of five topics. Refer the table below to find which topic you need to crawl.

UBID ends in	Topic
0 or 1	Entertainment e.g. Movies, TV shows, celebrity news, Books, Theatre etc.
2 or 3	Sports e.g. sporting event results, team changes / standings, etc.
4 or 5	Medicine, health e.g. epidemics, drug trials, disease outbreaks, health education, etc.

6 0r 7	Politics e.g. elections, political rallies, policy changes, government, etc.
8 or 9	Technology e.g. product reviews & launches, scientific discoveries, patent filings, etc.

Some helpful tips:

- You will need to figure out the twitter public API with regards to how to query for keywords and detected language.
- You will need to find some keywords for your designated category that are either being used across languages (something like a global event - US Open, VMAs, Apple keynote, Republican primaries, Ebola outbreak etc.) or specific keywords for each language.
- One of the easiest ways to find keywords is through news and trending topics.
- Take time in looking at the data returned by the API call. Understanding all fields in the returned response will help you in the next step.
- You might want to keep a copy of all your data. The next step may involve several iterations and team based integration later may not be straightforward.

2. Index tweets using Solr: The second task involves indexing the crawled tweets using Solr. A separate class will cover setting up Solr and additional documentation will guide you through configuring Solr on a cloud VM. This task mainly involves designing a Solr schema to index the tweets you have collected. Some aspects you should consider are as follows:
    - **Fields**: What fields would constitute a document? What are their data types? What about other attributes at a field level?
    - **Multilingual**: How will the different languages be handled? Will you create separate cores or separate fields?
    - **Analyzers and tokenizers**: How will the different fields be analyzed? How will you handle special tokens like hashtags, usernames, URLs and emoticons? What about spelling errors?
    - **Other features**: Solr supports many additional different features that may or may not be useful for your project. What other features will you use and why?

    Apart from the above you should also spend some time analyzing your index. You should be able to explain your index size, top terms etc.

Evaluation:

You will be asked to run some statistics on the indexes that your system produces. Examples of such statistics include the number of documents indexed, the top k terms, etc. You will NOT be asked to implement retrieval models at this stage.

What to submit:

You should plan on submitting all configuration files (schema.xml and solrconfig.xml) for all of your Solr cores using cse-submit script. More details will be provided in class on this.