Homework: Building an Apache-Solr / ElasticSearch based Search Engine, Ranking Algorithms and NER for Weapons Datasets
Due: November 6, 2015, 12pm PT

---

1. Overview

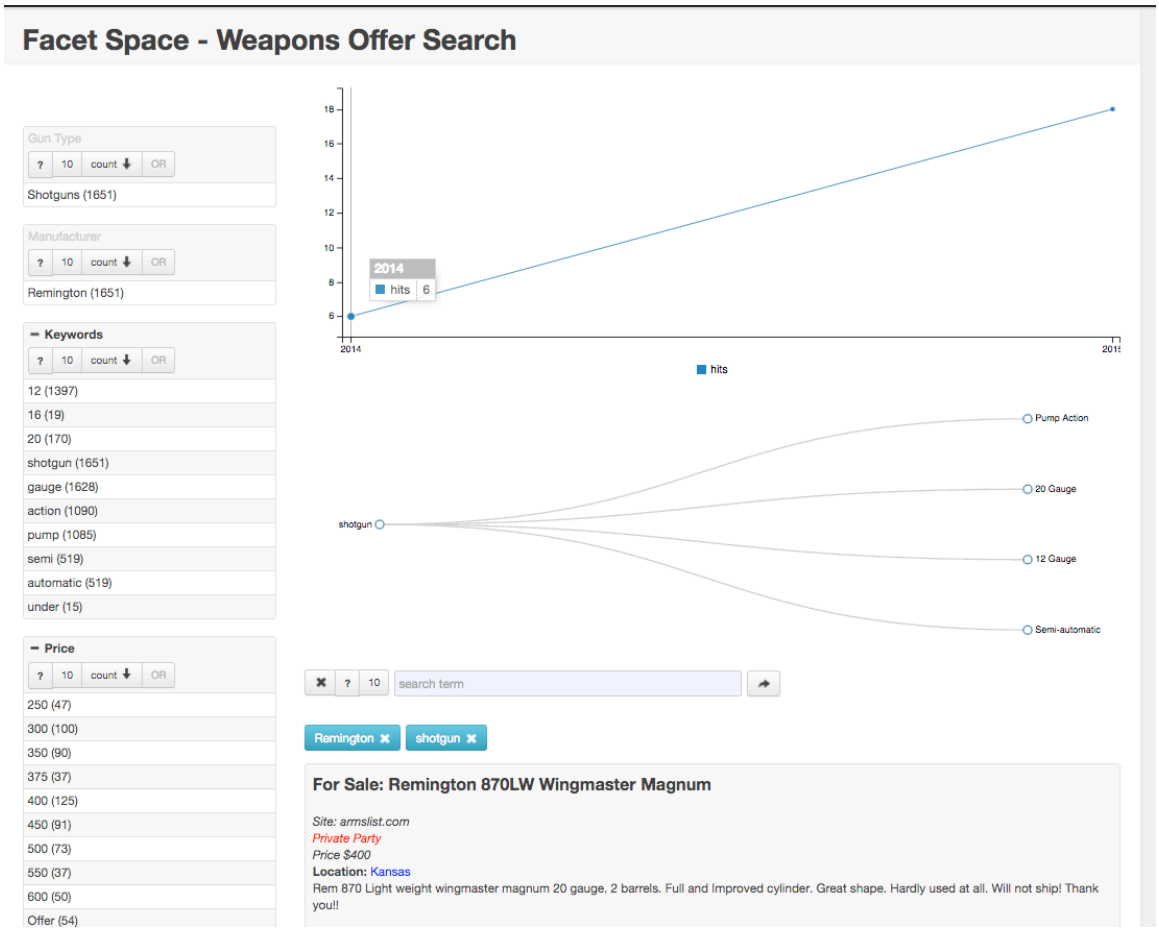

Figure 1. Weapons data search against a hypothetical index that you will create.

In this assignment you will build upon your first homework and develop capabilities including ranking algorithms and Named Entity Extraction (NER) for your crawled weapons datasets. You will develop and compare two sets of ranking and retrieval approaches: *content-based* approaches that will leverage the indexed textual content of your data and summary metrics using IR techniques such as term frequency-inverse document frequency (TF-IDF) to generate relevancy; and *link-based* approaches that will leverage citation relationships (graph-based) between the indexed documents and information other than the textual content of the document to perform relevancy analysis. In addition we will augment and analyze the weapons data through geospatial Named Entity Recognition along with extraction of measurements, numbers, and units from the data.

The assignment will help you answer relevant scientific questions related to geospatial properties; measurements present in the file; citation and information source for the data; and trends and topics over time for Weapons. With the current strife related to gun issues in the U.S. the analysis you do will be timely and help better inform you towards this national issue vis-à-vis the data you have collected regarding weapons.

2. Objective

The objective of this assignment is to develop a notion of *relevancy* related to the weapons datasets that you crawled and obtained in assignment #1. Please note. **You are not required to crawl for this assignment**. Instead, you will leverage the data that you obtained in assignment #1. If you do not have your data any more, please notify the Graders, TAs and Professor and be prepared to provide a reason as to why you don't have it since the Professor has mentioned several times to date to **keep your data**.

As we have learned in class, relevancy can be measured in Information Retrieval via several methodologies; the first generation of relevancy was related to text-based summarization and we have discussed techniques such as TF-IDF as a metric in these means. With the advent of Hypertext Induced Topic Search (HITS) and more-so with the PageRank algorithm developed by Brin and Page, we have also seen a notion of *link-based* relevancy that does not rely on summarized text, but instead graph-based relationships between documents indicating their associated relevancy to one another. We will build on these two notions in this assignment.

You will deploy the Apache Solr (http://lucene.apache.org/solr/) full text indexing system or the ElasticSearch indexing system (https://www.elastic.co/) that are both built on top of the Apache Lucene search engine (http://lucene.apache.org/). Solr, ElasticSearch and Lucene fully implement an inverted index and the vector space retrieval model, and provide a basis on which to develop relevancy mechanisms via text or otherwise. You can leverage Solr's native integration with Apache Tika in this assignment via the *ExtractingRequestHandler* plugin in Solr (note ElasticSearch's integration with Tika is more obtuse, so if you choose ElasticSearch you will have to develop your own annotation technique using Tika-Python). Solr's integration allows you to directly post documents to Solr and run Tika on the server side to extract metadata and to index it. You will also leverage your Nutch crawl data in this assignment and Nutch's direct integration with Solr and ElasticSearch via Nutch's indexing plugin to use and compare your already extracted Tika-based metadata and text to that provided by Solr's server side ExtractingRequestHandler and/or Elastic Search's indexed data.

While building your index you will make careful considerations based on the data model of your weapons data as to what fields are important for relevancy measurement. Is *timestamp* an important field? What about image metadata such as Color Component Quantization? In HTML pages, what metadata is available to you to relate to other documents? Can you identify specific parties that are selling weapons (Private; Business, etc.)? A set of challenge questions identified in Section 3 will frame your analysis.

3. Tasks

1. Develop an indexing system using Apache Solr and its ExtractingRequestHandler ("SolrCell") or using Elastic Search and Tika-Python
    a. Install Solr from the lucene_4_10 branch (instructions in Section 4)
        i. http://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_10/
    b. If you picked ElasticSearch install per this guide: https://www.elastic.co/guide/en/elasticsearch/guide/current/_installing_elasticsearch.html
    c. Upgrade SolrCell similarly to what you did in assignment #1
        i. Build Tika trunk (1.11-SNAPSHOT) with the following support
        ii. GeoTopic parsing
            1. Make sure you build Tika with GeoTopic parsing support per: http://wiki.apache.org/tika/GeoTopicParser
        iii. OCR
            1. Make sure you build Tika with OCR support per: http://wiki.apache.org/tika/TikaOCR/
        iv. cTAKES / UIMA
            1. Build Tika with cTAKES / UIMA support per: http://wiki.apache.org/tika/cTAKESParser
    d. Your system should take, as input, the original weapons data you crawled from assignment #1. Since you will be performing direct post of this data (without Nutch in this step), you must first using the ./bin/nutch dump command to export your data before posting it in to Solr or ElasticSearch.
    e. If you chose ElasticSearch, develop a program in Python that uses Tika-Python to index your metadata and data into ElasticSearch. Make sure you do this after 1c. is complete.
2. Leverage the Nutch indexing system to build up an Apache Solr index or an ElasticSearch index
    a. Upgrade your Tika instance from assignment #1 to include the content parser support mentioned in 1.c.ii, 1.c.iii and 1.c.iv.
    b. Compare the metadata extracted from using Tika in Nutch during crawling upstream to your SolrCell or ElasticSearch based Tika run generated in Task #1.
        i. What did Tika extract in Nutch compared to what SolrCell or ElasticSearch extracts?
3. Design and implement two ranking algorithms for your Weapons data documents
    a. A *content-based* algorithm that uses text retrieval methods including TFIDF to identify the relevancy of each document in your index. The algorithm should use the text stored in Solr/ElasticSearch fields (provided either by SolrCell/ElasticSearch and/or by Nutch and its use of Tika) to assess the relevancy of each document to a given query from the examples outlined in in Task #4.

b. A *link-based* algorithm that uses the relationships between your associated documents (the *metadata features*) to identify relevancy of the documents independent of the user's query. Link-based metrics compute relevancy based on properties such as geographic location (all images related to e.g., Rifles in Texas); temporal properties (all images and pages from September 2015 or from 2014); and other relevant features. You are free to use the extracted features from Tika along with GeoTopicParser and cTAKES/UIMA to achieve this. You may also look at other NER technologies, for example Memex-GATE and/or Behemoth:
    i. https://github.com/memex-explorer/memex-gate
    ii. https://github.com/DigitalPebble/behemoth
    iii. You will need to develop a program in Python, and/or Java and/or Bash that properly augments your Solr documents with the **tagged results** of your link-based relevancy algorithm.

4. Develop a suite of queries that demonstrate answers to the relevant weapons related questions below.
    a. What time-based trends exist in Gun ads? Can you correlate temporal and spatial properties with buyers? For example can you identify based on ad time-window and/or based on geospatial area places where people try and purchase guns on behalf of someone unauthorized to purchase them?
    b. Can you identify similar firearms image types (e.g., shotguns) that are sold in the same region and time? Does this indicate influx related to stolen goods?
    c. When a shipment of bulk firearms is stolen, the rate of ads and images may indicate an increase in sales of that particular make/model – can you identify these?
    d. Can you identify ads and/or weapons images that are posted by persons, whom are underage or in which the weapons are de-identified (by type and/or serial number, etc.)
    e. Can you identify ads and/or images that relate to the unlawful transfer, sale, and possession of explosives, WMD devices, and precursors ?
5. Develop a program in Python, Java, and/or Bash that runs your queries against your Solr or ElasticSearch index and outputs the results in an easy to read list of results demonstrating your relevancy algorithms and answers to your challenge questions from Task #4.
6. (Extra Credit) Develop a Lucene-latent Dirichlet allocation (LDA) technique for topic modeling on your index and use it to rank and return documents.
    a. Re-run your queries and examine the results from Task #4. What differences do you see? Can you explain them?
    b. Use: https://github.com/chrismattmann/lucene-lda/
7. (Extra Credit) Figure out how to integrate your relevancy algorithms into Nutch.
    a. Nutch has a scoring interface: http://nutch.apache.org/apidocs/apidocs-1.10/org/apache/nutch/scoring/ScoringFilter.html
8. (Extra Credit) Create a D3-based visualization of your link-based relevancy. Provide a capability to generate D3 relevancy visualizations as a Nutch REST service using Apache CXF. Integrate the service into nutch-python.

4. Assignment Setup

4.1 Group Formation

Please keep the same groups as for your assignment #1. If you have any questions please contact:

Mohit Bagde
bagde@usc.edu

Divydeep Agarwal
divydeea@usc.edu

Komal Dhawan
komaldha@usc.edu

Use subject: CS 572: Team Details

4.2 Dataset

Please start with your data that you have prepared in assignment #1. If you would like additional data from Amazon S3, please let the graders know and we will coordinate its delivery via amazon S3 buckets and read-only keys to **one member of your group**.

4.3 Installing and Building Apache Solr

You will need to build Apache Solr from the 4_10 branch of lucene-solr to take advantage of a fix that the Professor provided for integrating Tika with SolrCell and OCR:

http://svn.apache.org/repos/asf/lucene/dev/branches/lucene_solr_4_10/

You can find more information here:
https://issues.apache.org/jira/browse/SOLR-7137
https://issues.apache.org/jira/browse/SOLR-7139

Apache Solr comes with a web application server (Jetty), or you can also deploy and configure Solr with Apache Tomcat. Either way will work fine for this assignment and the instructions are provided here:

https://cwiki.apache.org/confluence/display/solr/Running+Solr+on+Jetty
https://cwiki.apache.org/confluence/display/solr/Running+Solr+on+Tomcat

You should also review the basic installation instructions:

http://wiki.apache.org/solr/SolrInstall

And the instructions for using SolrCell/ExtractingRequestHandler:

https://wiki.apache.org/solr/ExtractingRequestHandler

Once installed, you will need to configure Solr to accept your Weapons data model. You will also be responsible for integrating your ranking algorithms into Solr and for using Solr to query and answer your challenge questions.

Please review Solr function query documentation:
http://wiki.apache.org/solr/FunctionQuery

Your ranking algorithms can either be interactive (per query), or also on document index-time ranking. This is something you will need to figure out as a group based on the type of ranking algorithm you are developing.

For ElasticSearch, see the relevant documentation on the ElasticSearch site linked above.

4.4 Upgrading Tika with OCR, GeoTopicParser and cTAKES

Please follow similar instructions as described in assignment #1 and via the links above in Task #1c.

4.5 Indexing with Nutch

Please see: https://wiki.apache.org/nutch/bin/nutch%20solrindex for some guidance on how to index from Nutch into Solr. Please use your Nutch 1.11-trunk deployment from assignment #1 for this step.

4.6 Describing your Relevancy Algorithms and your Weapons Challenge Questions that they Answer

You will need to describe your ranking algorithms formally in your report in addition to implementing them. Please see:

https://en.wikipedia.org/wiki/PageRank

For some guidance on describing your algorithms we suggest identifying as part of your description highlight which fields from the two approaches for indexing: (1) Solr index with SolrCell or ElasticSearch; and (2) Nutch/Tika and indexed into Solr/ElasticSearch you are leveraging in each of your relevancy algorithms, and also identifying which algorithms are more appropriate to answer each scientific question and why.

5. Report

Write a short 4 page report describing your observations. Please answer how effective the link-based algorithm was compared to the content-based ranking algorithm in light of these weapons challenge questions? What questions were more appropriate for the link based algorithm compared to the content one? Describe in detail and formally both of your ranking algorithms. You should describe the input, what your algorithms do to compute a rank, how to test them (and prove that they are working as expected). Do NOT simply provide advantages and disadvantages from a quick Google search. You are required to think critically about this portion of the report and sincerely provide your feedback.

Describe the indexing process – what was easier – Nutch/Tika + SolrIndexing; or SolrCell or ElasticSearch?
Again a quick Google search will not suffice here. We are looking for your own words.

Please also note that the graders will be given great flexibility to add/subtract from various areas of the report based on your submitted work.

6. Submission Guidelines

This assignment is to be submitted ***electronically, by 12pm PT*** on the specified due date, via Gmail csci572fall2015@gmail.com. Use the subject line: CSCI 572: Mattmann: Fall 2015: Solr Homework: <Your Lastname>: <Your Firstname>. So if your name was Lord Voldemort, you would submit an email to csci572fall2015@gmail.com with the subject "CSCI 572: Mattmann: Fall 2015: Solr Homework: Voldemort: Lord" (no quotes). **Please note only one submission per team**.

- All source code is expected to be commented, to compile, and to run. You should have a program developed in step #3b to annotate your Solr documents with your link based relevancy, and at a minimum should also have a program developed in step #5 that demonstrates your queries and the result sets they obtain.
- Deliver your Solr configuration e.g., your `schema.xml and solrconfig.xml` files and any other configuration necessary to reproduce your results. Also deliver your elasticsearch mappings schema JSON file if you used elasticsearch.
- Teams will be asked if they would like to contribute their indexed dataset to our Amazon machine for inclusion in Memex and/or TREC. This would include your Solr index or your Elastic search\. If you want to do this, please identify in your report that you would like to do this, and send a message to the professor, to the TAs, and to the graders.
- Also prepare a `readme.txt` containing any notes you'd like to submit.
- If you have used any external libraries in your two programs, you should include those jar files in your submission, and include in your `readme.txt` a detailed explanation of how to use these libraries when compiling and executing your program.

- Save your report as a PDF file (`Lastname_Firstname_SOLR.pdf`) and include it in your submission.
- Compress all of the above into a single zip archive and name it according to the following filename convention:

  <lastname>_<firstname>_CSCI572_HW_SOLR.zip

  Use only standard zip format. Do **not** use other formats such as zipx, rar, ace, etc.

Important Note:

- Make sure that you have attached the file when submitting. Failure to do so will be treated as non-submission.
- Successful submission will be indicated in the assignment's submission history. We advise that you check to verify the timestamp, and download and double check your zip file for good measure.
- Again, please note, only **one submission per team**. Designate someone to submit.

6.1 Late Assignment Policy

- -10% if submitted within the first 24 hours
- -15% for each additional 24 hours or part thereof