Girish Projects

Girish worked on below projects and the source code located is also below.

Whitelisting/Name Matching Project - quine.algorithm.cs.sunysb.edu:/whitelisting/
Foreign Entities - quine.algorithm.cs.sunysb.edu:/foreign/

Steps for doing Name matching project are below.

http://hivelogic.com/enkoder/form

FREEBASE
1. Download the latest freebase dump name "http://download.freebase.com/datadumps" and the in the folder with the latest date download freebase-simple-topic-dump.tsv.bz2 and bunzip it and untar it.

cut -f2 ~/freebase/freebase-simple-topic-dump.tsv >freebase.txt run through unique to get the unique entities.

egrep -v
"^(January|Jan|Feb|February|March|Mar|April|Apr|May|June|Jun|July|Jul|August|Aug|September|Sep|October|Oct|November|Nov|December|Dec) [0-9]+$" freebase.txt > freebase_v2.txt

mv freebase_v2.txt freebase.txt

1. Remove the || from the freebase.txt as we use that as delimiter.
egrep -v "(\|\|)" freebase.txt>freebase_v2.txt

mv freebase_v2.txt freebase.txt
python convert_tomap.py   freebase.txt freebase.map 1

LYDIA
lydia_latest_stripped_1m.txt - contains the file before map
Once you get the lydia file from the hdfs - You need to run a script to remove all unwanted categories.
./cleanup_lydia_latest.py
./convert_tomap.py lydia.txt lydia_add.map 2

Comment : lydia.txt should contain the frequency entity category
cat freebase.map lydia_add.map > input_new.map
Comment: Optional use the wiki redirects also can be included, with freebase .. make then

unique. Use script - get_redirects and append using below command
python get_redirects.py redirect_links.txt redirect.map
cat input_new.map redirect.map > input.map

MAIN MAPPER/REDUCER
cat input.map | python mapper_freebase_v2.py | LC_COLLATE=C sort -k1,1 | python lydia_only_reducer.py

sort -nr -k1 lydia_only.txt >sorted_lydia_only.txt
sort -nr -k1 common.txt >sorted_common.txt
sort -nr -k1 freebase_only.txt > sorted_freebase_only.txt

LOCATION OF DATA :/home/freedonia/freebase_exp/final/New_list/BrandNew/mapreduce

442548
475938
455645 - with redirects
429374 - latest after post processing..

LYDIA ONLY POST PROCESSING:
1. sort -nr -k1 lydia_only.txt > sorted_lydia_only.txt - duplicate
2. Remove all lines with hyphens, * and / - can be moved to pre-processing
egrep -v "(\-|*|\/)" sorted_lydia_only.txt > sorted_lydia_only_v2.txt
3.

COMMON POST PROCESSING
Common file and lydia file post processing with wikipedia redirects.
collapse_common.py : This currently give all the collapsed entities that are in wiki-redirects only. The ones which are not found in redirects and in common needs to be included.
collapse_common.py redirects_file (raw what Bala gave me) un/sorted_common.txt lydia_add.map(for frequency)

python collapse_common.py name_redirect_links_jan2.txt sorted_common.txt lydia_add.map

bmudiam@quine:/whitelisting$ wc -l lydia_only.txt common.txt freebase_only.txt sorted_lydia_only_v2.txt collapsed_common.txt
455645 lydia_only.txt
401675 common.txt

11920669 freebase_only.txt
429374 sorted_lydia_only_v2.txt
338653 collapsed_common.txt
13546016 total

FREEBASE ONLY POST PROCESSING

COUNTS FOR FREEBASE DATA:
You need to get the count from the text corpus. The location for the script.
/home/freedonia/freebase_exp/Raw-Data/BrandNew
file : compare_with_corpus_devel.py
inputs : python compare_with_corpus_devel.py file_list.txt freebase_only.txt outfile.txt
hard-coded file name : search_list_new.txt (should contain freebase text that needs to be searched.). For this we need to extract all the entity names (exact) from freebase_only.txt . Entites between { }an delimited by ||

All the Python code:
canonical_restore_s.py - restores entities into canonical format.
get_redirects.py - gets all the redirects in freebase format
collapse_common.py - Collapse common with wikipedia redirects

lydia_only_reducer.py - This is the main reducer creates file common.txt , lydia_only.txt and freebase_only.txt

restore_s.py - script to restore the plurals stripped in common.txt and lydia_only.txt with the canonical name. This can be merged with lydia_only_reducer.py

common_merge.py - merges , merged_p_v2.txt (person and unknown merge) and wikipedia redirects.

mapper_freebase.py - Main mapper which takes both freebase and lydia input in single file. In the format discussed.
strip_merged_people.py - helper function to strip the merged people.
convert_tomap.py - converts the input to the format required for freebase_mapper.py
mapper_freebase_v2.py - Version 2, cleaner and better merging strategy .
total_merge_people.py - Script the collapses
get_redirects_dict.py

reducer.py

Get the top entities from the

cat input_new.map | python mapper_freebase_v2.py | LC_COLLATE=C sort -k1,1 | python lydia_only_reducer.py
http://stackoverflow.com/questions/4273922/unix-sort-to-sort-only-one-column

AGGRESSIVE MERGING

Lydia only person is merged with UNKNOWN from lydia at
freebase_exp/final/New_list/BrandNew/mapreduce/merge_test

merged file is : merged_p_v2.txt
python programs used : extract_entity.py extract_merged_persons.py merger_people.py
total_merge_people.py

Previous Document

Extracting the freebase entities

1. Download the latest freebase dump name "http://download.freebase.com/datadumps" and the in the folder with the latest date download freebase-simple-topic-dump.tsv.bz2 and bunzip it and untar it.

2. ./freebase-extract.sh freebase-simple-topic-dump.tsv

The output of this is temp.txt with the entities, frequency and category.

Getting only the words in both the files.

cut -f2 lydia.txt | sort | uniq > lydia_sort_1.txt
cut -f1 freebase.txt | sort | uniq > freebase_sort_1.txt

Print the common words

./comm -i -12 lydia_sort_1.txt freebase_sort_1.txt > Common.txt - common lines between words
./comm -i -23 lydia_sort_1.txt freebase_sort_1.txt > lydia_only.txt - lydia only words
./comm -i -13 lydia_sort_1.txt freebase_sort_1.txt > freebase_only.txt - free base only words

1. Cleaning up lydia.

* Eliminated dumb categories - address, age, date_period, day, month, month_period, time, time_period, time_zone, week_period, color, points, year_period

```
sed '/\tage\t/d' < Lydia-Temp.txt > Clean1.txt
sort -k2 < Lydia-essentials.txt > Sorted.txt
sed -n '/\tcity$/p' < lydia-only-records.txt > temp

egrep '^[-0-9a-zA-Z ]*$' search_list.txt > temp

egrep "[\']" search_list.txt  | less

 egrep -v
"^(January|Jan|Feb|February|March|Mar|April|Apr|May|June|Jun|July|Jul|August|Aug|September
|Sep|October|Oct|November|Nov|December|Dec) [0-9]+$"  freebase_only.txt >
freebase_only_without_dates.txt
 egrep -v
"(January|Jan|Feb|February|March|Mar|April|Apr|May|June|Jun|July|Jul|August|Aug|September|
Sep|October|Oct|November|Nov|December|Dec) [0-9]+$"  sorted_whitelist_final_output.txt >
sorted_whitelist_final_output_without_dates.txt

egrep  -v  "^The [A-Z][a-z]+$"  search_list_new.txt > search_list_new_without_the_words.txt
```

```
 616  grep "[a-zA-Z0-9]" search_list.txt | less
 617  grep "[^a-zA-Z0-9]" search_list.txt | less
 618  grep "[^a-zA-Z0-9]" search_list.txt | less
 619  grep "[a-z]" search_list.txt | less
 620  grep "[:alnum]" search_list.txt | less
```

```
621  grep "[a-zA-Z0-9]" search_list.txt | less
622  grep "[:alnum:]" search_list.txt | less
623* egrep "[a-b]" search_list.txt | less
624  egrep "[:alnum:]" search_list.txt | less
625  egrep "[a-zA-Z0-9]" search_list.txt | less
626  egrep "[a-zA-Z0-9]" search_list.txt | wc -l
627  wc -l search_list.txt
628  grep "[a-z]*" search_list.txt | less
629  grep "[a-z]* search_list.txt | less
630  grep '[a-z]*' search_list.txt | less
631  grep '^[-0-9a-zA-Z]*$' search_list.txt | less
632  grep '^[-0-9a-zA-Z ]*$' search_list.txt | less
633  grep '^[-0-9a-zA-Z ]*[^-]*$' search_list.txt | less
634  grep '^[-0-9a-zA-Z ]*$' search_list.txt | less
635  grep '^[-0-9a-zA-Z ]*$' search_list.txt | wc -l
636  egrep '^[-0-9a-zA-Z ]*$' search_list.txt | wc -l
637  egrep '^[-0-9a-zA-Z ]*$' search_list.txt > temp
638  mv temp search_list.txt
639  cp search_list.txt list_bck.txt
640  vi search_list.txt
641  egrep '^[0-9a-zA-Z ]*$' search_list.txt | less
642  egrep '^[0-9a-zA-Z ]*$' search_list.txt | less
643  egrep '^[0-9a-zA-Z ]*$' search_list.txt | wc -l
644  grep '^[-0-9a-zA-Z ]*$' search_list.txt |
645  grep '^[-0-9a-zA-Z ]*$' search_list.txt | egrep '^[0-9a-zA-Z ]*$' search_list.txt
646  grep '^[-0-9a-zA-Z ]*$' search_list.txt | egrep -v '^[0-9a-zA-Z ]*$' | less
647  grep '^[-0-9a-zA-Z ]*$' search_list.txt | grep '^[A-Z]??' | egrep -v '^[0-9a-zA-Z ]*$' | less
648  grep '^[^A-Z]??' search_list.txt | less
649  grep '^[A-Z]??' search_list.txt | less
650  grep -v '^[A-Z][A-Z]*' search_list.txt | less
651  grep -v '^[A-Z]{2}*' search_list.txt | less
652  grep '^[A-Z]{2}*' search_list.txt | less
653  grep '^[A-Z]{2}' search_list.txt | less
654  grep '^[A-Z][A-Z]' search_list.txt | less
655  grep -v '^[A-Z][A-Z]' search_list.txt | less
656  grep -v '^[A-Z][A-Z]' search_list.txt | wc -l
657  grep -v '^[A-Z][A-Z]' search_list.txt > temp
658  mv temp search_list.txt
659  wc -l search_list.txt
660  cp search_list.txt list_bck.txt
661  vi search_list.txt
662  grep '^[A-Za-z][0-9][A-Za-z0-9]*$' search_list.txt
663  grep '^[A-Za-z][0-9][A-Za-z0-9]*$' search_list.txt  | wc -l
```

```
 664  grep -v '^[A-Za-z][0-9][A-Za-z0-9]*$' search_list.txt  > temp
 665  mv temp search_list.txt
 666  vi search_list.txt
 667  grep '^[A-Za-z][0-9][A-Za-z0-9 ]*$' search_list.txt  | wc -l
 668  grep -v '^[A-Za-z][0-9][A-Za-z0-9 ]*$' search_list.txt  > temp
 669  mv temp search_list.txt
 670  cp search_list.txt list_bck.txt
 671  vi search_list.txt
 672  grep '^[A-Za-z0-9][A-Za-z0-9][A-Z]*[A-Z0-9a-z]*$' search_list.txt  | less
 673  grep '^[A-Za-z0-9][A-Za-z0-9][A-Z]{1,}[A-Z0-9a-z]*$' search_list.txt  | less
 674  grep '^[A-Za-z0-9][A-Za-z0-9][A-Z]{1,100}[A-Z0-9a-z]*$' search_list.txt  | less
 675  grep '^[A-Za-z0-9][A-Za-z0-9]*[A-Z]{1,}[A-Z0-9a-z]*$' search_list.txt  | less
 676  grep '^[A-Za-z0-9][a-z0-9]*[A-Z]{1,}[A-Z0-9a-z]*$' search_list.txt  | less
 677  grep '^[A-Za-z0-9][a-z0-9]*[A-Z]+[A-Z0-9a-z]*$' search_list.txt  | less
 678  grep '^[A-Za-z0-9][a-z0-9]*[A-Z]+[A-Z0-9a-z]*$' search_list.txt  | less
 679  grep '^[A-Za-z0-9]*[A-Z]+[A-Z0-9a-z]*$' search_list.txt  | less
 680  grep '^[A-Za-z0-9]*[A-Z]+$' search_list.txt  | less
 681  grep '[A-Z]+' search_list.txt  | less
 682  grep '[A-Z]' search_list.txt  | less
 683  grep '^[A-Za-z0-9][a-z0-9 ]*[A-Z]+[A-Z0-9a-z ]*$' search_list.txt  | less
 684  grep '^[A-Za-z0-9][a-z0-9 ]*[A-Z]?[A-Z0-9a-z ]*$' search_list.txt  | less
 685  ld
```

```
egrep -v "^[0-9]* years" lydia_new.txt  > lydia_without_years.txt
egrep -v "^[0-9].[0-9]* years" lydia_without_years.txt > lydia_tmp.txt
egrep -v "^[0-9].[0-9]* Years" lydia_tmp.txt > lydia_tmp1.txt
egrep -v "^[0-9].[0-9]* Year" lydia_tmp1.txt > lydia_tmp2.txt
egrep -v "^[0-9].[0-9]* year" lydia_tmp2.txt > lydia_tmp3.txt
```

Wikipedia suggestions

National air Duct Cleaners Association

Some Pointers

http://stackoverflow.com/questions/2153371/unix-comm-utility-allows-for-case-insensitivity-in-bsd-but-not-linux-via-i-fl
http://www.unixguide.net/unix/sedoneliner.shtml
http://snowball.tartarus.org/download.php

Stemming

1. Stemmed Lydia names only. The stemmed file contains words in the format <stemmed-word> -> <Original-word>. Get the stemmed words for comparison.
./stemwords -i lydia-unstemmed.txt -o lydia-stemmed.txt -p
cut -f2 -d"-" lydia-stemmed.txt > lydia-stemmed-compare.txt

2. Stemmed Freebase names only. Output file is of same format described above. Get the stemmed words for comparison.
./stemwords -i freebase_unstemmed.txt -o freebase_stemmed.txt -p
cut -f2 -d"-" freebase_stemmed.txt > freebase_stemmed_compare.txt

3. Compare them as usual.

comm -12 lydia-stemmed-compare.txt freebase_stemmed_compare.txt > Common_stemmed.txt
comm -23 lydia-stemmed-compare.txt freebase_stemmed_compare.txt > Lydia_only_stemmed.txt
comm -13 lydia-stemmed-compare.txt freebase_stemmed_compare.txt > Freebase_only_stemmed.txt

Regex:

1. sed -n '/\t[A-Za-z]*.*,.*[A-Z][A-Z]\t/p' < lydia-only-records.txt > temp

2. Eliminate lead and trailing space - sed 's/^[\t]*//;s/[\t]*$//' temp_base > final

3. sed -n '/Technology$/p' < lydia_only.txt >> temp

4.

Cleaning up

1. After translation -> transliteration -> duplicate and white line elimination - few thousands.
2. Check up http://www.outpost9.com/files/WordLists.html - dint use.
3. Using the word list from http://wordlist.sourceforge.net/
4. Fixed the categories and getting the new list from lydia.
5. Got a list of words with variants like sundays brittanys. (Faulty variances)
6. Got a list of abbreviations. Planning to make abbreviations out of lydia list. Create a hash map with abbreviations as key and expansions as values.
7. We are not going to suggest abbreviations as wikipedia, don't we ? (445 abbr)

8. Check whether the words eliminated from Lydia are worth eliminating.
9. With the list of abbreviations I eliminated entries like FRI, THURS.
10. I collected Canadian sports team as well.
11. Remove the common "The" words. Script needed. DONE - 2671 words matched.

Gnuplot

```
gnuplot> set xlabel "Frequency"
gnuplot> set ylabel "Percentile Rank"
gnuplot> set autoscale
gnuplot> plot "./Lyd-Rank.txt" using 1:2 title 'lydia' with linespoints
gnuplot> set logscale x
gnuplot> plot "./Lyd-Rank.txt" using 1:2 title 'lydia' with linespoints
gnuplot> plot "./Lyd-Rank.txt" using 1:2 title 'lydia-only' with linespoints, \
> "./Comm-Rank.txt" using 1:2 title 'Common-Entities' with linespoints
```