Assignment #3
Name Discrimination

given by Zornitsa Kozareva
Information Sciences Institute/University of Southern California
Spring 2012

Date Assignment Given:	March 8, 2012
Date Assignment Due:	March 29, 2012

Note: This assignment requires a lot of programming and some knowledge of Java, start early!

*Send your technical questions to the TA Dirk Hovy by e-mail dirkh@isi.edu (you can cc me if you want)

Finding information about people, organizations and locations in the World-Wide-Web is one of the most common activities of Internet users. Because names are highly ambiguous often the returned results are a mixture of Web pages about different people/locations/organizations that share the same name. The Name Discrimination task concerns the automated identification of ambiguous names and the grouping together of the returned Web pages into clusters such that each cluster corresponds to the same name.

Task Definition

The input is a set of web pages returned by a web search engine when a given person name was issued as a query. For the current final project, we will use only the first one hundred documents retrieved by the search engine.

The output of your system must be a clustering of web pages, such that each cluster corresponds to only one individual.

The gold standard (truth) clustering of the Web pages will be provided from the very beginning. Your goal is to find the best solution exploring, comparing and contrasting:

- different types of context representation and information such as unigrams, bigrams, skip-grams, co-occurrence, named entities among others
- tf*idf weighting
- various clustering methods (k-means, bottom-up, top-down)
- Latent Dirichlet Allocation

You must turn in to kozareva@isi.edu and dirkh@isi.edu

- your source code
- a comparative study on at least two different feature representations and two different clustering algorithms
- a detailed description explaining which settings worked the best and why
- an error analysis on the wrongly clustered Web pages
- a suggestion on how to improve the task, or a suggestion on how you would define the task if you were the first person to come up with it

Data Description

You are provided with two folders that have the following structure

```
webps
      \-- web_pages       //raw web pages downloaded for each name
      \-- truth_files     //human clustering of the documents for each name

scorer_1.1
                //documentation, source and jar files of the evaluation package
```

In **\--web_pages**, you fill find for each name up to 100 web pages, which were returned from Yahoo! when the person name was issued as a Web query. The pages contain the original formatting (html, xml), which must be cleaned prior to the clustering processing.

In **\--truth_files**, you fill find the Gold Standard (i.e. the true clustering) for each person name. The Gold Standard files are named "person_name.clust.xml".
Each file contains a root element "<clustering>" followed by one "entity" element for each entity. The entity element has an identifier attribute ("id") with an integer value. Nested in the "entity" element there are "doc" elements (pages that refer to this particular entity), each of which has a "rank" attribute that corresponds to the ranking information provided in the xml file described above. Note that a document might have been clustered in more than one entity. This is the case when multiple person names referring to different entities appear in a single document. Also, note that a person name may have a namesake that is not a person (for instance an organization or a location). In those cases the non-person entity will have its own cluster. Finally, when the annotator could not cluster a page it was included under a "discarded" element. The reasons for this might be the non-occurrence of the person name in the page (probably because Yahoo index had outdated information when the corpus was built) or simply that the human annotator could not decide whether to cluster that page. Discarded pages are not taken into account for the evaluation.

Here is an example of what the gold standard files looks like:

```
<clustering>
    <entity id="0">
        <doc rank="0"/>
        <doc rank="5"/>
    </entity>
    <entity id="1">
        <doc rank="1"/>
        <doc rank="3"/>
        <doc rank="5"/>
        <doc rank="10"/>
    </entity>
    ...
    <discarded>
        <doc rank="8"/>
        <doc rank="9"/>
    </discarded>
 </clustering>
```

Note that empty lines are permitted. Space and tab do not have special meaning in the file.

Some files that appear in the list of downloaded documents might contain no text, probably because it was not possible to download them. Those pages where not clustered by the human annotators and should not appear in the "clust.xml" files.

Scoring Package:
The scoring folder includes the jar file with the source code and a basic documentation.
Any suggestions, improvements or new measures are very welcomed (write to Javier Artiles javart@gmail.com who is the organizer of the Web People Search Task for 2007, 2009 and 2010).

This program scores the performance of one or more systems according to several optional evaluation measures.

```
USAGE:
 java SystemScorer [keysDir] [systemsDir] [outputDir] [MEASURES] [BASELINES] [OPTIONS]

 [keysDir]          Directory containing all the gold standard for the clustering problems.
                    Files must be well formed XML, follow the WePS 2007 clustering format
                    and filenames end in 'clust.xml'.

[systemsDir]        Directory containing all the systems solutions to evaluate using the
                    following structure

                  systemsDir/TEAM_A/problem1.clust.xml
                  systemsDir/TEAM_A/problem2.clust.xml
                  systemsDir/TEAM_A/...
                  systemsDir/TEAM_B/problem1.clust.xml
                  systemsDir/TEAM_B/problem2.clust.xml
                  systemsDir/TEAM_B/...
                  systemsDir/...

[outputDir]          Directory where all the results will be written

MEASURES:
-ALLMEASURES     Evaluates all the available measures
-P         Purity
-IP        Inverse purity
-FMeasure_0.5_P-IP             F-measure for Purity and Inverse Purity (alpha=0.5)
-BER      BCubed Recall (extended for multiclass problems)
-BEP      BCubed Precision (extended for multiclass problems)
-FMeasure_0.5_BER-BEP      F-measure for BCubed Precision and Recall (alpha=0.5)
-PR       Pairs measure using Rand Statistic
-PJ       Pairs measure using Jaccard Coefficient
-PF       Pairs measure using Folkes and Mallows

BASELINES:
-AllInOne
-OneInOne
```

-Combined

OPTIONS:

-overwrite   overwrites previous evaluation files (.eval) if necessary.
-average   prints the averaged scores for all the teams

EXAMPLE (using the official annotation set as key and also as a team, evaluating baseline answers):

```
$ /usr/lib/jvm/java-6-sun-1.6.0.03/bin/java -cp distributions/1.1/wepsEvaluation.jar
        es.nlp.uned.weps.evaluation.SystemScorer weps07test/truth_official/
        weps07test/test_system/  tmp -ALLMEASURES -AllInOne -OneInOne -Combined -
        average
```

```
WePS 2007 Evaluation Package (http://nlp.uned.es/weps)

Key clustering files path:    weps07test/truth_official
Answer clustering files path: weps07test/test_system
Output evaluation files path: tmp
Measures:                  [P, IP, FMeasure_0.5_P-IP, BEP, BER, FMeasure_0.5_BEP-BER,
        PM, PJ, PR, ]
Baselines:                 [COMBINED_BASELINE, ONE_IN_ONE_BASELINE,
        ALL_IN_ONE_BASELINE]
Overwrite:                 false

Evaluating clustering answers (team truth_official) from weps07test/test_system/truth_official
Saving team evaluation to: tmp/truth_official.eval


Evaluating clustering answers (baseline COMBINED_BASELINE)
Saving team evaluation to: tmp/COMBINED_BASELINE.eval

Evaluating clustering answers (baseline ONE_IN_ONE_BASELINE)
Saving team evaluation to: tmp/ONE_IN_ONE_BASELINE.eval

Evaluating clustering answers (baseline ALL_IN_ONE_BASELINE)
Saving team evaluation to: tmp/ALL_IN_ONE_BASELINE.eval

topic      BEP   BER   FMeasure_0.5_BEP-BER   FMeasure_0.5_P-IP   IP   P   PJ   PM   PR
truth_official              1,0   1,0   1,0   1,0   1,0   1,0   1,0   1,0   1,0
ONE_IN_ONE_BASELINE    1,0   0,43   0,57   0,61   0,47   1,0   0,0   1,0   0,83
COMBINED_BASELINE     0,17   0,99   0,24   0,78   1,0   0,64   0,17   0,34   0,17
ALL_IN_ONE_BASELINE   0,18   0,98   0,25   0,4    1,0   0,29   0,17   0,34   0,17
```

System Output

You are expected to provide an output clustering for each person name

```
Abby_Watkins
Cathie_Ely
Dan_Rhone
Jane_Hunter
Michael_Howard
Thomas_Baker
Tim_Whisler
```

The format of the output should be the same as the Gold Standard format described above. The data output for each person name set should be created in one separate file. The file name should be the person name (blanks replaced by "_") with the ".clust.xml" extension.

Some Useful Materials

Our lecture on Named Entity Discrimination from class
http://www.isi.edu/natural-language/teaching/cs544/
Name Entity Discrimination, Clustering (slides)

Ted Pedersen: Language Independent Methods of Clustering Similar Contexts (tutorial)
http://www.fask.uni-mainz.de/lk/videoarchive/

SenseClusters Toolkit by Ted Pedersen
http://www.d.umn.edu/~tpederse/senseclusters.html

Use the clustering available in Weka

Use LDA from Mallet: http://mallet.cs.umass.edu/api/cc/mallet/topics/LDA.html
A paper on how to use LDA for name discrimination
http://www.isi.edu/~kozareva/papers/kozareva_emnlp11_ned.pdf

WebPS challenges: http://nlp.uned.es/weps/weps-1/weps-1-task-guidelines