From the guys who brought you Dijkstra: The Man the Myth the Legend, Noam Chomsky: 10 Reasons He's Better Than You, and Context Free Grammars and You comes a major group presentation

Badass Films presents "Fuzzy Clustering" Harrison McKenzie Chapter Nathan Giusti Evan Hecht Aleksandr Movsesyan Jude Law Kevin Bacon Morgan Freeman Michael Caine Music By Danny Elfman Directed By Tim Burton Catering Provided by Cal Poly Campus Dinning and the San Luis Obispo Humane Society

December 7th 2009 at a classroom near you

http://wiki.csc.calpoly.edu/csc560/wiki/Bi-Queryous
Who the hell are you and why are you on my lawn?!

- Wikipedia - It's Awesome
- Internal links
- Searching, this thing is huge!

- The conference (INEX)
- What is INEX?
- Lousy Australians
Meet the team!

- Nathan "the Gun" Giusti
- Harrison "I have 3 last names" McKenzie Chapter
- Alecks "the great white Russian" Movsesyan
- Evan "are you Bi-Queryous too" Hecht

Together over 10 computer science internships and knowledge of just about every computer thingy known to man. Together, we are all: Bi-Queryous
The Problem

- The INEX Wikipedia corpus is a large collection of Wikipedia pages encoded and structured using XML.
- Querying information about these XML documents can take a long time, due to the size of the corpus.
- Identifying communities of pages which have low or zero possible relevance to a query type could shrink the search space required to find a query's results.
Problem Statement

- For a set of documents $D$, find a set of communities $C$ for which all members of $C$ are highly related. The relation which groups each $C$ should be some query-able component.
The Data

The data comes in 6 zip files and subdivided into four main categories. First:

- The first zip file contains Tags and Trees
  - The tags file has lines of tag frequencies
    <document ID> <tree ID 1>:<frequency> ... <tag ID n>: <frequency>
  - The tree file has lines of the tree structure
    <document ID> <treeID> <treeID> <length if the String> <depth first traversal>
The Data (cont)

Second
- The file containing links.
- This file is a list of pages and the pages to which the current pages links to.
- The file is in the format:
  <document ID> <link ID 1> ... <link ID n>
Third

- The files containing entity information
  - The files are encoded in the following format
    `<document ID> <feature ID 1>::<frequency> ... <feature ID n>::<frequency>`
  - This zip file also contains files that are as follows
    - `entity.tag.freq.id` - entity tags sparsely encoded
    - `entity.tag.stats` - CSV of stats about tag features
    - `entity.text.freq.id` - text inside entity tags sparsely encoded
    - `entity.text.stats` - CSV of stats about text features
Fourth

- The files here are the BOW and index file
  - File BOW
  - File Index
Fuzzy K-means

A fuzzy version of K-means, \( u_k(x) \) represents x's degree of belonging to cluster k.

- \( \Sigma u_k(x) = 1 \)
- \( \text{center}_k = \text{avg}( u_k(x) \times x ) \)
- \( u_k(x) \propto \text{dist}(\text{center}_k, x) \)

Algorithm:
- Choose num. clusters, random \( u_k(x) \) values
- Until convergence:
  - Compute new centroids
    - Compute new \( u_k(x) \)
Fuzzy Maximum Likelihood Estimation

Iteratively estimate the most likely non-exclusive cluster memberships.

Algorithm:
- Choose starting memberships, calculate centroids.
- Until convergence:
  - Find probability functions for cluster membership based on centroids and current hidden values
  - Find memberships which maximize the probability functions

Deals well with varying cluster size and density. Very sensitive to starting parameters.
Mixing K-means and MLE

Algorithm:
- Choose starting clusters
- Use K-means to find clusters
- Use MLE, starting with the K-means clusters
- Measure cluster 'performance'
- Increase number of clusters until optimal performance

Cluster Performance Measures:
- Separation between clusters
- Minimal cluster volume
- Maximal data point concentration around cluster center