Assignment 4

• Indexing
• You can do it in groups of 1, 2 or 3
• Try to use hot topic queries (e.g. news)
Search Engine Scraping


- http://www.google.com/search?q=sara+javanmardi
Using API

- **Yahoo! API**
  Your key: jkAJFxDV34EdbrBxx.jyu8RER9RYT3AWfkAu6FozAoZ3LhdxMfmiEs1LK6Ef2ALkaabGNzeT

- **Bing API**
  Your key: FBD074880BDA4F7CB3C24DD576E654F0B7CA7EFB

- Google doesn’t provide any API
- **Google ajax search**
Developer Registration

Fields marked with an asterisk * are required.
There are errors with your submission. Please check fields highlighted in red.

*Yahoo ID: javanmard

*Authentication method: Click here for more information
  - Generic, No user authentication required
    - This appid will allow you to make calls to our non-authenticated web services
  - Browser Based Authentication
    - Use this option for browser applications

*Developer/Company Name: University of California, Irvine
  For example: ‘Joe/Jane Developer’ or ‘BigCo Inc.’

*Product name: IR Winter 2010
  For example: ‘My Yahoo! Enabled Web App’

Web Application URL: 
  For example: ‘http://myapp.com/welcome.html’

*Contact email: sjavanma@uci.edu
  For example: ‘developer@domain.com’

Phone number: 
  For example: ‘123-456-7890’

*Description of application:
  (250 characters or less):
  A small university project
Create a new AppID

To create a new AppID, enter your information below and click Agree to accept the API terms of use.

**Required information**

- **Application name**: [Input field] (60 character limit)
- **Description**: A small university project (500 character limit)
- **Company name**: University of California, Irvine (100 character limit)
- **Country/region**: USA, CA (100 character limit)
- **Email address**: sjavarna@uci.edu (e.g., webmaster@example.com)

We will use this address to notify you of issues that affect API usage.

- [ ] I also want to receive promotional offers from Bing at this email address.

**Optional information**

- **Website**: [Input field] (80 character limit)

**API terms of use**

**BING WEB SERVICE API TERMS OF USE**
Last Updated: June 2009

THANK YOU FOR CHOOSING MICROSOFT!

1. What does this Contract cover? This is a contract between you and Microsoft Corporation ("Microsoft"). Some of the Microsoft Bing Web Service API (the "API"). The APIs are intended to deliver relevant results (collectively, "Bing services") for rendering within a customer-facing or end-user-facing website ("Website") or application. Through reflect the content type, format, and detail desired. The Bing service(s) may enable access to multiple content type results, the Bing services now enable Microsoft search ads ("Microsoft Ads"). You may sign up by clicking here applicable to Microsoft Ads to have Microsoft Ads delivered with the algorithmic Bing results that are provided by the Bing services, and any related application id ("AppID(s)").

Please note that we do not provide warranties for the services. The contract also limits our liability. These terms, available" basis, in accordance with §13, and you acknowledge that licencing with regard to delivery of Bing results and subject further to your acknowledgement that we will not be liable to you or to any third party if we fail to ac

- **Service Level Goals**
  - **Uptime**: 99.5%
  - **Query Response Time**: 1.5 seconds

- [ ] Click here if you agree to the terms and conditions detailed above.
Using APIs

• Each group needs to have its key
• Send request politely otherwise you will be blocked.
• You can ignore non-html search results if any
Posting List

To create the posting lists, you have 3 options

1) term : docID

2) term : (docID:termFrequency)

3) term : (docID, position of the term in the document)

Note:
Use option 2 or 3 because in the next assignment we need to have *tf.idf*
Inverted Index

• Each index term is associated with an *inverted list*
  – Contains lists of documents, or lists of word occurrences in documents, and other information
  – Each entry is called a *posting*
  – The part of the posting that refers to a specific document or location is called a *pointer*
  – Each document in the collection is given a unique number
  – Lists are usually *document-ordered* (sorted by document number)
Example “Collection”

$S_1$ Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

$S_2$ Fishkeepers often use the term tropical fish to refer only those requiring fresh water, with saltwater tropical fish referred to as marine fish.

$S_3$ Tropical fish are popular aquarium fish, due to their often bright coloration.

$S_4$ In freshwater fish, this coloration typically derives from iridescence, while salt water fish are generally pigmented.

Four sentences from the Wikipedia entry for *tropical fish*
Simple Inverted
Index

and 1
aquarium 3
are 3
around 1
as 2
both 1
bright 3
coloration 3
derives 4
due 3
environments 1
fish 1
fishkeepers 2
found 1
fresh 2
freshwater 1
from 4
generally 4
in 1
include 1
including 1
iridescence 4
marine 2
often 2
only 2
pigmented 4
popular 3
refer 2
referred 2
requiring 2
salt 1
saltwater 2
species 1
term 2
the 1
their 3
this 4
those 2
to 2
use 2
tropical 1
typically 4
water 1
while 4
with 2
world 1
### Inverted Index with counts

- supports better ranking algorithms

<table>
<thead>
<tr>
<th>Term</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>aquarium</td>
<td>3:1</td>
</tr>
<tr>
<td>are</td>
<td>3:1</td>
</tr>
<tr>
<td>around</td>
<td>1:1</td>
</tr>
<tr>
<td>as</td>
<td>2:1</td>
</tr>
<tr>
<td>both</td>
<td>1:1</td>
</tr>
<tr>
<td>bright</td>
<td>3:1</td>
</tr>
<tr>
<td>coloration</td>
<td>3:1</td>
</tr>
<tr>
<td>derives</td>
<td>4:1</td>
</tr>
<tr>
<td>due</td>
<td>3:1</td>
</tr>
<tr>
<td>environments</td>
<td>1:1</td>
</tr>
<tr>
<td>fish</td>
<td>1:2</td>
</tr>
<tr>
<td>fishkeepers</td>
<td>2:1</td>
</tr>
<tr>
<td>found</td>
<td>1:1</td>
</tr>
<tr>
<td>fresh</td>
<td>2:1</td>
</tr>
<tr>
<td>freshwater</td>
<td>1:1</td>
</tr>
<tr>
<td>from</td>
<td>4:1</td>
</tr>
<tr>
<td>generally</td>
<td>4:1</td>
</tr>
<tr>
<td>in</td>
<td>1:1</td>
</tr>
<tr>
<td>include</td>
<td>1:1</td>
</tr>
<tr>
<td>including</td>
<td>1:1</td>
</tr>
<tr>
<td>iridescence</td>
<td>4:1</td>
</tr>
<tr>
<td>marine</td>
<td>2:1</td>
</tr>
<tr>
<td>often</td>
<td>2:1</td>
</tr>
<tr>
<td>only</td>
<td>2:1</td>
</tr>
<tr>
<td>pigmented</td>
<td>4:1</td>
</tr>
<tr>
<td>popular</td>
<td>3:1</td>
</tr>
<tr>
<td>refer</td>
<td>2:1</td>
</tr>
<tr>
<td>referred</td>
<td>2:1</td>
</tr>
<tr>
<td>requiring</td>
<td>2:1</td>
</tr>
<tr>
<td>salt</td>
<td>1:1</td>
</tr>
<tr>
<td>saltwater</td>
<td>2:1</td>
</tr>
<tr>
<td>species</td>
<td>1:1</td>
</tr>
<tr>
<td>term</td>
<td>2:1</td>
</tr>
<tr>
<td>the</td>
<td>1:1</td>
</tr>
<tr>
<td>their</td>
<td>3:1</td>
</tr>
<tr>
<td>this</td>
<td>4:1</td>
</tr>
<tr>
<td>those</td>
<td>2:1</td>
</tr>
<tr>
<td>to</td>
<td>2:2</td>
</tr>
<tr>
<td>tropical</td>
<td>1:2</td>
</tr>
<tr>
<td>typically</td>
<td>4:1</td>
</tr>
<tr>
<td>use</td>
<td>2:1</td>
</tr>
<tr>
<td>water</td>
<td>1:1</td>
</tr>
<tr>
<td>while</td>
<td>4:1</td>
</tr>
<tr>
<td>with</td>
<td>2:1</td>
</tr>
<tr>
<td>world</td>
<td>1:1</td>
</tr>
</tbody>
</table>
Inverted Index
with positions

• supports proximity matches
Proximity Matches

• Matching phrases or words within a window
  – e.g., "tropical fish", or “find tropical within 5 words of fish”

• Word positions in inverted lists make these types of query features efficient
  – e.g.,

<table>
<thead>
<tr>
<th>tropical</th>
<th>1,1</th>
<th>1,7</th>
<th>2,6</th>
<th>2,17</th>
<th>3,1</th>
</tr>
</thead>
<tbody>
<tr>
<td>fish</td>
<td>1,2</td>
<td>1,4</td>
<td>2,7</td>
<td>2,18</td>
<td>2,23</td>
</tr>
</tbody>
</table>
Fields and Extents

• Document structure is useful in search
  – *field* restrictions
    • e.g., date, from:, etc.
  – some fields more important
    • e.g., title

• Options:
  – separate inverted lists for each field type
  – add information about fields to postings
  – use *extent lists*
Extent Lists

• An *extent* is a contiguous region of a document
  – represent extents using word positions
  – inverted list records all extents for a given field type
  – e.g.,

<table>
<thead>
<tr>
<th>fish</th>
<th>title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,2</td>
<td>1:(1,3)</td>
</tr>
<tr>
<td>1,4</td>
<td>2:(1,5)</td>
</tr>
<tr>
<td>2,7</td>
<td>2,18</td>
</tr>
</tbody>
</table>

extent list