Schedule

- Last thursday – Project overview and some ideas
- Today – More detail on projects
  - Size, write-up, and grading criteria
  - More project ideas
  - Walkthrough of how to use some of the tools we are making available
- Next week: Lectures on vector space IR model
Recap on the project

- An opportunity to do a reasonable sized project on a topic of interest related to the course.
- 40% of grade
- Work in teams of two
  - Fill in the form to give us an update on groupings
  - If you don’t yet have a partner:
    - Stay after class to look for a partner if you don’t have one
    - Try asking on news:su.class.cs276a
    - TA can help you find a partner
Timeline

- Oct 3, Oct 10: discuss project ideas and resources in class
- Oct 17 (one week from today!): Hand in project abstract (1 page, on paper, in class, to TA, or in box by Gates 419)
- We will email feedback by Oct 24
- Nov 12: Submit summary of progress, and plans for completing the project (2–3 pages)
- Dec 3: Submit project
- Dec 5: Present project (very brief “executive summary”) in class
Project size

- Always difficult to define…
- But a 3 unit course, 2 people, 40% of the grade…
  - … perhaps 60 person hours of work?
- Can alternatively be defined in terms of the size of the project report
  - 8 pages max in style of conference paper
- We can help tune size in our feedback on abstracts, but it’s hard
- Really, things depend more on quality, ideas, crisp results than number of hours spent
Grading dimensions

- Creativity in defining the problem to be investigated
- Quality of the methods used investigating it
- Thoroughness in considering and justifying your design decisions
- Quality of your write-up
  - Reporting methods, results, discussion, etc.
- Quality of evaluation of your system
- Clean, intelligible code
- You will not be penalized if your system performs poorly, providing your initial design decisions weren't obviously unjustifiable, and you have made reasonable attempts to analyze why it failed, and to examine how the system might be improved
Quality

- Try to make it something that shows something interesting, not just an exercise in programming
  - But it can be a small something – be focused and not too ambitious
Project report

- 8 pages in the style of a conference paper
  - You’ll probably want to do it 2-column
- Quality of write-up is very important
  - You can’t expect us to find something clever that you did buried in your code
  - You have to tell us about it and explain why it is clever
- The model is an academic conference paper
  - If you haven’t read any, look at a few for examples of focused exploration of a (frequently quite specific) problem
What should be in the report?

- It’s hard to give a one-size-fits-all plan but …
  - A clear research question or application, and hypotheses about a good approach to it
  - A clear and complete discussion of the theory/algorithms/method used
  - A high-level description of the implementation
  - Testing, and discussion of the results obtained
    - Clear graphs, tables, experimental comparisons
  - A discussion of alternatives and their performance
    - (preferably thoughtful revisions not random changes)
  - Brief but adequate discussion of related work
Rough grading guidelines

- In practice, we need some flexibility to assess quality of projects, but in a typical case, the rough point distribution we intend to use is:
  1. Originality of idea: 10 pts
  2. Experimental methodology: 10 pts
  3. Results, fit and finish: 10 pts
  4. Quality of writeup including adequate scholarly comparison to prior work: 10 pts
- Total: 40 pts
Final Deliverables

- Online submission of the project code
- On paper submission of report
  - Makes it easier for us!
- Online submission of the report
  - So we can put up a website of projects
- Capsule presentation of project in last class
More Project Ideas
Detecting index spamming

- *I.e., this isn’t about the junk you get in your mailbox every day!*

- As will be discussed on Tuesday, most ranking IR systems use frequency of use of words to determine how good a match a document is, and at any rate having lots of terms in an area makes you more likely to have the ones users use.

- There’s a whole industry selling tips and techniques for getting better search engine rankings from manipulating page content.
#3 result on Altavista for “luxury perfume fragrance”
Detecting index spamming

- Clicking anywhere on the page (or on the next several screenfuls) took one to the "real" front page of the site.
- A couple of years ago, there was lots of use of "invisible" text in the background color, but I think there is less of that now, as search engines check for it as a sign of spam.
- Can one use term weighting strategies that make an IR system more resistant to spam?
- Alternatively, can one detect and filter pages attempting index spamming?
  - E.g. a language model run over pages.
  - (From the other direction, are there good ways to hide spam so it can’t be filtered??)
Investigating performance of term weighting functions

- People in IR have explored a range of families of term weighting functions
  - Frequently getting rather more complex than the simple version of tf.idf which we will explain in class

- Investigate some different term weighting functions and how retrieval performance is affected
  - One thing that many methods do badly on is correctly relatively ranking documents of very different lengths – but this is a ubiquitous web problem – so that might be a good focus
E.g. of a “real world” term weighting function

- “Okapi BM25 weights” are one of the best known weighting schemes
  - Robertson et al. TREC-3, TREC-4 reports
  - Discovered mostly through trial and error

$N$ is the number of documents in the collection

$n_t$ is the number of documents containing term $t$

$tf_{t,d}$ is the frequency of term $t$ in document $d$

$w_{t,d}$ is the contribution of term $t$ to the relevance of document $d$

$$w_{t,d} = 0.4 + \frac{0.6 \cdot tf_{t,d}}{tf_{t,d} + 0.5 + 1.5 \frac{\text{length}(d)}{\text{avglen}}} \cdot \frac{\log N + 0.5}{n_t} \cdot \frac{n_t}{\log N + 1}$$
Investigating performance of term weighting functions

- Using HTML structure:
  - HTML pages have a good deal of structure (sometimes) – in terms of elements like titles, headings etc.
  - Can one incorporate HTML parsing and use of such tags to significantly improve term weighting, and hence retrieval performance?
Learning term weighting functions

(This is a more ambitious project than the last one)

- If one has a parametric family of term weighting functions (such as Okapi BM25), and some data set (such as TREC) for which one has queries with answers, then one could view it as an optimization problem to find the function/weights in the family that optimize performance on some score.

- Need to have score function smooth so can optimize rather than it becoming a combinatorial problem.

- TREC query sets are quite small: is overtraining a big problem? Test this on new data.
Alternative displays of results

- The standard for displaying IR results is an ordered list
- But there are some alternatives:
  - www.kartoo.com
  - http://www.sims.berkeley.edu/~hearst/cac-overview.html
- Design and evaluate an alternative user interface to results
  - (Why has the ordered list dominated so?)
Finding the news

- One thing many standard web search engines do poorly is finding current news
- A prominent additional dimension is date
- By and large, people want recent news or at any rate news from a particular time
- Can one combine topic relevance with recency to give good search results on a news topic (e.g., “Bulgarian economy”)?
Finding the news

- One could crawl the top-level pages of news sites, and locally index them
  - Careful not to hit sites too often!
- Or, for a personal assistant system, it might be reasonable to not pre-index but to go off and find information on the basis of requests, based on their category, which would need to be detected, and normally also organizes the news site
Language identification

- People commonly want to see pages in languages they can read
- But sometimes words (esp. names) are the same in different languages
- And knowing the language has other uses:
  - For allowing use of segmentation, stemming, query expansion, ...
- Write a system that determines the language of a web page
Language identification

Notes:
- There may be a character encoding in the head of the document, but you often can’t trust it, or it may not uniquely determine the language.
- Character ngram level or function word based techniques are often effective.
- Pages may have content in multiple languages.
- Google doesn’t do this that well for some languages (see Advanced Search page).
  - I searched for pages containing “WWW” (many do, not really a language hint!) in Indonesian, and here’s what I got...
Okay!

KCM - Kompas Cyber Media 2002
Kurs 1 US$: Rp 9.000 - 9.030 sumber : AYU, Jakarta hujan 23°C-34°C
sumber : BMG. Updated: Senin, 07 Oktober 2002 - 12:56 wib, ...
Description: Edisi online koran dengan oplah terbesar di Indonesia, terbitan Yayasan Gempa Bumi
Category: World > Indonesia > Benta > Koran
www.kompas.com/ - 46k - Cached - Similar pages

1A1: Sudan Page @Sudan.Net
SUDAN PAGE. Home keywords: Sudan Khartoum Sudan Sudan Al-Sudan El-Sudan
Khartoum Sudan Sudan Sudan Sudan Sudan Khartoum Shendi ...
Description: Latest News, Discussion, Country facts, Government Information, Music and more.
Category: Regional > Africa > Sudan > News and Media
www.sudan.net/ - 3k - Cached - Similar pages

Untitled Document
www.asean.or.id/ - 2k - Cached - Similar pages

Welcome To Malaysia Airlines
Adelaide, ...
Description: Includes online schedules and fare information.
Category: Recreation > Travel > Transportation > Air > Airlines > Malaysia
www.malaysiaairlines.com.my/ - 25k - Cached - Similar pages
SivaSakti.com - In-depth Yoga Resource

... : Ready for Acupuncture? http://www.acupuncture.com pertinently asks the question: "What the Heck is Acupuncture?" and answers is even more pertinently. Auch! ...
Description: Offers tantric articles, Ebooks, asanas, yantras and meditation music.
Category: Society > Religion and Spirituality > Tantra
sivasaakti.com/ - 23k - Cached - Similar pages

Please select the country you are in
-Select-. ...
Description: Free bilingual web based e-mail for the Arabic and English speaking Intern
Category: Regional > Middle East > ... > Internet > E-mail Service
www.maktoob.com/ - 6k - Cached - Similar pages

SearchIndonesia.com - Think Fresh, Search Indonesia.

English. Situs Fresh Baru, dua kali sehari! Cari Fresh Indonesia : Lebih
Detail. Situs Top 10 Situs Populer. siBerita siFinasial ...
Description: Searchable directory of Indonesian sites
Category: Regional > Asia > ... > Business and Economy > Internet > Directories
www.searchindonesia.com/ - 41k - Cached - Similar pages

<< Era Aviation, Inc. >>
10/7/02, Origin Origin. ...
Description: Commercial airline with specialized services from air medical services to first-class seeing tours
Category: Regional > North America > ... > Alaska > Transportation > Aviation
www.era-aviation.com/ - 53k - Cached - Similar pages

Astagal.com
Lacak Artikel Astaga. ...
Description: Situs portal berita Indonesia, dengan berbagai fasilitas seperti email gratis, fax, ar
cuaca,...
Category: World > Indonesia > Komputer > Internet > Web
www.astaga.com/ - 40k - Cached - Similar pages
Or something else...

- Again, the ideas we’ve given certainly aren’t meant to be an exhaustive list
- You’re probably better than us at coming up with creative new ideas!
- Anything else related to the topics of the course (broadly construed) is fine, providing it’s of reasonable scope, etc.
Tools & Resources II
For those wanting to have a web interface to their projects, one option is to use CGI.

CGI accounts have been set up for all registered students.

Details at http://www.stanford.edu/leland/cgi/

Note: some amount of additional magic is used to get this to work with Java, so there may be a few kinks.

Might need some PHP or Perl glue.
CGI access: setup

- Substitute your Leland id for `user`

```bash
cd ~user
mkdir cgi-bin
fs setacl cgi-bin user.cgi write
```
CGI access: the script

- In ~user/cgi-bin/, create a script `program.sh`
  - `chmod 700 program.sh`
- The following is also in cs276a/software/cgi-lib/program.sh

```bash
#!/bin/tcsh

echo Content-type: text/plain
echo

source /afs/ir/class/cs276a/software/setpaths.sh

# the following should be one line
setenv LD_LIBRARY_PATH
${LD_LIBRARY_PATH}:/afs/ir/class/cs276a/software/cgi-lib

# run your program, giving the full path to java
/usr/pubsw/bin/java cs276a.examples.JWNLExample $argv
```
CGI access: the script cont.

- Also see
  cs276a/software/cgi-lib/program_wrapper.pl
  for an example of extracting form parameters

- The URL of the script (what should go in the "action" for forms):
  http://cgi-courses.stanford.edu/~user/cgi-bin/program.sh
  or
  http://cgi-courses.stanford.edu/~user/cgi-bin/program_wrapper.pl
CGI access: wrap-up

- Your program can output HTML
- Details are beyond the scope of this session...
  - tutorials on web
  - books available
  - ask on newsgroup for advice
- If neither you or your partner have ever done web scripting before, it might be better to stick to a standard GUI or command line interface.
WebBase: Feeder vs Handler

- Two styles of accessing a stream of pages
  - pull-style iterative interface (Feeder)
    - getNextPage()
  - push-style iterative interface (Handler)
    - processPage() callback is registered with a driver

- Both provide a uniform access method to different sources of Web repositories
WebBase: Feeder vs Handler

- pull-style iterative interface (Feeder)
  - Your program retains thread of control
  - Convenient for client code

- push-style iterative interface (Handler)
  - Driver program retains thread of control
  - Sometimes inconvenient, but if you have multiple indexes or analyses you want to generate simultaneously, this allows you to instantiate multiple handlers to process the same web stream.
Simple demo application

- http://cs276a.stanford.edu/projects/docs/examples/cs276a/examples/titleindexer/

- Demo program that builds an index over the titles from the WebBase Stanford.EDU web crawl using Lucene. The WordNet library is used to provide index expansion. The Google API is used to correct spelling errors in queries. The sources are in software/examples/cs276a/examples/titleindexer/
Index construction: outline

- Create an **IndexWriter**
- For each page
  - Create **Document** for the page
    - must parse HTML if necessary
  - Add it to **IndexWriter**
    - **IndexWriter** will use an **Analyzer** to generate a **TokenStream**, adding **Tokens** to the inverted index.
      - **Tokenizer**: Convert document to **TokenStream**.
      - **TokenFilter**: Filter the **TokenStream** (stem, stopword-removal, ...)
      - The **Analyzer** should be configured with one **Tokenizer** and one or more **TokenFilters**.
Index search: outline

- Create an `IndexSearcher`
- For each query
  - Create a `Query` using a `QueryParser`
    - `QueryParser` will also use an `Analyzer` to tokenize and filter the query (but will also build up a query tree)
  - Use the `IndexSearcher` to retrieve the `Hits` for the `Query`
  - Iterate over the `Hits` and print them out