Along with the explosive growth of online textual information (e.g., web pages, email, news articles, social media, and scientific literature), it is increasingly important to develop tools to help us access, manage, and exploit the huge amount of information. Web search engines, such as Google and Bing, are good examples of such tools, and they are now an essential part of everyone's life. In this course, you will learn the underlying technologies of these and other powerful tools for accessing and mining text information. You will be able to learn the basic principles and algorithms for information retrieval as well as obtain hands-on experience with using existing information retrieval toolkits to set up your own search engines and improving their search accuracy.

Unlike structured data, which is typically managed with a relational database, textual information is unstructured and poses special challenges due to the difficulty in precisely understanding natural language and users' information needs. In this course, we will introduce a variety of techniques for accessing and mining text information. The course emphasizes basic principles and practically useful algorithms. Topics to be covered include, among others, text processing, inverted indexes, retrieval models (e.g., vector space and probabilistic models), IR evaluation, text categorization, text filtering, clustering, topic modeling, retrieval system design and implementation, issues of web search engines, and applications of text retrieval and mining.

This course is designed for graduate students and senior undergraduate students of the School of Information and Computer Science and Engineering. The course is lecture based. Grading is based on three individual assignments, a midterm exam, and a course project.

Learning Objectives:

- Understand how search engines work.
- Understand the limits of existing search technology.
- Learn to appreciate the sheer size of the Web.
- Learn to write code for text indexing and retrieval.
- Learn about the state of the art in IR research.
- Learn to analyze textual and semi-structured data sets.
- Learn to appreciate the diversity of texts on the Web.
- Learn to evaluate information retrieval systems.
- Learn about standardized document collections.
- Learn about text similarity measures.
- Learn about semantic dimensionality reduction.
- Learn about the idiosyncrasies of hyperlinked document collections.
- Learn about web crawling.
- Learn to use existing tools of information retrieval.
- Understand the dynamics of the Web by building appropriate mathematical models.
- Build working systems that assist users in finding useful information on the Web.
Papers from SIGIR, WWW, and other venues to be assigned in class

Sample assignments:

1. Getting familiar with text analysis (problem sets + light programming);
2. Basic retrieval systems (Light C++ coding and analysis);
3. Mini-Trec – build and evaluate a light search engine (semi-automatic data harvest, query selection, evaluation);
4. Text classification (programming/data analysis)
Kaggle-in-class competition will be used in one or more of the assignments.

Final Project:

An open-ended research project in one of three categories:

- Research paper - using the SIGIR format. Students will be in charge of problem formulation, literature survey, hypothesis formulation, experimental design, implementation, and possibly submission to a conference like SIGIR or WWW.
- Software system - develop a working, useful system (including an API). Students will be responsible for identifying a niche problem, implementing it and deploying it, either on the Web or as an open-source downloadable tool.
- Survey paper - identify a topic of research in IR and summarize 15-20 recent papers on it, along with an introduction that compares and contrasts the papers involved.

Grading:

- Class Participation: 5%
- Homework assignments: 40%
- Midterm Exam: 20%
- Course Project: 35%
  - Proposal: 5% (due at the 6-th week)
  - Progress report: 5% (due at the 10-th week)
  - Final presentation: 5%
  - Final report: 20%

Tentative Schedule:

Week 1: Introduction to information retrieval; Mathematical basics
- Vector spaces and similarity;
- Probabilities and Statistics;

Reading: Vannevar Bush "As We May Think" Bush 45.

Week 2: Text Analysis and Preprocessing.
- Document processing, stemming;
- String matching;
- Basic NLP tasks – POS tagging; shallow parsing.

Reading: Thorsten Brant, "Natural Language Processing in Information Retrieval"

Week 3: Overview of text retrieval systems.
- System architecture;
- Boolean models;
- Inverted Indexes;
- Document ranking;
- IR Evaluation;
Reading: [Manning] Chapter 1 "Boolean retrieval"; Chapter 8 "Evaluation in information retrieval"

Week 4: Retrieval models and implementation: Vector Space Models
- Vector space models;
- TF-IDF weighting;
- Retrieval axioms;
- Implementation issues;

Reading:
* [Manning] Chapter 6 "Scoring, term weighting & the vector space model"

Week 5: Probabilistic models; statistical language models
- Okapi/BM25;
- Language models;
- KL-divergence;
- Smoothing;

Reading: [Manning] Chapter 11, Chapter 12; [Zhai] Chapter 3

Week 6: Query expansion and feedback
- Query reformulation;
- Relevance feedback;
- Pseudo-relevance feedback;
- Language model based feedback;

Reading: [Manning] Chapter 9, Chapter 12; [Zhai] Chapter 5

Week 7: Web Search Engines;
- Models of the Web;
- Web crawling;
- Static ranking: PageRank; HITS;
- Query log analysis;
- Adversarial IR;

Reading: [Manning] Chapter 19-21; MapReduce for building inverted index (optional, read Ch2-4)

Week 8: Information filtering. User Interfaces.
- Adaptive filtering;
- Collaborative filtering;
- User Interfaces.

Week 9: Text classification;
- Naïve Bayes; k-nearest neighbors;
- Feature selection;
- Semi-supervised Learning;

Reading: [Manning] Chapter 13,14, and 15 up to SVM.

Week 10: Text clustering;
- Vector-space clustering;
- K-means; EM algorithm;
- Text shingling;

Reading: [Manning] Chapter 16, 17

Week 11: Graph-based methods;
- WordNet; Document and word graphs;
- Network analysis;
- Random Walks;
- Harmonic Functions;

Reading: [Manning] Chapter 21; [Newman]
Week 12: Latent Semantic Analysis; Topic Modeling;
- Latent semantic indexing;
- Singular value decomposition;
- Probabilistic Latent Semantic Indexing;
- Latent Dirichlet Allocation;

Reading: [Manning] Chapter 18; [Zhai] Chapter 7.1; PLSA note

Week 13: IR applications;
- Information extraction
- Question answering
- Opinion summarization
- Computational advertising

Week 14: Project Presentations.

Midterm will be take-home, at around Week 10 or Week 11.