EPL660: Information Retrieval and Web Search Engines
FINAL PROJECT

Written research projects must be done individually or in groups of no more than two. Implementation projects can be done in groups of up to 3.

Each group or individual must submit your preference projects proposal (e.g. W2, I2, W4) to be approved (email to instructors: P. Antoniou and cc-ed G. Pallis) no later than March 1.

Written Projects
Written projects involve doing an in-depth study, survey, or evaluation of one or more topics related to information retrieval and Web search engines. The project can take a form of a research paper examining the use of a specific technique or model in various IR systems, or it can be a detailed case study involving two or more existing IR systems. In either case, the paper should contain a summary and a technical evaluation of the state-of-the-art related to the particular topic studied. If the paper involves a case study, then a thorough comparative evaluation with other similar systems must be provided. A research paper should present a new idea or provide a detailed survey of methods to solve a specific IR-related problem. The approach presented should be, at least in part, a novel and original contribution, and should ideally be evaluated experimentally. A research paper could be good start for a Masters or Ph.D. research project. The maximum length for the written projects is 20 single-spaced pages (12 point font), including figures and references. The evaluation of the papers will be based on clarity, thoroughness, and soundness of ideas and concepts presented, as well as the overall organization of the paper.

Note: Written project should not simply be a summary of some of the material covered directly in the lectures, but rather should go beyond this material in one or more specific areas related to that material.

W1. A comparative study of implementation techniques for scalable information retrieval on large-scale search engines or Web-based information systems (such as Google, Facebook, etc.). This study must include an analysis of challenges in managing and leveraging large data repositories and various proposed and implemented solutions (such as Big Table, Map Reduce, and other approaches based on "cloud computing"). The study can also focus on implementation platforms that enable scalable retrieval (e.g., Hadoop).

W2. Study of the use of social network analysis (SNA) and its use in information retrieval. This study should include a detailed summary of various techniques from SNA and their use in providing relevant information to users in online social network and/or traditional search engines. This project should also include related works in social network aware search, and the related subtasks of information retrieval, aggregation, and ranking in a social context.

W3. A comparative study of Information Retrieval approaches in Decentralized Social Networks. There is a transformational change in Online Social Service provision, pushing the state-of-the-art from centralized services towards totally decentralized systems that will pervade our environment and seamlessly integrate with future Internet and media services. OSN decentralization can address privacy considerations and improve service scalability, performance and fault-tolerance in the presence of an expanding base of users and applications.

W4. Exploring various techniques for Web IR based on hyperlink analysis and mining. The study should include examination of techniques based on linkage as a measure of authority of the information source (e.g., HITS or Pagerank algorithms), as well as other techniques to use ratings or popularity as measures of quality or authority**.**

Implementation Projects

Implementation projects involve the development and evaluation of an original application using information retrieval techniques. The application must be tested and evaluated using appropriate test data sets. The application must also involve the use of one or more of the modeling techniques relevant to the course topics. Your application may also include a significant extension of an existing applications or techniques discussed in class materials or other sources (in this case, the application must be extended to include additional or more sophisticated types of features). The deliverable for the project must include the fully documented code, distribution files, including any third party sources, installation/deployment documents (including examples, screen shots of test runs, etc.), data used for the evaluation of the application, and a detailed project report providing a description of the components of the application and the results of evaluation.

I1 Build your own search/retrieval system:

- Should include implementations for the basic components including separate crawler, indexer, and query processing components (including a reasonable query interface)
- Should work on a local document corpus in a directory structure or as a Web search engine (applied to a limited set of Web sites or for a specific domain)
- The indexing component should parse and index documents using inverted file format with relevant term frequency information
- Should make use of stemming and stop lists (you can use existing tools for this part).
- The system should use TF-IDF weights (and possibly additional weighting schemes) for index terms
- The base implementation should use the vector-space model with Cosine similarity to be used for the matching queries and indexed documents. Optionally, you can implement other retrieval models such as probabilistic models or models based on link analysis.
- It should be possible to save the index to an offline storage and reload it for subsequent retrieval sessions (during a retrieval session, the search engine should run in the background as a server process and handle incoming queries).
- Optional components or functionality can be added depending the desired features or complexity of the project, including: additional weighting schemes, part-of-speech tagging, phrase indexing, n-gram indexing, proximity operators, personalized search, and relevance feedback.

I2 Implement a personalized information filtering system:

- Your system should provide the capability for selective dissemination of information based on a user profile.
- The system should obtain and subsequently update a user's profile represented as a set of topics (e.g., using a vector-space representation).
- Based on the user profile, the system (in the background) should search for items of interest to the user. Depending on the type of target domain, these items could be interesting Web pages, news stories, blog posts, tweets or posts on other social networking sites, or even objects of interest (movies, books, consumer items, etc.). The applications can be a general information filtering agent, or an agent designed to work in a specific target domain (e.g., a personalized shopping agent, a news filtering agent, etc.).
- The user's profile should be updated when the user provides feedback on one or more of recommended items (e.g., using relevance feedback).
- The system should minimally include components to create and maintain an

index of items/documents selected, a component to maintain and update a user profile, and a component to search selected Web sites in the background for items of interest.
- Optionally, the system can include additional features such as clustering and categorization of items selected for the user; the ability to update search for items similar to a recommended item selected by the user (e.g., "more like this" capability), etc.

I3 Design an enhanced user interface for a retrieval system:

- Your interface should help guide the user in formulating a query. You can explore options such as the use of a classification hierarchy (such as Yahoo's category labels), providing the capability for natural language queries (possibly through the use of WordNet and basic natural language processing tools such as part-of-speech tagging), adding context-awareness by maintaining a user profile (based on past searches or other types of preference elicitation) in order to reduce ambiguity in queries, etc.
- Your system should also provide and enhanced interface for the user to browse the retrieved documents and provide mechanisms such as relevance feedback and query by example.
- Finally, the system should have the ability to cluster the retrieved documents (preferably using hierarchical clustering) and present the clusters to users for easier navigation and browsing.
- For this project you don't have to implement your own indexing and matching algorithms, however, you may need to modify an existing system (with source code) to incorporate the additional capabilities. You may also need to do post processing of documents retrieved as results of a search.

I4. Propose a project:

- Email a paragraph describing your proposed project to the instructors of the course. Include as much detail as possible. The instructors will reply with any concerns about the content or scope of the project. If you are proposing a project with a partner, one partner should email the description and the other partner should email a confirmation of his/her involvement.

Guidelines

Written Projects

Written projects will be evaluated based on thoroughness, soundness, clarity and organization. The overall structure of the paper is up to you, but you must have the following sections in addition to the main body of the paper:

Abstract: This is a short synopsis of the main points of the paper. This should be 200-300 words, and should appear along with the title and your name, ID number, and email, on the first page. The rest of the paper should start on page 2.

Conclusion: Summarize your conclusions and findings. Keep this to 300-400 words.

References: This is a list of references that you have used in doing your research and throughout your paper. The references should be numbered and the number for the reference should appear in the appropriate places in the text of the paper where the reference was used (it is not enough to list a bibliography at the end of the paper without actually using any references within the body of the paper). You can look at any of the papers included in the optional reading for acceptable uses of references. URL references should only be used for referring to specific system Web sites and not as a way to reference published work.

Your final paper should be sent by email either PDF or Postscript formats. Submissions in other formats will not be accepted.

Project presentation (15 min per team + 5 min Q/A)

Implementation Projects

You will need to electronically submit a compressed file by email containing your project distribution files and documentation. Your project documentation should contain the following components:

- A detailed description of your system (including specific techniques and algorithms you used), and the interaction between the components (make references to code segments, modules, methods, functions, etc. as necessary). If you used any outside sources in your implementation, please clearly indicate which sources, and how and where they were used.
- A complete sample run of your program with description, illustrating how your system works, along with any intermediate input or output used for the sample run.

Your project distribution files should contain the following:

- Complete source code (be sure that your source code is fully documented and easy to read).
- Binary files (e.g., executables, DLLs, Class files) or other components necessary to run your program.
- Readme file containing instructions on how to compile, install, and/or run your program.
- If your application is CGI-based or otherwise has a server component, please provide a URL for a demo version of your system.

Project presentation (15 min per team + 5 min Q/A)