Course Project

Introduction

The purpose of the course project is to provide students an opportunity to study and work on a topic in-depth so as to obtain experience with either (1) practical experience in implementing advanced IR algorithms in a real system, or (2) application of advanced IR technologies to develop an interesting novel application system. Group projects with a team of up to **TWO** students are allowed and encouraged. Also, **COMP 5970** students can **only** partner with another **COMP 5970** student, while **COMP 6970/6976** students can only partner with **COMP 6970/6976** student.

Grading Criteria

Your project will be graded primarily based on the following weighting scheme:

1. Project Proposal Presentation (10%)
2. Project Proposal Report (20%)
3. Project Final Presentation (30%)
4. Project Final Report (40%)

Topics

There are three distinct project topic tracks. They differ mainly in the proposed end goal of the project.

1. **Software Track (For COMP 5970 only).** This track aims to contribute a major piece of open-source software to the community. This may be in the form of a brand-new software package, or as an extension of an existing toolkit or library.

   Some topics of interest are given below:

   - Core AI (e.g. deep learning for IR, embeddings, intelligent personal assistants and agents).
   - Question Answering (e.g., factoid and non-factoid question answering, interactive question answering, community-based question answering, question answering systems).
   - Conversational Systems (e.g., conversational search interaction, dialog systems, spoken language interfaces, intelligent chat systems).
   - Explicit Semantics (e.g. semantic search, named-entities, relation and event extraction).
   - Knowledge Representation and Reasoning (e.g., link prediction, knowledge graph completion, query understanding, knowledge-guided query and document representation, ontology modeling).
   - Filtering and Recommending (e.g., content-based filtering, collaborative filtering, recommender systems, recommendation algorithms, zero-query and implicit search, personalized recommendation).
   - Document Representation and Content Analysis (e.g., summarization, text representation, linguistic analysis, readability, NLP for search applications, cross- and multi-lingual

search, information extraction, opinion mining and sentiment analysis, clustering, classification, topic models).
- Knowledge Acquisition (e.g. information extraction, relation extraction, event extraction, query understanding, human-in-the-loop knowledge acquisition).

2. **Startup (Business) Track (For both COMP 5970 and COMP 6970/6976).** This track is application-oriented and should result in a demo system that clearly demonstrates novel useful functions from a user's perspective. Think of this as a "minimum viable product" for a startup idea.

   Some topics of interest are given below:

   - Local and Mobile Search (e.g., location-based search, mobile usage understanding, mobile result presentation, audio and touch interfaces, geographic search, location context in search).
   - Social Search (e.g., social networks in search, social media in search, blog and microblog search, forum search).
   - Education (e.g,. search for educational support, peer matching, info seeking in online courses/MOOCs).
   - Legal (e.g., e-discovery, patents, other applications in law).
   - Health (e.g., medical, genomics, bioinformatics, other applications in health).
   - Knowledge Graph Applications (e.g. conversational search, semantic search, entity search, KB question answering, knowledge-guided NLP, search and recommendation).
   - Other Applications and Domains (e.g., digital libraries, enterprise, expert search, news search, app search, archival search, new retrieval problems including applications of search technology for social good).

3. **Research Track (For COMP 6970/6976 only).** This track should culminate in a research paper that could be suitable for submission to a journal or conference in information retrieval or related areas (like SIGIR, KDD, ACL, WSDM, etc.)

When picking a topic in this track, try to ask yourself the following questions:

- What is the main research question? A research contribution is generally in the form of providing an answer to a new question that has never been addressed by any published work or a better answer to an old question than the previous answers. A research question can be of many different forms. In engineering, a common form of question is: how do we build X? Another common form is: how can we build a better technology to do X than the current technologies available? Sometimes, however, a question may be a "why" question such as "why is algorithm A better than algorithm B under condition C?"
- What is the expected novelty of your work? Is your research question a new question or one that has been studied by others? If it has been studied before, why should we expect your new solution to be better than the previous ones? What's your new idea?
- How important is your question? Why do we have to address the question? What if we don't address it? Who will benefit from studying the question?

A good topic is one that is novel, that you like to work on, that people care about, and that you have some idea about how to carry out the proposed work.

Project Proposal Report

You must write a two-page proposal before you actually begin your project in-depth. These will be submitted via Canvas. In the proposal, you should (1) indicate the track of your project (i.e., software track, or startup track), (2) address the following questions, (3) include the names and email addresses of all the team members, and (4) identify the coordinator of the project in case the team has more than one member, who would take the primary responsibility of coordinating the work of all team members; the coordinator is also our primary contact for providing feedback about the project.

As long as these questions are addressed, the proposal does not have to be very long. A couple of sentences for each question would be sufficient.

If your project is aiming at the **software track**, you should focus on the following in your proposal:

- Define clearly what you’re planning on implementing. (What is its function? What will it do?)
- Is this going to be a contribution to an existing toolkit or library (strongly encouraged), or is this a standalone tool? If it’s a contribution to an existing toolkit or library, outline exactly why you think your contribution will be desirable for the maintainers of that software. If it’s a standalone tool, justify this decision. Why does it make sense for this to not be contributed as a module or extension of an existing piece of software?
- Briefly identify who the users of your software will be, and why it is important to write your software.
- Do we already know how to implement your software? That is, is there prior art for implementing the specific algorithm(s) you plan on using in your implementation? What is the challenge?
- How do you plan to build it? If this is a contribution to an existing piece of software, you should demonstrate that you know the procedure for contributing to that software. Briefly describe that procedure and identify key contributors that may likely review your contribution. If this is a standalone tool, identify what technologies you plan on leveraging to implement your software. This may be programming languages, supporting libraries, etc.
- How will you demonstrate the usefulness of your tool?
- Provide a rough timeline to show when you expect to finish what. List a couple of milestones if possible (they can be tentative).

If your project is aiming at the **startup track**, you should focus on the following in your proposal:

- Define clearly the societal need your system addresses. Why is it important to address this need?
- Who would be your customers if this were to become a real startup company? Discuss how you might decide to monetize your system.
- Who are your competitors? Are there existing systems that address this same need, or similar needs? How is yours going to be different from or better than these existing systems? Do we already know how to build such a system? What is the challenge?
- How do you plan on building your system? Include things like your planned technology stack (programming languages, supporting libraries) and deployment strategies (MS Azure, AWS, DigitalOcean, etc.).
- How will you demonstrate the usefulness of your tool?
- Provide a rough timeline to show when you expect to finish what. List a couple of milestones if possible (they can be tentative).

As long as these questions are addressed, the proposal does not have to be very long. A couple of sentences for each question would be sufficient.

If your project is aiming at the **research track**, you should focus on the following in your proposal:

- What is your research question? Clearly define the research problem/question.
- Why is this an interesting question to ask and why would we care about the answer to this question or a solution to the problem?
- Has any existing research work tried to answer the same or a similar question, and if so, what is still unknown? (In other words, what is the novelty of your research question?) Provide a brief list of related work.
- How do you plan to work out the answer to the question? (At the proposal stage, you are only expected to have a sketch of your methods.)
- How would you evaluate your solution? That is, how do you plan to demonstrate that your solution/answer is good or is reasonable?
- A rough timeline to show when you expect to finish what. List a couple of milestones if possible (they can be tentative).

Project Proposal Presentation

This will be a 10-minute presentation based on your project proposal report.

Project Final Report

At the end of the semester during the final exam week, every project team must submit a project report. The form of your final report will differ depending on which project track you choose.

- **Software track:** The main artifact to deliver here is the software itself, which will make up the majority of the grade for your report.

  You, however, must also submit a written report. This report should serve two purposes: (1) it should provide a description of the algorithm(s) implemented to solve the task in sufficient mathematical detail to understand the implementation challenge, and (2) it should provide long-form documentation in the form of a tutorial that describes how to use your software.

  Your software description should be approximately 5-6 pages and should describe implementation challenges and provide pseudocode algorithms for the main contributions of your software. If possible, please also provide benchmarks of your software against other methods (either other implementations of the same method, or implementations of a similar method or baseline).

  In your documentation, you should provide enough details that a new user could utilize your software. There is no page limit here, but you should attempt to be as detailed as is necessary, but not overly verbose. Your documentation should be practical. (No, your code is not documentation.) We should be able to follow your tutorial and actually be able to use your software.

  Because of the nature of the final deliverable for such a project, a strong emphasis will be placed on your actual software itself. You will need to provide a link to your project, which should be made available as open-source software under an appropriate license of your choosing (e.g., GPL, Apache, MIT, etc.).

  Grading. We will focus on three main areas: (1) **[20%]** Software description (clarity, completeness); (2) **[30%]** Software documentation (clarity, completeness); (3) **[50%]** Software's importance, quality, and impact (**2-3% bonus points** if your patch has been accepted or is *actively* in-review during project grading!).

- **Startup track:** The main artifact to deliver here is a real, live demo system that represents your minimum-viable-product and could provide value to real users.

  You will also submit a brief (5–6 page for COMP 5970, 9-10 pages for COMP 6970/6976) "vision" report outlining the real societal need your startup could address, who your competitors are, how your system is substantially different than all of your competitors, and how you plan to monetize and grow. You should include a link to where your live system resides and some instructions on how we can try it out. Describe briefly the architecture of your system.

  Grading. We will focus on: (1) **[20%]** Your Startup Motivation (clarity, presentation quality); (2) **[30%]** Your written "vision" report (clarity, completeness); (3) **[50%]** Your demo system itself (importance of the problem being addressed, quality of implementation as assessed by the live demo, and potential impact of the company).

- **Research track:** Your final report should be in the form of a research paper that could be suitable for publication in a major conference or journal in information retrieval or related areas. These should be two-column and use the ACL Latex Template. Your paper should be **at least** 9 pages using this template, and **no more** than 10 pages.

  You can think of it as an expanded written description of your proposal with more substantial research results and details. Thus, the same general guideline for your project proposal applies, i.e., you should (1) explain your problem clearly; (2) provide sufficient motivation for your work and explain how your work is connected with the existing/previous work; (3) explain your methods with sufficient details; (4) discuss the research results; (5) summarize your work, draw conclusions if possible, and discuss how you think the work can be further improved/extended.

  A good report is not just a straightforward description of what you did; it should also include a discussion of the challenges that you had to solve and how you solved them. It is especially important to clearly state the novelty of your work (e.g., a novel research question that wasn't previously answered, or a new method that hasn't been tried for an existing problem, or a novel system function that has never been supported by any existing system), and explain why we should care about this novelty. In general, it is important that you think very clearly about what are the major points you want to make and include *arguments* and empirical evidence that support your points. For example, whenever you make a choice of some method or design an algorithm in a particular way, always justify it, at least, try to discuss what are the possible choices and explain why you have made this choice. Also, you may want to summarize or plot your experiment results in a particular way to best illustrate your findings. Always keep in mind what exactly you expect your readers to learn from your report, including both positive and negative findings. As a research report, it is especially important to clarify what **new** knowledge you have discovered through working on the project. In other words, focus on describing what your readers can **learn** from reading your report that they would not be able to learn from any other published literature.

  Grading. We will focus on three factors: (1) **[20%]** clarity and completeness of the report itself (i.e., whether you have clearly described what you have done and addressed all the questions that you are supposed to address); (2) **[30%]** amount of work that you have done; and (3) **[50%]** the quality of your project as reflected in the importance of problem being addressed, the quality of solution, and the impact of your project.

Each project group needs to submit a single report. **Please make sure to include, for each member of your team, a couple of sentences to describe what he/she did exactly for the project.**

Project Final Presentation

This will be a 20-minute presentation based on your project final report.