# Project Proposal

**What are the names and NetIDs of all your team members? Who is the captain?**

Sanjib Ghosh: [sanjibg2@illinois.edu](mailto:sanjibg2@illinois.edu) (Team Captain)

Bo-Ryehn Chung: [brchung2@illinois.edu](mailto:brchung2@illinois.edu)

Matt DiNauta: [dinauta2@illinois.edu](mailto:dinauta2@illinois.edu)

Pawel Zuradzki: [pzurad@illinois.edu](mailto:pzurad@illinois.edu)

**What topic have you chosen? Why is it a problem? How does it relate to the theme and to the class?**

Our project falls under the “Intelligent Learning Platform” theme. Our idea is to develop a method that links Coursera course content (via lecture video transcriptions) to academic journal articles. This will enrich the Coursera course content by allowing students to see how topics being taught have appeared in scientific literature, hence more ‘intelligent learning.’ To accomplish this, we will mine for topics in a dataset of academic papers and a dataset of lecture video transcriptions, applying NLP techniques either directly taught in CS410 or closely related.

**How will you demonstrate that your approach will work as expected? Which programming language do you plan to use?**

We’ll use the Python programming language.

There are a couple approaches we will explore to demonstrate our project (we will choose one of the following):

1. We plan to create a simple command line application as a proof-of-concept. A user can enter a search query and be returned a list of topics as results. The user then selects a topic. Upon selecting, the user will be presented with related course transcripts and papers.
2. Create an index of topics, with references to the lecture segments related to that topic, and references to papers related to that topic.

**Please justify that the workload of your topic is at least 20\*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.**

A list of the main tasks as we’ve currently conceiving of the project:

1. Scrape or download Coursera lecture transcripts. (5 hours)
2. Preprocess the text data: tokenize/stem, BoW with unigrams/ngrams (2 hours)
3. Build a topic model for the Coursera lecture transcript data. We plan to explore different levels of granularity, e.g. associate the entire video with a topic, a paragraph, a sentence, an n-gram. (30 hours)
4. Build a topic model for the academic paper data. We will mine the topics from paper abstracts, using [this](https://www.kaggle.com/Cornell-University/arxiv) dataset available on Kaggle.com. (10 hours)
5. Link the results of 2 and 3 together, implement basic search functionality, and build the command line application. If time permits, we may also add functionality that returns to the user “related topics” and link to the video clips relevant to the topic. (20 hours).
6. Alternative approach for matching topics of lectures to papers: score on abstract-to-document text similarity instead of topic matches. Provide evaluation (confusion matrix, F1, etc). (20 hrs)
7. Scoring Evaluation (15 hours):
   1. need relevant/not relevant labels and topic labels for some test queries. Could label with pseudo feedback, or explicit relevance feedback (manually label the topic and relevant/not relevant)
   2. avg precision for specific queries
   3. MAP/gMAP across multiple queries
   4. nDCG for multi-level relevance
   5. research scoring metrics for topics
8. DevOps, Deployment (5 hours)
   1. Github repository: <https://github.com/sanjibg01/CS410-project-team-trailblazers.git>
   2. Command line interface (ex: argparse package)
   3. Publishing as a package for distribution and reproducibility
      1. Ex: `$ pip install git+<https://github.com/user/our_project.git>`
   4. Public data store for datasets (Google Drive, GitHub[?], S3).