

CS 410 Tech Review

Angus Jyu

University of Illinois at Urbana-Champaign

Review of Team Trailblazers' Proposal

Intro:

The proposal I am reviewing is from the team that has named themselves the “Team Trailblazers”. The team has four members, Sanjib Ghosh: sanjibg2@illinois.edu (who is the Team Captain), Bo-Ryehn Chung: brchung2@illinois.edu, Matt DiNauta: dinauta2@illinois.edu, and Pawel Zuradzki: pzurad@illinois.edu. Their project falls under the “Intelligent Learning Platform” theme. Their 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, they 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.

Body:

First, we have the proposed approaches. The first approach was that they planned 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. I think this is a solid approach, and it's similar to one of the approaches that my team project will be attempting.

The second approach that they planned was 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. This is also a similar approach to one that my own team will be embarking on, and I think it's a solid approach.

Overall, I think both approaches are solid. If there was one thing I might nitpick, it's that maybe the two approaches are too similar and thus not giving enough room for flexibility.

Next, we can move on to the next section of the proposal, and in this case, the final one: the workload length. The project has 7 tasks: One, scrape or download Coursera lecture transcripts. Two, preprocess the text data: tokenize/stem, BoW with unigrams/ngrams. Three, build a topic model for the Coursera lecture transcript data. Four, build a topic model for the academic paper data. Five, link the results of 2 and 3 together, implement basic search functionality, and build the command line application. If time permits, they will also add functionality that returns to the user "related topics" and link to the video clips relevant to the topic. Six, create an 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). Seven, do a scoring evaluation. Overall, I think the outline of this process is very solid. The steps connect/build upon each other well.

In terms of time allocations regarding the steps of the process, I think it may take more time for Steps 1 and 2, and less time for Step 3 (though I mostly mean redistributing the time allocations, I don't think the project as a whole takes less/more time than is projected). I think Step 1 might take ~8 hours, Step 2 might take ~5 hours, Step 3 might take ~25 hours. Those are just my personal estimations, though.

Conclusion:

Overall, I think the proposal outlined is solid as a whole and doesn't have much to critique. I think both approaches are valid (as they are very similar to the approaches my own team is taking), and the process makes sense and is structurally easy to follow. While I did include some nitpicks for "criticisms", they are very minor and aren't necessarily inherent problems.

References:

sanjibg01. (n.d.). SANJIBG01/CS410-project-team-trailblazers. GitHub. Retrieved November 7, 2021, from <https://github.com/sanjibg01/CS410-project-team-trailblazers>.