1. Team

The names and net IDs of the team members are the following:

- Manu Vinod Shesha (manuv3) (Captain)
- Angus Jyu (angusfj2)
- Sofia Godovykh (sofiaag5)

2. Topic

The topic is “**Intelligent Topic Modeling and Index building of Course on EducationalWeb**”. This loosely aligns with Theme 2: Intelligent Learning Platform, with the sub-area of “ConceptView”.

2a. Problem

Courses like “CS-410 Text Mining and Text Analysis” are presented on MOOCs (like Coursera/EducationalWeb) as a series of lectures, each lecture roughly discussing one main idea and some related concepts. However, the title of a lecture video may reveal underneath topics/concepts, to a small degree. As a user, it may be difficult to find out where was a topic or concept discussed in all of course lectures. The other challenge is that many times individual lectures contain more than one topic, and all the topics may not be apparent by just looking at the lecture title.

2b. Proposal

We believe Topic Modeling may be one effective way to solve this challenge. Intuitively, a list of topics extracted from a particular lecture transcript, should provide a snapshot of what the lecture covers. At the same time, such a model, once built, for a Course, can act as building block for building other solutions which can ease user’s learning experience. As a possible application of such a model, we want to demonstrate a topic “index” for the course.

So, at high level, we want to:

Stage 1: Discover primary topic(s) as well as secondary topics in a given lecture. Here, we want to do comparative analysis of algorithms: PLSA and LDA, to measure the relative effectiveness of these algorithms in a paradigm of scientific/technical corpus (like course transcript). This will also give us opportunity to compare effectiveness and ease of use of tools like Lemur, Gensim etc.

Stage 2: Explore ways to do Automated Labeling, to create phrases (like 2-gram phrases), which define the “concepts” in more meaningful ways, than just “bag of word” representation of topics in traditional sense of Topic Modeling. For example: individual topics like “Dirichlet” and “distribution” can be more meaningful for a user if we can do a bit of semantic analysis and discover labels like “Dirichlet

distribution”. The idea is to use techniques which can improve parameters: **Topic Coverage** (which means that discovered topics fully represent the document) and **Topic Differentiation** (which means that the topics are specific enough to differentiate between two documents). Here we want to explore the techniques like using NLP “chunkers”, and “Context Model”. The inspiration is the research paper by Professor C Zhai: **Automatic Labeling of Multinomial Topic Models.**
(Please note that this is an analysis/discovery goal for this project and may not necessarily apply to our final solution. At minimum, we will discuss what techniques were applied and how it impacted the topic models)

Stage 3: Extend EducationalWeb UI to provide the index-like representation of the topic models, which can provide a mapping of topics to relevant lectures. This will greatly enhance the learning experience as it will be useful for the learner to quickly navigate to specific segments of lecture based on "topic". This idea goes beyond the basic search provided by EducationalWeb, where a user can search for a topic and this search task is treated as a simple "bag of word" search to give a whole list of results which may and may not be relevant for the user. Also, a traditional search on such learning platforms are not comparable to full-fledged browser, and here, the assumption is that the user has some idea on what he is searching for. Instead, we want to provide a push-kind of model, where we generate an index of relevant topics covered in a course (spread across a series of lectures).
(As an option, we will try to index specific video segments, based on topics. However, this is an optional goal for this project, depending on current capability of EducationalWeb)

3. Datasets, Algorithms and Tools

3a. Datasets

- CS410 course transcipts available on EducationWeb/Coursera as our Collection Corpus.
- Human generated topics (by TAs and former students) for the lectures here, to evaluate our models.
- Built-in Background language models, in tools like Gensim, Lemur, etc.

3b. Algorithms

We plan to use **PLSA** (Probabilistic Latent Semantic Analysis) or **LDA** (Latent Dirichlet Allocation) to generate the topic models.

For automated labeling, we would use the techniques like Chunking, Ngram testing, and Semantic Relevance scoring, discussed in the paper, referenced in section 2b.

3c. Tools

- Gensim
- EducationalWeb

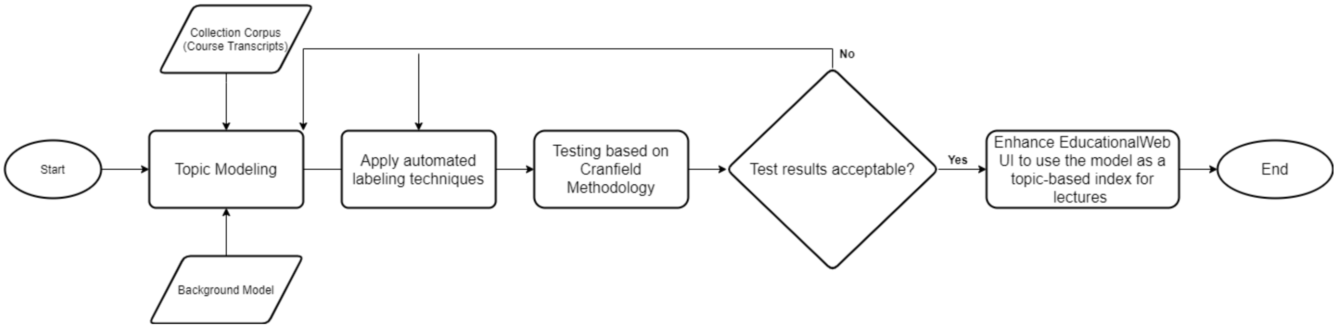
3d. Language

- Python
- C++

4. Approach Testing

4a. Procedure

Below flow chart highlights the process:


Collection Corpus (Course Transcripts)
No
Start
Topic Modeling
Apply automated labeling techniques
Testing based on Cranfield Methodology
Test results acceptable?
Yes
Enhance EducationalWeb UI to use the model as a topic-based index for lectures
End
Background Model

4b. Testing

We will demonstrate whether the approach works as expected or not through testing and comparisons. We will draw up a model of what the ideal index output should be by manually going through the videos and parsing out the key topics that ideally would be indexed, and then compare this ideal model to the produced model to see how close to the ideal the model gets. We intend to use the Python/C++ programming language for this testing.

5. Workload Time Estimations

Our time estimations regarding the workload will be separated into three main tasks.

- Task 1 [Discover primary topic(s) as well as secondary topics in a given lecture]: **15 hrs**
  For this task, we will do some comparative analysis of PLSA and LDA, as well as different tools like Jensim, Lemur etc. to get best Topic models.
- Task 2 [Automated Labeling techniques]: **20 hrs**
  For this task, we will need to try different heuristics suggested in the paper as well as additional techniques like text scrapping from slides. like we estimate the time cost to be around 20 hours.
- Task 3 [Cranfield Test Suite]: **15hrs**
  For this task, we would build a small script, to provide a reliable, repeatable evaluation of our generated models.
- Task 4 [Extend EducationalWeb UI] : **20 hrs**
  Analyze the current UI design and add a section with index-like representation of the topic models, which can provide a mapping of topics to relevant lectures.

In total, we expect the project to have a time cost of ~**70 hours**, which surpasses the 20*3 = 60-hours requirement.

6. References

[1] http://timan102.cs.illinois.edu/explanation/

[2] https://docs.google.com/spreadsheets/d/1V_PvgwqUifq6QlmXl-0EDQutaCVErYfSDl09HJwE6W0

[3] https://radimrehurek.com/gensim/

[4] http://www-personal.umich.edu/~qmei/pub/kdd07-label.pdf