Progress Report

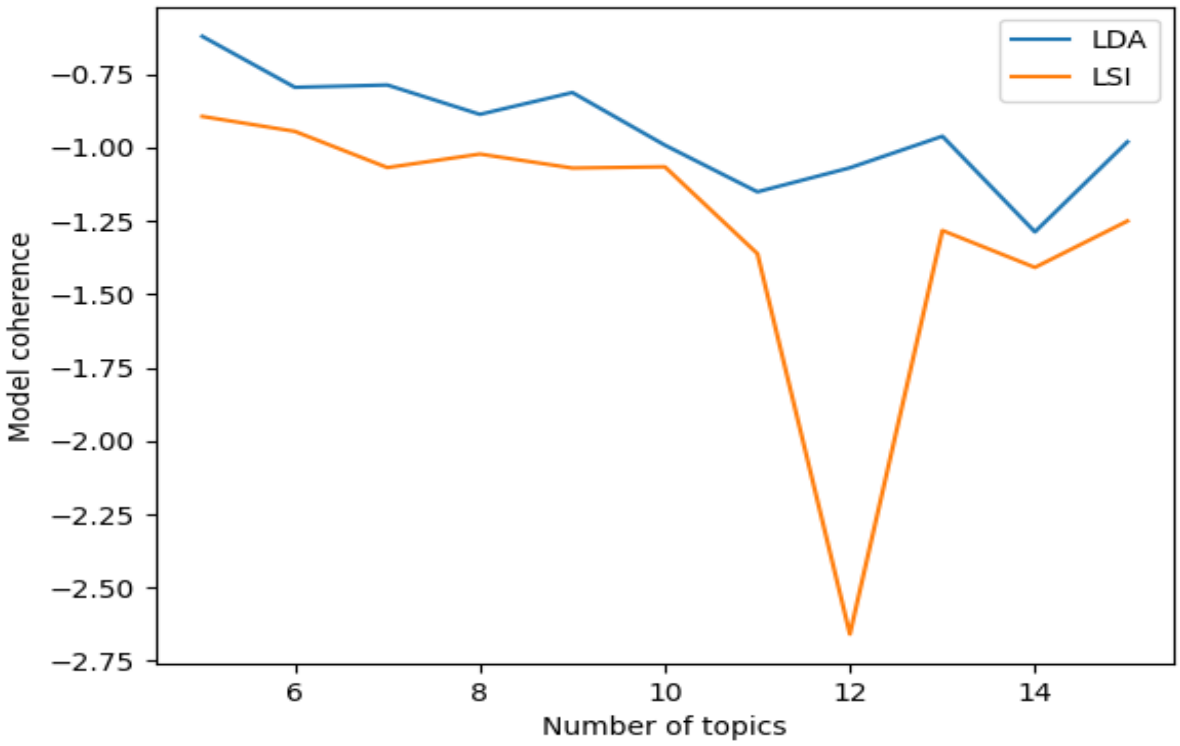
What we have achieved so far:

- ★ Our project requires CS410 transcripts and slides as input corpus for extracting models and for building indexes. So we wrote a crawler (using Selenium with Python) for Coursera, which crawled successfully through the course and downloaded all transcripts and slides. This would have been very time consuming manually. We also believe that this is a reusable component and should work for other courses on Coursera.
    - The downloaded files are here: https://github.com/manuv3/cs410-project/tree/main/data

- ★ We were able to get to speed on using Gensim and added scripts to build corpus out of raw text. This included steps:
    a. Tokenization using NLTK RegexpTokenizer
    b. Removing stop words provided both in NLTK and Gensim
    c. Lemmatization using NLTK WordNetLemmatizer
    d. Generating and persisting VSM (Vector Space) model as dictionary
    e. Generating and persisting MM (Market Metrics) format based Corpus.

  This code was witten making use of Gensims streaming enabled APIs to achieve "constant memory" processing, which would be very useful with large corpus.

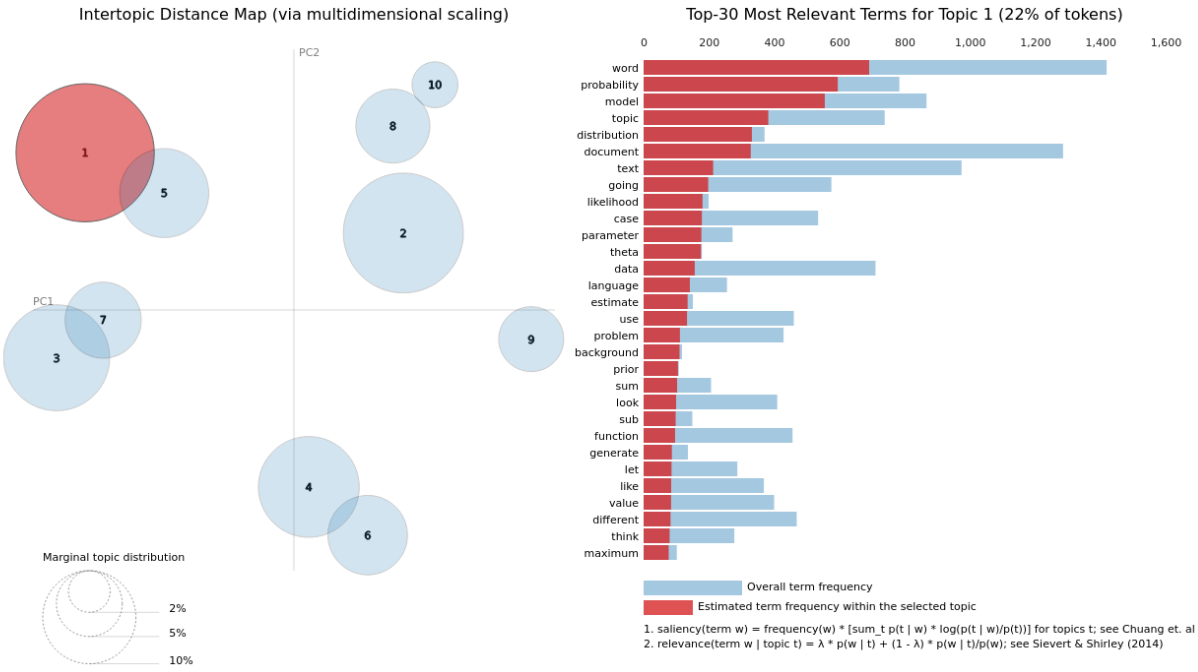
- ★ We built two Topic models based on LDA (Latent Dirichlet Allocation) and LSI (Latent Semantic Analysis), to compare the relative performance. The performance was computed based on "Topic Coherence", using "u_mass" score per topic.
    - We found that LDA was indeed providing more coherent topics. As a result, we plan to use it for our project.



★ Currently, we have managed to create “somewhat” coherent topics as shown below:

topic_1	document query word vector model term function space like retrieval look count weight score idf actually matching problem probability frequency
topic_2	category categorization document network example different case result measure application method text evaluation general human important decision feature right way
topic_3	user information search document engine relevant filtering item based feedback example utility retrieval query threshold data text need talked problem
topic_4	word probability model topic distribution document text going likelihood case parameter theta data language estimate use problem background prior sum
topic_5	rating aspect weight model hotel reviewer different overall result review value going word case use average mean like set course
topic_6	document page relevant user query case search precision think look recall going result different mean measure actually right link retrieval
topic_7	category feature value training data rating classifier problem function going parameter categorization beta use object approach learning different similar regression
topic_8	text data topic mining example opinion context analysis word technique time knowledge different general lot language natural course sentence use
topic_9	document term word value index function count topic use example general id inverted going like code data key algorithm file
topic_10	word context similarity relation cluster entropy clustering probability information going case object eats general group conditional look different similar occur

★ The above shown topics seem to have decent intertopic distance:



All the code to our project can be found here:
https://github.com/manuv3/cs410-project/tree/main/code

Challenges and next steps

- ★ Some of the important technical terms have very low probability across the topics, most probably due to low TF. We want to apply some heuristics to boost their probability. Maybe we will leverage additional documents created by parsing the slides as well as the title of the lectures. This means we want to leverage context in topic modeling process.

- ★ We want to explore methods to generate meaningful phrases (bigram) out of the topics or documents.
    - We want to implement techniques like using NLP “chunkers”, and “Context Model” discussed in the research paper by Professor C Zhai: Automatic Labeling of Multinomial Topic Models.
    - Professor Zhai has also suggested the use of Word2vec algorithm (based on skip gram modelling)

- ★ We still need to analyze the UI design of EducationalWeb, to design our UI component (which is basically an index of topics/concepts).