Team Members

NAME	USER ID	ROLE
BRIAN REINBOLD	brianjr3	Captain

MOOC platforms like Coursera have already innovated on the traditional classroom, making it easier for students to learn at their own pace. However, these platforms currently lack "intelligence." Now with emerging technologies in text mining, natural language processing, and machine learning; there is an opportunity to create new tools and functionality to enhance the learning experience. This project relates to the smartmoocs platform in the intelligent learning platforms theme. Specifically, it will "explore better ways to segment lectures based on topic transition."

Currently, the lectures are segmented into uniform, 1-minute length segments. Ideally, these segments should be determined based on topic change. Having well-defined segments will then make it easier to determine sub-topics within a lecture, which can make it easier for users to navigate. This project will require data mining transcripts and determining topic change within text documents, which are very relevant to this class.

A major part of this project will be to download all the lecture transcripts and then process them to generate a dataset for the analysis. For the analysis, I will first divide a lecture transcript into *T* documents where $T = \frac{Total\ Lecture\ Time}{Time\ Interval}$. In other words, the lecture transcript will be transformed into a sequence of documents across time at, for example, ten second intervals. I will then adapt several techniques found in Wang and Goutte 2017 to detect topic change by evaluating the similarity between two consecutive documents. If document D_{t-1} and D_t are similar, then it is unlikely a topic change has occurred, but if they are not similar, then it is more likely that a topic change has occurred. Wang and Goutte 2017 used cosine similarity:

$$D_{\cos}(t) = 1 - \cos(P_W(t), P_W(t-1)),$$
$$\cos(P_W(t), P_W(t-1)) = \frac{\langle P_W(t), P_W(t-1)\rangle}{\|P_W(t)\|\|P_W(t-1)\|}$$

and they used Jensen-Shannon divergence:

$$D_{\mathrm{JSD}}(t) = \mathrm{JSD}(\overline{P}_W(t), \overline{P}_W(t-1)),$$
$$\mathrm{JSD}(P,Q) = \frac{1}{2}\sum_i p_i \log \frac{2p_i}{p_i+q_i} + \frac{1}{2}\sum_i q_i \log \frac{2q_i}{p_i+q_i}.$$

$D_{cos}(t)$ and $D_{JSD}(t)$ form a time series, and several statistical techniques can be used to detect a change point, which is a location where the underlying stochastic process changes. If a change point is found, then a topic change has likely occurred. There are several change point detection algorithms like Bayesian change point detection of Barry and Hartigan (1993), and the nonparametric, hierarchical divisive algorithm of James and Matteson (2015).

Finally, I will run my algorithm on all lecture transcripts and output times at which topic changes occurred per lecture. Now evaluating whether my technique is better is more challenging since this is an unsupervised learning technique, i.e. I do not already know the ideal topic change times.

One possibility to objectively measure the quality of the topic changes is to compare the inter- and intra-cluster distances between my proposed time segments and the baselines. Currently, the baseline topic segments are one-minute intervals. Another useful baseline is to assume that there are no topic changes within a lecture. This assumption is reasonable since not every lecture will have subtopics, especially in short lectures. The words within each time segment form clusters, and if these segments are well defined, each cluster should correspond to a topic. The words within a cluster should be more related than the words outside the cluster. This implies that well defined time segments should form clusters that minimize their intra-cluster distance and maximize their inter-cluster distance.

Another approach is I can annotate several lectures with my own judgments of topic change. Obviously, I do not have time to annotate all videos, but being able to compare the baselines with my proposed algorithm's suggested segments will help qualitatively determine if there is any improvement. At the least, it may help determine if this approach is a step in the right direction.

I will primarily use the python programming language. The references for the statistical techniques to detect change points in a time series cited libraries in the R programming language. If I cannot find an alternative in python, then I will use R for detecting change points.

The tasks involved in this project will easily meet the 20-hour threshold for a group of one. Table 1 below lists projects tasks with estimated times to accomplish them. The bulk of the work will be gathering and processing the data. I listed multiple tasks for computing similarities and different techniques for detecting change points. It is not necessary to accomplish all varieties – just one of each, however, if there is time, then there are additional plans to extend the project.

Table 1: Task List

Task	Time Range Estimate (Min-Max)
Download transcripts	1-2 hours
Parse transcripts	6-8 hours
Construct time series based on cosine similarity	1-2 hours
Construct time series based on Jenson-Shannon divergence	1-2 hours
Create visuals of time series	1-2 hours
Implement Bayesian change point detection of Barry and Hartigan (1993)	1-2 hours
Implement the nonparametric, hierarchical divisive algorithm of James and Matteson (2015)	1-2 hours
Compare clusters from proposed segmentations with baselines	2-3 hours
Substitute TF-IDF weighting	1-2 hours
Substitute Okapi-B25	1-2 hours
Generalize code so can be ran on all lectures	2-3 hours
Create documentation	1-2 hours
Prepare presentation	1-2 hours
Total	**20-34 hours**

References

Barry, Daniel and Hartigan, J. A. 1993. “A Bayesian Analysis for Change Point Problems.” *Journal of the American Statistical Association,* vol. 88, no. 421, 1993, pp. 309–319.

James, Nicholas A. and Matteson, David. “ecp: An R package for nonparametric multiple change point analysis of multivariate data”. *Journal of Statistical Software*, vol. 62, no. 1, 2015, pp. 1–25.

Wang, Yuli and Goutte, Cyril. “Real-time Change Point Detection using On-line Topic Models.” Association for Computational Linguistics, 2018, pp. 2505-2515.