Coursera Transcripts Analyzer
Final Documentation

Project topic: Coursera Transcripts Analyzer - Topic Modeling for Course Transcripts
Team: Free Topics Team
Team Leader: Dajun Lin (dajunl2@illinois.edu)
Teammates: Tsz Yui So (tyso2@illinois.edu), Amy Zhao (yiminz3@illinois.edu)

Github repo: https://github.com/alexlin1822/410CourseProject.git
Application link: https://coursera.streamlit.app
Demonstration video link:https://github.com/alexlin1822/demo410.mp4

1. Overview of the project

Coursera is getting increasingly popular; there are a substantial number of courses on Coursera. And each course in Coursera has a lot of knowledge points. Our app, a web based application, can compute the weight of various important chapter topics in a course, to determine some important knowledge points. These analysts' results could help users focus on the key knowledge points of the course. This is the purpose of our project.
In addition, our app can also analyze different courses at the same time to reveal whether there are similar knowledge points between different courses, for example, whether there are overlapping knowledge points in applied machine learning and data mining courses. This can help users use their own knowledge to better understand and learn new knowledge.

2. Overview of structure and functions of our application

Simply, the core of our application is a topic analysis model because our project is to analyze Coursera transcripts to extract topics to extract a list of main topics included in the video. It is a web based application with python.
Specifically, we use Python with NLTK and Gensim to perform topic modeling and identify which topic is discussed in a Cousera course transcripts. In particular, we apply the LDA (Latent Dirichlet Allocation) topic modeling technique to convert a set of Cousera courses to a set of topics.

Finally we utilize pyLDAvis package which extracts information from a fitted LDA topic model and visualizes topic clusters to help users interpret the topics in the model that has been fit to a corpus of text data.

Libraries:
Spacy, Gensim, Nltk, pyLDAvis, Pandas, Numpy, Streamlit

3. How the software is implemented

Our application is a web application and hosted on the Streamlit app framework. The main application code is in the app.py file.
There are three main steps in the process. Including text cleaning, topic analysis, and display.

A. Text Cleaning

Specifically, we use a function to the text content of the courses. Then return a list of tokens.
Second, for getting the root word, we used WordNetLemmatizer. And the NLTK's Wordnet is used to find the meanings of words, antonyms, synonyms. We created the function to implement them.

B. Topic Analysis

This is the core step of our application. The latent Dirichlet allocation (LDA) model is the main model used in our application. After text cleaning, we send the text content processed in the previous step to the LDA model. The model will find 5 topics from the text content and return the result.

C. Display and interaction

Firstly, the text result will be listed below the search bar. The result contains high-frequency words and their probabilities, as well as calculation formulas.
To allow users to view the results more intuitively, we applied pyLDAvis. This component generates an interactive chart.Through chart and interactive operations, users will not only easily know the most relevant vocabulary of the analyzed text, the topic, but also, intuitively feel their importance through the size of the bubbles. This is especially useful for users.

4. How to use our Application

A. Open the link https://coursera.streamlit.app in your browser.

B. Input the course number in the text input box. The course numbers range from 1 to 11. (For demonstration, we imported the transcript of 11 courses to the system)
C. Then the 5 most relevant topics and the graph for visualizing the topics will be displayed on the browser.
D. For example, by entering “1” in the course number input box, we get to see these top 5 topics selected by the model. And the bubbles visualize the topic clusters for user interpretation.

Input your course number here and press enter (from 1 to 11):

```
1
```

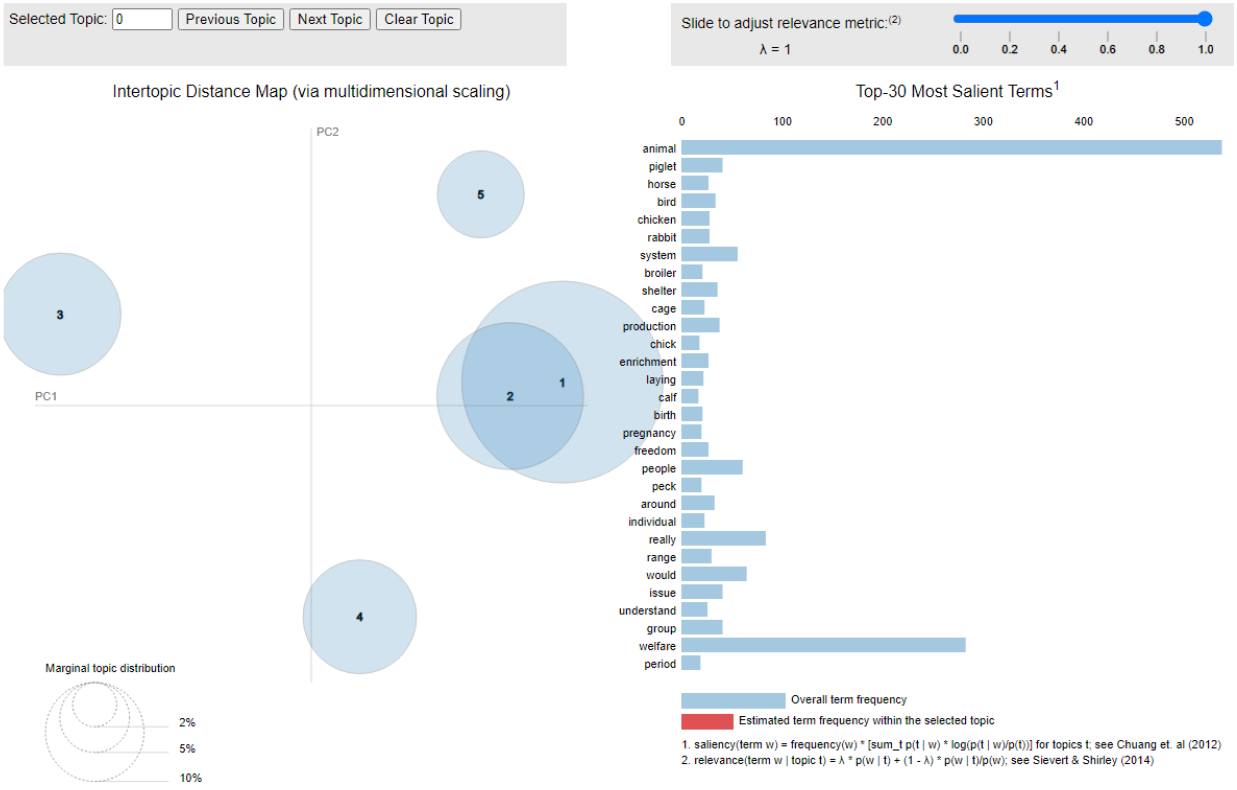
```
(0, '0.017*"horse" + 0.009*"welfare" + 0.008*"behavior" + 0.007*"people"')

(1, '0.013*"system" + 0.013*"bird" + 0.012*"chicken" + 0.010*"production"')

(2, '0.020*"piglet" + 0.012*"rabbit" + 0.012*"welfare" + 0.008*"calf"')

(3, '0.063*"animal" + 0.031*"welfare" + 0.010*"different" + 0.008*"behavior"')

(4, '0.034*"animal" + 0.013*"welfare" + 0.010*"behavior" + 0.010*"shelter"')
```



5. Contribution of each team member

Dajun Lin: Build a web app using streamlit, Reformat the transcripts, and refactor the code to use multiple documents to form a corpus, deployment onto Streamlit
Tsz Yui So: Build a web app using streamlit, Use pyLDAvis to visualize topics, perform testing
Amy Zhao: Write documentation, prepare powerpoint slides, perform testing