Guess the Class

Winter 2021 CS105

By: Richard Duong, Shreya Balaji, Benson Wan

Introduction

Overview

In our project we create a program that can identify what the class subject is based on a given video transcript and explore the relationships between words used in different classes. We gathered videos from YouTube based on specific class subjects under various majors such as psychology, computer science, political science, biology, english, math. We trained the data with machine learning algorithms.

Techniques and Classes Used

Our program trains the data with machine learning algorithms with a focus on KNN testing. Afterwards, we'll run tests to see if our trained sets are producing expected answers and analyze the dataset for surprising results.

Member Contributions:

We all contributed to the project.

- Data Collection: Richard, Shreya, Benson
- Data Preparation: Richard, Shreya, Benson
- Data Cleaning: Richard, Shreya, Benson
- KNN: Richard
- Report: Richard, Shreya, Benson
- Graphics: Richard, Shreya, Benson
- Data Analysis: Richard, Shreya, Benson
- Presentation Slides: Richard, Shreya, Benson

Project Scope

Type of Data Used

_____We will be selecting class lectures posted on Youtube. The class lectures will have either English captions or auto-generated English captions. These lectures are publicly available and will not require any additional hoops to use. We are downloading lecture series that are stored in playlists. To limit the inconsistencies of documents and ensure a strong accuracy score, we will only choose playlists that have lectures within the range of 30 minutes - 2 hours. The captions that are downloaded will be transcriptions generated by the pytube library which requests data YoutubeAPI.

Details on Data

_____We identified 150 topics from 13 subjects to find captions for. For each of the topics, we found a matching playlist. As a result, we collected about 4,000 video captions. However, we will be adjusting our dataset to 2600 captions by using 200 captions from each of the 13 subjects. This is because KNN is sensitive to uneven amounts of data from each category. In our complete dataset, we have a lower end of 220 political science captions, and an upper end of 470 psychology captions. We will be using the same 2600 captions for exploratory data analysis. In our revised dataset, we have 124,000 unique terms and 18,000,000 words in our dataset.

Breakdown of dataset:	Selected Subjects:			
• 13 subjects • 150 topics • 2600 captions in dataset • 124,000 unique terms (features) • 18,000,000 words in total	BIO	CS	MATH	PSYC
	BUS	ECON	PHIL	
	CHE	ENGL	PHYS	
	CHEM	HIST	POSC	

Project Design

Data Collection

Our dataset is made up of Youtube videos. Youtube videos of recorded lectures were collected from various subjects under different majors. To ensure that there was a good spread of videos to represent each major we asked other students for suggestions on classes to look into.

We maintained labels and links to all youtube playlists which can be referred to in the Official Playlist Links in Data folder. To convert our data from audio to text we made a function using pytube that takes in videos or playlists and saves the transcript for each into a JSON file.

Data Cleaning:

After the data collection phase was completed, we moved onto cleaning the data in order to prepare it for our machine learning model. Our data (the collected youtube captions) contained punctuation, video timestamps, video actions displayed through text, and text dictating the speaker, all of which are unwanted content because they wouldn't actually help make our model more accurate. In order to remove these pieces of unwanted information, we wrote a python program that scanned over each unfiltered caption transcript and replaced all the unwanted characters/words with spaces. This was done with the use of regular expressions so that specific characters/words could be detected in an efficient manner.

Another method we used to clean our collected data was converting all alphabetical characters to lowercase. The purpose for this was to ensure that our machine learning model wouldn't be case sensitive towards words that were the same but had different letter casing. With this data cleaning method implemented, all edge cases concerning letter casing were covered.

Caption transcript before cleaning:

```
1
00:00:00,000 --> 00:00:05,281
Hello there. I'm learning about python!

2
00:00:05,378 --> 00:00:10,339
But sometimes, it is challenging.

3
00:00:11,889 --> 00:31:00,000
[Music]

4
00:31:00,991 --> 00:45:00,012
Professor: Let us take time to meditate
```

Caption transcript after cleaning:

```
hello there i'm learning about python

but sometimes it is challenging

let us take time to meditate
```

Data Vectorization:

To prepare our dataset, we used 2 different vectorization models. For exploratory data analysis, we used a bag of words model. This model helps us count the frequency of words in all documents belonging to a label. Bag of words is not a good model to use with our KNN classifier. Bag of words is preferred to be used with a Naive Bayes model instead. We instead used the TF-IDF model which produces scores based on how often the term shows up overall multiplied with how selective the term is among all the documents. To limit what features we would vectorize, we removed insignificant words using nltk's stop words list. Stop words are words that have no significance and show up in huge amounts. Examples of this are articles and pronouns. We also simplified our feature vectors using the Snowball Stemmer (Porter2 Stemmer). The stemmer helps us group words with similar stems. For example, the words "jumping", "jumper", and "jump" all have the same stem. Instead of being 3 different features, we would group all 3 of these to "jump".

Data Analysis:

We did exploratory data analysis using the bag of words model. In essence, the word analysis was mainly done by grouping together words in a subject that fit a certain condition. For example, to identify how many dates were used in a subject, we built a regex that would identify dates, which would return a boolean value. We built lists of words that matched this regex, and returned its length, which would give us the count of dates for that given subject.

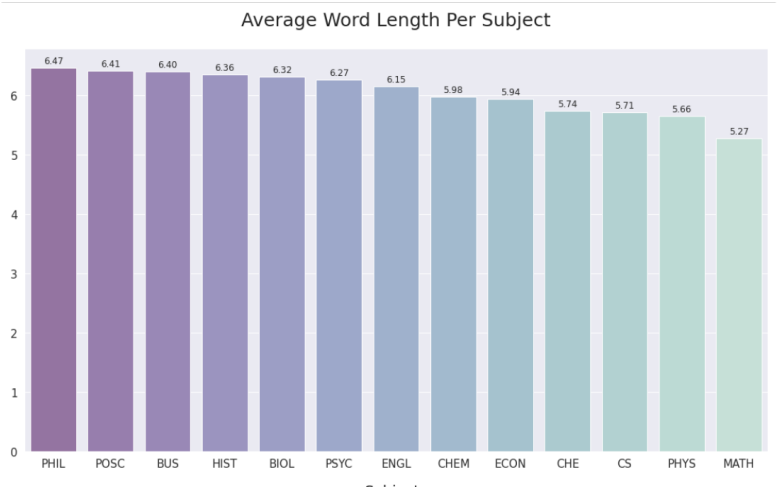
When building the predictor using supervised learning algorithm KNN, we had to use the TF-IDF model for higher accuracy. To build the training model for KNN, we divided our dataset into a ratio of 1:9. We used stratified sampling to select 10% of captions from each subject, to be part of the testing group. The remaining 90% would contribute to the training model. We fit the training model, and transformed the test data to see what our classifier would predict. The transform of the test data was compared against predictions to see how accurate our classifier was.

	total	0	1	2	3	4	5	6	7	8	...	10	11	12	13	14	15	16	17	18
BIOL	20	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	...	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL	BIOL
BUS	17	BUS	BUS	BUS	BUS	BUS	PHIL	BUS	BUS	BUS	...	BUS	BUS	BUS	BUS	BUS	PSYC	PSYC	BUS	BUS
CHEM	19	CHEM	CHEM	CHEM	CHEM	CHEM	PHYS	CHEM	CHEM	CHEM	...	CHEM	CHEM	CHEM	CHEM	CHEM	CHEM	CHEM	CHEM	CHEM
CS	20	CS	CS	CS	CS	CS	CS	CS	CS	CS	...	CS	CS	CS	CS	CS	CS	CS	CS	CS
ECON	19	ECON	BUS	ECON	ECON	ECON	ECON	ECON	ECON	ECON	...	ECON	ECON	ECON	ECON	ECON	ECON	ECON	ECON	ECON
ENGL	14	ENGL	ENGL	ENGL	ENGL	PSYC	ENGL	ENGL	CHE	BUS	...	CHE	PSYC	ENGL	ENGL	ENGL	ENGL	ENGL	ENGL	ENGL
HIST	17	HIST	HIST	HIST	HIST	HIST	HIST	ENGL	HIST	HIST	...	PHIL	HIST	HIST	HIST	HIST	HIST	HIST	HIST	HIST
MATH	20	MATH	MATH	MATH	MATH	MATH	MATH	MATH	MATH	MATH	...	MATH	MATH	MATH	MATH	MATH	MATH	MATH	MATH	MATH
PHIL	18	PHIL	PHIL	PHIL	PSYC	PHIL	PHIL	PHIL	PHIL	PHIL	...	PHIL	PHIL	PHIL	PHIL	PHIL	PHIL	PHIL	PHIL	PHIL
PHYS	16	PHYS	PHYS	PHYS	PHYS	PHYS	PHYS	CHE	MATH	MATH	...	PHYS	PHYS	PHYS	PHYS	PHYS	PHYS	CHE	PHYS	PHYS
PSYC	20	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC	...	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC	PSYC
CHE	17	CHE	CHE	CHE	CHE	CHE	CHE	CHE	CHE	CHE	...	CHEM	CHE	CHE	CHE	CHE	MATH	CHEM	CHE	CHE
POSC	17	POSC	POSC	POSC	POSC	POSC	POSC	POSC	POSC	POSC	...	POSC	POSC	POSC	POSC	PHIL	POSC	PHIL	POSC	POSC

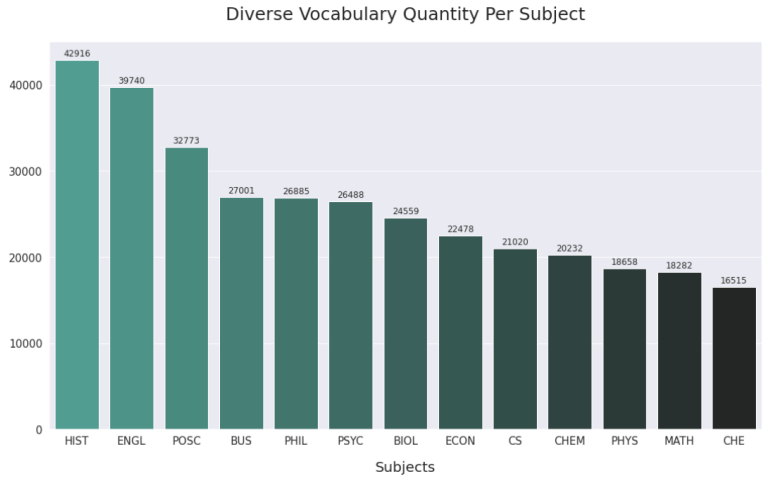
Exploratory Data Analysis/Results:

Generic Questions on Data:

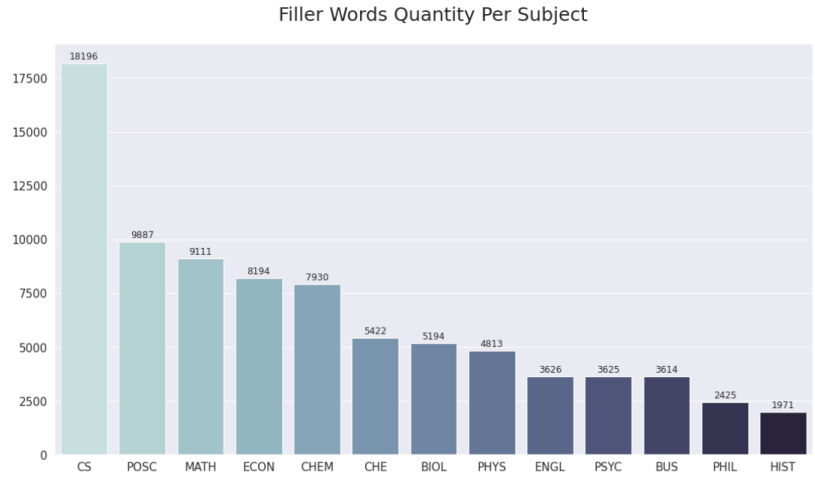
- Which subject has the longest words?
    - Philosophy



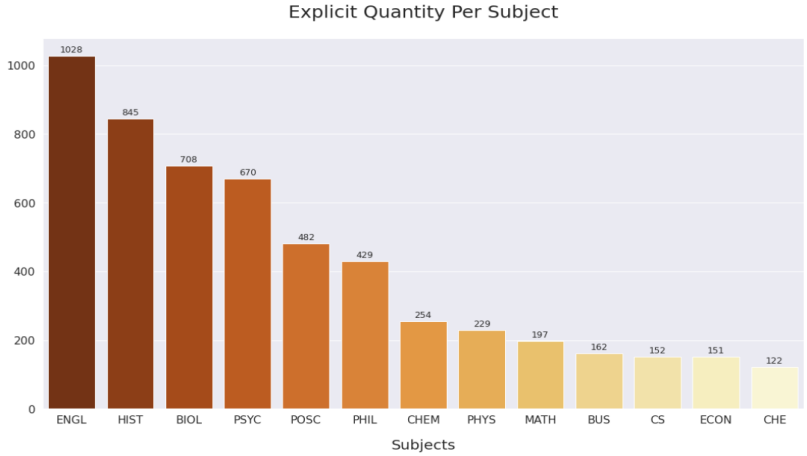
- Which subject has the most diverse vocabulary?
    - History



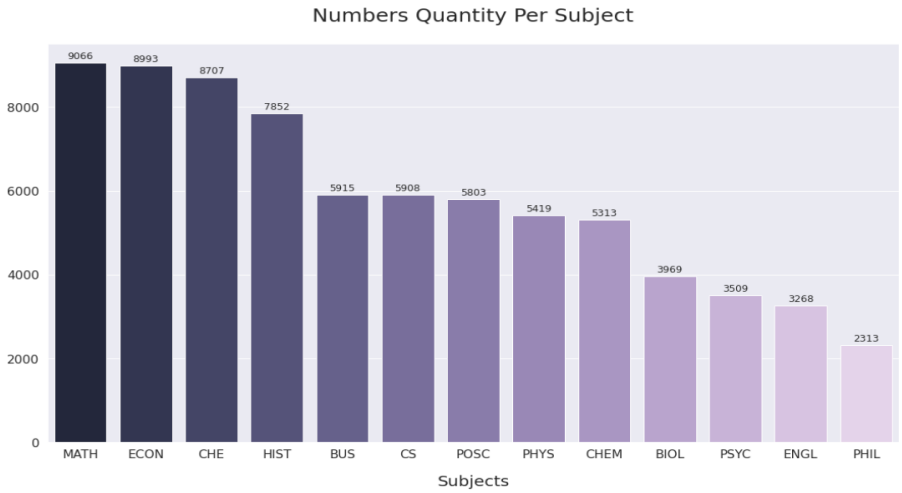
- Which subject has most filler words(ah, um, uh)?
    - Computer Science



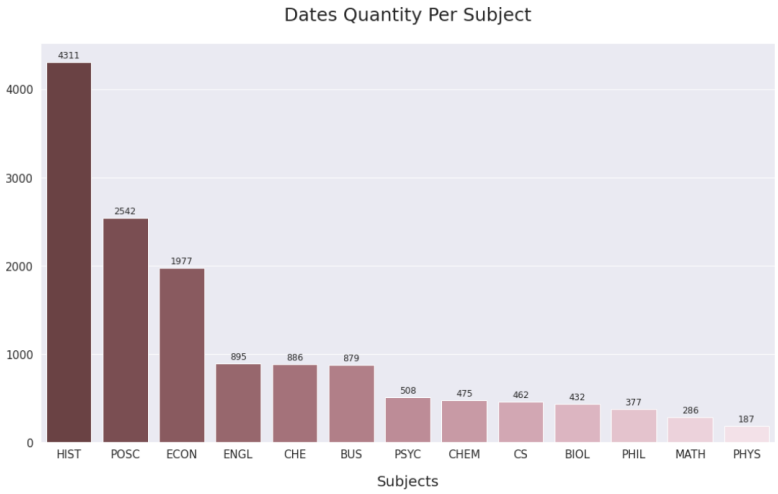
- Which subject has the most bad words?
    - English



- Which subject has the most numbers?
    - Math



- Which subject has the most dates?
    - History



Accuracy of KNN Model:

_____Using KNN our best model (Model 4) can predict a class with an accuracy of 90.77%. There were four classes that had a large effect on the accuracy of this model. Chemical & Environmental Engineering and Physics were often categorized as Chemistry due to how similar those classes were. English and Political Science had an inadequate data set; it was difficult to find a variety of classes for those lectures. Without those 4 classes (English, Political Science, Physics and Chemical & Environmental Engineering) we came up with a model (Model 6) that had a 96.11% accuracy. However, because that model was not an accurate representation of all the subjects in our dataset we could not use that model.

Error Analysis:

_____We made several decisions to mitigate error in our data selection. Knowing how sensitive KNN is to outliers and a poor distribution of data, we made an effort to pick an equal number of captions under each class. We also aimed to cover a wide variety of topics for each class, rather than exclusively choose introductory courses. That way, we would be able to reach out to obscure classes under that topic as well.

Our data selection also has areas that increase area as well. For example, the length of videos in our selected data varies wildly between 30 minutes to 2 hours. This may skew some of our data, if for example, most physics playlists are 2 hours long, and most english classes are 30 minutes long. Another issue is that we used a combination of english captions and auto generated captions. Auto generated captions can make a lot of mistakes with professors who have an accent, which would skew the data even more.

Bias and Variance:

Our model has very high variance with the dataset we've selected. We purposefully aim to grab videos from a wide array of topics to generate obscure terms from each subject. As a result, we have roughly 180,000 features to use. To select our testing and training sets, we used stratified sampling to ensure that our models still maintain high variance. In one of our models, we also vectorized using both unigram and bigram models. This generates twice as many vectors and increases the variance in our data.

Our model also has areas where we've introduced bias. Our feature vector is very tough to do calculations on if we left it at 180,000 features, so we did things to reduce the dimensions, and increase bias. For example, our method of cleaning introduces bias. Regular expressions are undoubtedly very tough to perfect, so there is a high possibility we may have also filtered out significant data. Stemming bins features together which also increases bias.

Conclusion:

Overview:

After implementing and training our model with the K-Nearest Neighbors algorithm, filtered transcripts, and multiple trial runs using different parameters, our fourth model peaked with an accuracy rating of 90.77%. The specific parameters that allowed us to attain this rating was a focus on unigrams and bigrams, requirements for a word to appear more than 10 times throughout the corpus while appearing in less than 75% of all transcripts, and a k-value of 20. Furthermore, our sixth model actually acquired an even greater accuracy rating at 96.11%, but we weren't able to use this as the finalized model because 4 subjects had to be removed from the data set due to issues impacting the prediction results. Therefore, our fourth model is the most accurate seeing that it received the highest accuracy rating with no subjects left out during training with KNN.

- Tools used: python, pandas, pytube, matplotlib, seaborn, scikit learn, jupyter notebook, regex (regular expressions), multithreading, git, pickle
- Methodologies used: k-nearest neighbors, stratified sampling, snowball stemmer, tf-idf, unigram & bigram models

Areas for Improvement:

- Do not pick auto-generated captions for higher accuracy predictions.
- Pick a higher quantity of playlists for each topic within a label.
- Partition the corpus into sections instead of entire video transcripts.
- Run k-fold validation testing to improve model accuracy.
- Improve sampling methods so that more distinct courses are chosen.

Our Project:

1. Github Repository
2. Data Sheet
3. Presentation

Resources:

4. Data Pre-processing and Dimensionality Reduction Techniques
5. Annotate Bar Graphs using Matplotlib
6. Bag of words Model
7. K-Nearest-Neighbors
8. Pytube API Documentation
9. Seaborn Documentation
10. Seaborn Graphs
11. Porter Stemmer vs. Lancaster Stemmer
12. Stemming/Lemmatization
13. Stemming support for sklearn Vectorizer
14. TF-IDF
15. TF-IDF Tutorial