Course Syllabus Analysis NLP Project

Alex Glover

Introduction

For most students in university the initial exposure they have with a new course at the beginning of the semester is the syllabus provided by the instructor. The syllabus communicates what topics will be covered in the course, the learning outcomes intended with the material, and other details that vary based on what the instructor chooses to include. The densely filled document prepares students for what they will experience for the duration of the course's scheduled time.

Problem Statement

University level course syllabi describe the course material and associated expectations to establish mutual understanding between an instructor and their students at the beginning of the semester. This project aims to read through the content of course syllabi, identify meaningful and important phrases within the text, expand the important phrases using open source knowledge maps, leverage unsupervised association rule learning to quantify relationships between concepts represented across documents, and train supervised machine learning models to predict the amount of academic rigor required to be successful in a given computer science or information science course at ETSU.

Motivation

Many students read the syllabi provided by their instructors at the beginning of the semester to prepare and consider what life will be like for the next few weeks or months while participating in their courses. Anyone who has attended university would agree that some courses are more demanding than others due to a number of factors. The type of assignments, number of assignments, modality of instruction, and complexity of content are just a few factors that contribute to how much academic rigor the average student will need to exert to be successful. I think it would be interesting to explore what processes would be necessary to be able to read a course syllabus and determine that level of effort in preparation for the semester.

Research Questions This Project Answers

This project is a foundational exploration to determine what processes are necessary to extract content from university level computer science and information science course syllabi and transform it into a meaningful feature set for supervised machine learning applications. University syllabi, especially in technical domains, are written with language that is not restricted to the vernacular most people are comfortable using daily. They are succinct by design which limits the presence of interchangeable terms for topics. These characteristics create additional challenges deriving meaning from the text and identifying commonalities between documents. An ideal feature set for the mission of predicting the average student's experience in a course would be a feature set that encapsulates the content a student is exposed to, the expectations of what a student does with that information, and additional factors that the syllabi may or may not share. In this project I want to identify roadblocks in building a meaningful feature set describing these courses with the limited and varying disclosure a syllabus provides.

Literature Review

Knowledge maps are structures trained on large sources of text to identify relationships between terms. These relationships vary from synonyms and antonyms to relevant domains using terms or behaviors associated with terms. Human language is complex and abides by different rules depending on the particular language, but the mission of conveying information remains. ConceptNet is an open sourced knowledge graph designed with the mission to be a source providing understanding of language used by the general population [cite here]. It is trained with a variety of sources, the largest being Wiktionary. The core relationships represented by the graph include symmetric relations such as "related to" and "similar to". Asymmetric relations they recognize include "causes", "made of", and "used for".

Unsupervised association rule mining is a process of identifying patterns within a dataset that have not been specified by a user. The apriori model is an example of this concept that analyzes lists of items to determine the likelihood a term would fit in the context of a list of items

Dataset Description

University course syllabi share the purpose of describing the content of the course, but instructors also take certain liberties to expand beyond the minimum provided content. All syllabi provide a brief description of the course content, the learning outcomes, required materials, and standard disclaimers of disability services and protocols for academic

misconduct. Many instructors also provide sections such as the types of assignments they will assign, the number of the various assignments, the weight of assignments contributing to the overall grade calculation, and the schedule for the course. The order of content, font, and template/format of the syllabus varies by instructor and course. All the syllabi in my dataset are in PDF format and vary in page length from 1 page to 18 pages.

Project Approach

Data Collection

The Department of Computing provides a dropbox shared by instructors dedicated for their course syllabi of the semester for sharing publicly. The Fall 2024 semester offers 61 unique courses. By the time I had access to this dropbox it contained syllabi for 68 courses, and I added the syllabus for CSCI 4037 and CSCI 5037 as well. I removed the syllabi pertaining to BlueSky because of the limited representation of it in the dataset, leaving my dataset to consist of 65 unique course syllabi.

This project utilizes supervised machine learning, so I needed to have associated ratings of academic rigor for each course in my dataset. No standardized metric currently exists to approximate the amount of academic rigor a student exerts in a semester-long course. Meaningful approximations could be developed over long periods of time with student surveys and additional data-driven considerations guided by research, but that is outside the realm of feasibility with the time and resource constraints of this project. The primary focus of this project is selective feature set construction with concept mapping expansion, so I determined a suitable and accessible measurement for this project would be to ask the educated opinions of the instructors that teach these courses.

After obtaining IRB approval, I sent an email to every instructor in the department with a course included in my dataset as a one-question survey. I asked instructors to rate the courses they are teaching this semester as an integer on the scale from 0 to 100 approximating how much academic work is required to be successful in their course. In this scale a 0 means students receive an A in the course simply by enrolling in it with no work done. A 100 means the course is the most difficult course in a student's degree program and requires as much work as a full-time job. I asked instructors to provide their answer considering the perspective of the average student in their course while also considering the context of the demands involved of other courses in programs their students typically are registered in.

Data Cleaning

The text extraction from PDF process for this project was complicated for a few reasons. The order, inclusion, and presentation of content varies greatly in this dataset which limited automation options. Initially I used the pdfplumber python library to extract text from the PDFs. This library extracted text with very high accuracy but reads all content horizontally. In any instance where a syllabus uses a multi-column format, offset textbox, or any presentation other than single-column, it mixes content from different blocks of text. I then tested the pytesseract python library, which intuitively groups texts in a variety of formats very well. Unfortunately that library generates hallucinations and artificial typos not present in the original text. This occurrence varies based on the font and template a given syllabus is formatted with, but far too often for manual correction to be feasible.

My solution to this problem was to leverage the strengths of both libraries together. For every page a document I used both libraries independently and tokenized both. Using the tokenized output from the pytesseract library, which is grouped appropriately, I cross-referenced each token sequentially with the tokenized output from pdfplumber. If an exact match for a pytesseract token was found in the pdfplumber token set, the token is assumed to be correct and the match is removed from the pdfplumber token set. If no exact match exists, the first token found in the pdfplumber token set to meet or exceed the similarity threshold of 0.8 from the difflib get_clos_matches function is assumed to be the correct version of the assumed pytesseract typo. If no pdfplumber token meets the similarity threshold, I kept the token to manually check and verify if the token is meaningful or a hallucination. This method worked incredibly well, reducing the number of manual corrections I needed to make by approximately 75% which was feasible. The persistent mistakes remaining in the text was confusing a stand-along capital “i” with the pipe character and replacing “ai” with “al”.

With clean text files of syllabi content, I preprocessed the text to lowercase, removed all punctuation (with the exception of # and -), and removed all numbers. I then replaced all instances of the token “dl” with “d2l” to maintain meaning and removed all tokens with a length of 1 to resolve the pipe issue since no real meaning was compromised with this action in this project. Finally, I removed instances of tokens repeating back-to-back which occurred due to previous preprocessing steps or table interpretations that do not provide meaningful information for my purposes.

Feature Extraction

After cleaning and preprocessing, I implemented TF/IDF vectorization on my dataset to extract n-grams. I extracted the top 25 n-grams for each document and reviewed them. A previously mentioned complication with this dataset is that content order and inclusion varies.

All of these syllabi include certain sections of information that provide no value to this project: primarily the disability accommodation and academic misconduct sections. Additional content and language exists throughout syllabi that provides little to no value in this project such as mentions of students or days of the week. The variety of order and inclusion of content made section selection based on relevance infeasible for this project, so I opted to specify persistent irrelevant terms in a custom stopwords list instead. The resulting list of n-grams derived from the collection of documents contains far more relevant terms describing the content of the courses than terms of little value.

With these lists of important terms and phrases for each document, I connected to ConceptNet with their web API. Their API allows you to submit a term or phrase and specify a type of relationship they support and recognize to return terms and phrases related to your provided term. For every term in my collection I extracted terms in English with the following associations with a limit of 5: related to, synonym, part of, used for, capable of, causes, has property, defined as, manner of, has context, similar to, and receives action. Most terms in my dataset do not have return values for all these associations, but they have values for at least one if not more. The results from this connection are interesting. Some returns from the API expand the context of the course very well, such as expanding “network security” to include “threat” and “protective”. Others have no relevance to computer science or information science, introducing terms such as “Chinese mythology” and “professional wrestling”. Unfortunately, ConceptNet does not have a large amount of training on terms specific to the computer science domain. Expansion using ConceptNet would likely return relevant terms for a project like this better if the domain was university courses in political science or business as opposed to technical fields.

Feature Selection

The N-grams and ConceptNet expansion terms become features inside of the feature set describing the courses in the dataset. The relationships between the selected terms / phrases and the courses need quantifiable representation in the feature set to become meaningful to machine learning algorithms. One option to accomplish this would be a binary association where the relationship between a term and a course is 1 if that term is included in the course term set and 0 if it is not included. This method would accurately represent which terms appear in the course set, but provides no ability to represent relationships between courses and terms similar but not included in the set. Degrees of relationships between terms provide more insight into the similarities between courses. The apriori unsupervised association rule mining model quantifies the relationship between terms with the metrics support, confidence, and lift. I ran this algorithm through every one-to-one permutation of the set of terms derived from the corpus to generate the supported metrics. I then created a “Relevance” metric which is

confidence multiplied by lift. This metric evaluates the likelihood of cooccurrence for the two terms and either amplifies or condenses that value to account for the level of correlation present. These values range from 0.004 to 65. For each course I use the list of associated terms and for each term I calculate the average relevance score when the "base" value is equal to the course terms and the "add" value is the term in the feature set. No score is calculated for instances of the "base" and "add" values being equal in the apriori algorithm.

I implemented two different approaches to ensure representation of instances where the term is included in the document term set. One method is adding 130 to the list of relevance terms to be averaged. I chose this value because it is twice the value of instances where the "base" term and "add" term always appear together, so it signifies the addition of a disproportionately important term in the set. Another method is to multiply the overall average by 2 to further amplify the signal of importance. I applied this to the dataset of only the n-grams derived from the syllabi and the dataset expanded with ConceptNet to test the feature set construction that a supervised regression model can learn from best.

Choosing to work with target values obtained through survey responses from instructors exposed this project to the risk of a lack of responses. Unfortunately out of the 65 courses in my dataset I only received responses for 23 courses. Considering how small the original dataset is, reducing the dataset to 23 courses would limit the content too much. I compensate for the lack of responses with K-means clustering. After performing TF/IDF vectorization I clustered the documents with K-means clustering. I identified the k value that results in the highest silhouette score while insuring all clusters contain at least one document with a rating target value. For all documents that do not have a rating value I assigned it the average value of the assigned documents within its cluster. This strategy preserves the limited dataset for preliminary exploration, but is an additional factor that will skew results that should be considered in the results.

Model Building

Infinitely many supervised regression architectures exist that can learn from a dataset constructed in the manner of this workflow. The primary experimentation for this project is on feature set construction as opposed to supervised learning, so I chose to create a single architecture to train and test with all 4 feature sets to evaluate performance. I also chose to implement leave-one-out-cross-validation (LOOCV) because the number of observations is quite low but the number of features is quite large. LOOCV allows the maximum preservation of context during training and my limited number of unique courses makes the computational overhead of the method remain feasible. I created a simple MultiLayer Perceptron model through the TensorFlow library with 4 layers using the Adam optimizer, the mean squared error loss function, and 100 epochs of training. The parameters and hyperparameters of this model

are not tuned for any feature set in particular because it is very likely that each of these feature sets would require varying parameter and hyperparameters for optimization. Variations in these would interfere with evaluating the performance of the different feature sets. Additionally, the target values for this dataset rely on too much individualized subjectivity and approximation for accuracy to be prioritized. The order of the resulting values is an interesting indicator of model performance.

Results and Insights

The initial model evaluations of mean squared error, root mean squared error, and mean absolute error indicate that the feature sets most intuitive to learn from in order from most to least are: Expansion, N-grams, Expansion with Multiplier, and Ngrams with Multiplier. All three metrics follow this pattern. This indicates that the more subtle method of appending 130 to the list of values to be averaged in cases of the feature term appearing in a course's term set is a more intuitive approach than the amplified multiplication method. While the expansion terms provided by ConceptNet contained more irrelevant terms than desired, the feature sets that include the additional terms are more intuitive for the MLP architecture in use than the N-grams derived from the syllabus alone.

Across all four feature sets the courses rated by instructors tend to result in errors of greater magnitude than the courses with artificial ratings derived from the K-means clustering process. This result is logical considering the instructor ratings are subjective while the clustered ratings are derived from computation based on subjectivity. This outcome indicates the patterns identified in the clustering process are similarly identified by the MLP model from the constructed feature sets.

The MLP predictions from the expanded datasets yield values in closer proximity for a given instructor than the predictions from the n-gram datasets. The predictions vary between feature sets. All of the predictions rate the Research Methods graduate course as one of the least demanding courses in the list, which is interesting because, anecdotally, that course is considered one of the most challenging and demanding courses in the department among students. The Natural Language Processing course is consistently predicted to be one of the most demanding in the dataset.

Answers to Research Questions

This project provides insight into several of the initial roadblocks and potential for further development for constructing a feature set describing a university course from its syllabus. The first roadblock is the construction of the syllabi. Several features I planned to include at the start of this project had to be abandoned because not all syllabi disclose the

information. The types of assignments, quantity of assignments, and weight distribution contributing to the final grade calculations are incredibly influential factors in the amount of work a student must exert in a course. Many instructors do not include some or all this information.

While more work must be done to determine the best feature set construction for this type of project, the results indicate the use of knowledge mapping to expand the dataset is beneficial even in domains like technology with which the knowledge map has limited training. The apriori model provides meaningful quantification of relationships. A roadblock I encountered with that algorithm is deciding whether to only use one-to-one relationships from the model or to expand the number of terms allowed in the "base" input. I restricted the use to one-to-one relationships for this project, but experimentation with alternative numbers of items as base inputs have the potential to produce values with more insight.

Conclusion

Construction of feature sets that describe text for supervised learning is a complex process with many options at every step. While the performance of the resulting model is not strong enough for use as-is, it demonstrates the benefits of the use of knowledge maps to expand datasets of succinct text for greater overlap. Different methods of quantifying relationships within a given set result in datasets of varying intuitive properties for deep learning models.

Future Directions

This project is a preliminary exploration into building intuitive feature sets derived from technical text. It would benefit greatly from use of knowledge maps trained on technical language. It would also benefit from a measurement of academic rigor that is less subjective.

I see this project as the foundation of more complex processes of analyzing content and core competencies in courses to determine the effectiveness of instruction for students. Identifying the skills students should be gaining from courses and the concepts they struggle with most can indicate if the foundational topics are being taught effectively and identify skills that require reinforcement for success. This work would require cooperation from instructors to share course material and knowledge maps trained on advanced technical material to adequately capture relationships between courses and the assignments given to students.

Things I Leaned From This Project

In this project I learned how to extract text from PDFs using a variety of formats. I learned about narrowing down documents into important terms without manually selecting sections of text. I also learned about using unsupervised association rule learning to quantify relationships. Overall, this project has made me much more familiar with text processing and problem solving.