KALLEM, SNEHITH REDDY. M.S. Model for analyzing course description using LDA topic modeling. (2022)
Directed by Dr. Somya D. Mohanty.

This study demonstrates a way to generate a Topic model using LDA (Latent Dirichlet Allocation) topic modeling for the courses of multiple universities in the USA, which is relatively significant. This model will specifically be able to differentiate the course structure between various universities, such as the University of North Carolina at Wilmington, the University of North Texas, the University of South Carolina, and the University of Western Carolina. This model will help find the related courses of a selected department of study, or so they thought. The LDA (Latent Dirichlet Allocation) topic model is used to infer topics from the content in the university course description. Further, this study showed how to generate a Topic model using LDA (Latent Dirichlet Allocation) topic modeling for the courses of multiple universities in the USA. This study will: Explain how to Infer topics from the corpora consisting of various universities' text of course details; Helps to find out the related courses of a selected department of study in a big way; Group the topics into different communities by calculating the Modularity with the help of the Louvain method; Analyze how the courses are related to the topics, for the most part subtly inferred for each University; For a selected Department of study, see what all courses belongs to this department with the help of topics generated. This study helps us to identify the courses which have a relation with a selected department of study. The graph representations mainly included in this paper will generally explain our Approach.

MODEL FOR ANALYZING COURSE DESCRIPTION USING LDA TOPIC MODELING

by

Snehith Reddy Kallem

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Greensboro
2022

Approved by

Committee Chair

APPROVAL PAGE

This thesis written by Snehith Reddy Kallem has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair ____________________________
Somya D. Mohanty

Committee Members ____________________________
Lixin Fu

____
Yingcheng Sun

____
Date of Acceptance by Committee

____
Date of Final Oral Examination

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Somya D. Mohanty, for his continuous support of my research, for his patience, and for his insightful comments and suggestions. I have also benefited greatly from discussions with my committee members, Dr. Lixin Fu and Dr. Yingcheng Sun , as well as from feedback from other faculty and students at the university of North Carolina at Greensboro Department of Graduate studies.

I would like to thank my parents for their support and encouragement throughout my studies. I would also like to thank my friends, who have supported me and listened to me throughout this process.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Topic modeling is a technique that helps to find the topics in a given data set and then can see the words most frequently occur in a topic[16][17]. In universities, a lot of data is in the form of courses, students, teachers, and universities. Topic modeling will help analyze these data and transform them into topics[17]. These topics will help compare the universities. So this is a beneficial tool for students looking for a course to take and want to compare courses of different universities carefully. For example, if a student wants to take a data science-related course, then they has to analyze how that course is related to other courses in the same department and how it differs. It is a beneficial tool for students who need clarification about the course. Topic Modeling lists topics from each university and how they relate to each topic. It will be easy to compare the similarities between the universities by finding the common topics and their weights. Also can discover how each university is different by finding its unique topics and weights.

Also, this topic modeling technique can be applied to other data types like news, social media posts, blog posts, etc., and explained better by checking out some examples published by the product users. Topic modeling can also be implemented on any website. In this, a user can search for any topic and get the related courses in one click. It will help the students get detailed information about the course topics. One can use this model for an academic purpose or any other purpose. This model can be used by any individual who is interested in the field of knowledge sharing and analysis. With the help of this model, users can easily extract information from the document's content.

This model will help to primarily identify different hidden topical patterns present across the documents, which are the course and its description[10]. With the help of this model, annotate documents according to the topics created as part of this Topic modeling. Later helps to organize, summarize and search based on the annotations made, or so they, for the most part, though.

1.1 Topic Modelling

Topic modeling analyzes the data and finds topics related to other courses. It helps to analyze the difference between that courses and find the topics different from other courses. There are many other use cases. Social media explores trending topics to see which are the most popular topics in social media, and topic modeling helps to analyze these topics and find the topics that are prevalent in social media[9]. Then see which are the most trending topics. In marketing, topic modeling helps to analyze the data, find the topics that are helpful in marketing, and see which marketing strategies are valuable and useful[12]. These are some of the use cases of topic modeling.

In Topic modeling, the words are not assigned a numerical value but are grouped into clusters of similar phrases. For example, "politics" and "politics essays" differ. They should never be grouped, but they are grouped in the topic model because they have similar words in them, e.g., "government," "policy," "politicians," etc. In this way, it helps to find out the topics of the documents without having to read them. The topic model is a statistical model used to find the most probable topics of a corpus of documents. It works by clustering documents together based on their similarities. The topics of a topic model are a set of words that are similar in topic.

An example of topic modeling would be, "In a study of top-ranking Russian politicians, researchers found that people with a higher level of education tended to have more liberal views of homosexuality, and people with higher levels of education tended to have more liberal views on homosexuality." Topic modeling can help in discovering the topics in a collection of data. It is a descriptive technique, meaning it is more of a hypothesis about the data rather than a definitive statement.

It is a probability-based technique that tries to map words to topics. It assumes that each piece of text can be mapped to a set of topics and each topic to several documents. It then tries to find the probability that each topic maps to the text, given each word's probability of occurring in that topic. The probability can be calculated by the topic distribution or estimated from the corpus. An algorithm will run to find the topics in the corpus and group them. For a topic analysis, the words in the corpus are assigned a probability of belonging to the topic[2].

Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), and Non-Negative Matrix Factorization are the current techniques for extracting topic models. Latent Dirichlet Allocation will be the main topic of this article (LDA)[3].

1.2 Basics of LDA Topic Modelling

Latent Dirichlet Allocation topic modeling is a well-liked technique to extract topics from a corpus. It is an Unsupervised clustering method that is frequently utilized for text analysis[10]. In this Topic modeling technique, the documents are modeled as a collection of word topics, with words being represented as topics. Considering an example for a better explanation of the LDA topic modeling, consider a collection of documents. Each document belongs to topics like Computer Science, Accounting, Music, etc. Additionally, some of the papers may belong to multiple topics. This model helps to sort documents into topics because there are only documents, not their topics.

By definition, LDA will arrange the documents according to their topics. LDA will group the documents based on topics like Computer Science, Accounting, and Music into a triangle because of assuming three topics with the topics at the corners. As previously mentioned, each document can belong to different topics with a weight to that topic. For instance, a record on Data Analysis may belong to Computer Science and Accounting topics. In statistics, such distribution is called a Dirichlet distribution and is defined by the parameter α[8].

The above example is explained using the LDA diagram below.

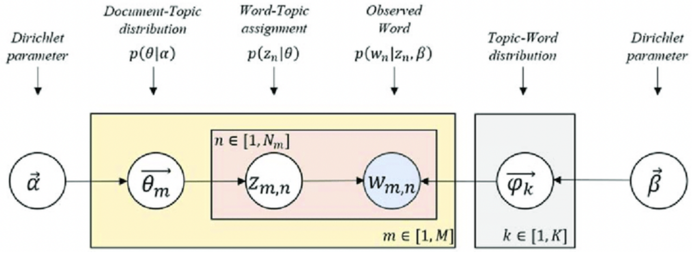

Figure 1.1. LDA Topic Modelling Explanation

From the above diagram, all the corpus is present in the yellow box, represented by M. For example, if they have ten documents, then $M = 10$. And the peach color box represents the document's word count and is characterized by N. Many words are in the peach box, $w_{m,n}$ located in the blue circle in one of those words. According to LDA topic modeling, every document is linked to a topic represented as $z_{m,n}$. Now, the θ symbol, which means the distribution of topic words in the corpus, is provided by the assignment of $z_{m,n}$ to a topic word in these documents[10].

1.3 Basic Definitions

To understand the LDA topic modeling and its Methodology, first need to know very few general definitions used in this topic modeling, contrary to popular belief. Below are the topics that should be known before going further in LDA topic modeling.

1.3.1 Beautiful Soup

The Python package subtly extracts data from XML and HTML files. It provides a natural means of traversing, searching, and altering the parse tree in conjunction with your preferred parser. It shows that it allows for a natural means of traveling, exploring, and adjusting the parse tree in conjunction with any sort of preferred parser in a subtle way. This library will significantly save hours while scrapping the data from an HTML file[13].

1.3.2 Gensim Library

Gensim is an open-source python toolkit representing documents as semantic vectors as quickly as possible for humans and computers. It processes unstructured texts using an unsupervised machine learning algorithm. It contains algorithms like Word2Vec, FastText, Latent Semantic Indexing (LSI, LSA, LsiModel), Latent Dirichlet Allocation (LDA, LdaModel), and others, automatically identify the semantic structure of documents by analyzing co-occurrence patterns within a corpus of documents. Any plain text document (sentence, phrase, word, etc.) can be quickly expressed in the new, semantic form after these statistical patterns have been identified, and any document's similarity to other documents (words, phrases, etc.) can then be checked.

It is used for machine learning, natural language processing, and data mining, to name a few use cases. It can be used for many purposes, from research results to data science projects. It is written in Python and runs on Linux/Mac OS X, Windows, and other Unix-like systems. This information can then be used for many applications, from search to social networking, analytics to research. Gensim is extremely fast and scalable and can handle enormous amounts of data, making it an excellent choice for data-intensive applications. In addition, it is easy to install, use, and extend[15].

1.3.3 Corpus

A corpus or a test corpus is a language resource consisting of many texts. A large enough and pertinent corpus of material is necessary to train a computer to understand word meanings. It is referred to as a corpus and simply a set of documents. In this paper corpus consists of the course descriptions of different courses in a university.

Such a test corpus can be used to evaluate the proposed algorithm's quality and compare different algorithms. For example, one of the most commonly used corpora in natural language processing is the web. A typical web corpus consists of millions of words. When anyone is looking up a word in Google, the first search result will most likely be a link to the search result for the query most closely related to the original question. It is a typical example of how a large text corpus is used to rank search results. One of the most popular web corpora is Google's search index. The other important aspect of the web corpus is that it is composed of millions of documents. Such a corpus can be used to evaluate the proposed algorithm's quality and compare different algorithms.

1.3.4 Bag Of words(doc2bow)

A bag of words represents the data while modeling the corpus within the machine learning algorithm. It is a way of extracting features from the text and a representation of text that describes the word occurrence within the document. It consists of a word and its measure of its presence in that document[17].

1.3.5 LDA and its process.

Documents are a mixture of topics, and topics are a mixture of tokens/words are the assumptions made by LDA. Documents are the distribution of topics, and the topics are the distribution of words. A corpus can also be represented as a document-to-word known as DTM(Document term matrix)[6]. The corpus is a document to the word matrix, in which each row is a document, and each column is the words/tokens. LDA now converts this matrix into a document term and topic word matrix.

The document term matrix contains the possible topics a document can hold, and the topic word matrix has the words that those topics can include. Finding the ideal document topic matrix and topic word matrix will enable LDA to determine the best Document-Topic distribution and Topic-Word distribution[6]

1.4 Related Work

Considering a paper on Online course recommendations by Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, and Jose Ochoa Luna, A model generated will, for the most part, be able to recommend available courses on sites like- Coursera, udacity, etc., which is quite significant. Now it will be possible to find courses from almost all knowledge domains, which is quite significant. Humans find out the course according to their background, which involves access to each platform, searching the courses, reading the course syllabus, and choosing very appropriate content in a particular major way—using LDA topic modeling, which infers the topics from the content given in a college course syllabus, literally shows that using LDA topic modeling which infers the topics from the content given in a college course syllabus, which is quite significant. Also, topics are extracted from online, generally open-source syllabi, which is fairly significant. Later these two sets of topics are matched using a content-based recommendation system and generally recommend the relevant online course, demonstrating how topics are extracted from an online sort of open-source syllabus, which is quite significant[1].

To recommend an online course, each course is considered a document to annotate. LDA is used to specifically identify the document's hidden semantic structure, which is generally relatively significant.

Chapter 2: Methodology

2.1 Data Scrapping

There are many courses available in each university for both graduate and undergraduate to have some data to generate an LDA model, which is quite significant. Pull the course catalog of each university into a data set with required columns by web scrapping the URL of the university undergraduate and graduate course catalog using the Beautiful Soup library in python. There are many courses available in each university for both graduate and undergraduate levels to have data to generate an LDA model directly, or so they, for the most part, though. For this, first, inspect the HTML structure of your target site with the help of the browser's development tools, contrary to popular belief. After understanding the URL structure and the HTML, pulling the required details of the courses and saved it in a .csv file, or so they thought.

2.2 Pre-processing/ Cleaning Data

Before profiling an application, one should ensure that the data is in working order. It means removing invalid or incorrect data and profiling your application to ensure running before it starts gathering profiling data. It will give the best results. Data has to be cleaned to be used in machine learning. This process can be done manually by entering information into a spreadsheet or importing it into a database. Once the data is in a format used in machine learning, it's called "data prep."This is the first step, where information is validated, cleansed, and transformed into a more useful form for analysis. For a successful examination, data needs to be prepared and standardized.

The data extracted can contain unwanted characters that need to be removed before creating a corpus, contrary to popular belief. For example, particularly few courses generally have prerequisites in the course description; in this case, split and remove it from the report, which is quite significant. One more instance is typically where the course id consists of both course department and course number. For the most part, split and mainly save it in a new column, which can be used to determine the course level in a big way. Following are very few Examples of data that need to be taken care of, or so they thought.

1. Remove the unwanted texts from the documents.
2. Replace '-' with a space.
3. Encoding the strings using ASCII values.
4. Removing the Stop words.

2.3 Tokenization Using Gensim

Now let's understand how this model works in detail. The data received is in a structured format, say a CSV or a TSV or a Parquet, or any other data format. There are specific rules that need to be followed to process that data. In the case of a CSV or a TSV, the first step is to split the data into separate columns. After loading data into the graph, it is divided into values and rows. These steps are essential as they define how the data will be processed. Data received in a specific format must be processed according to that format[11].

There are different techniques to do this. For example, it parses the sentence into words or uses a unique algorithm to break the sentence into smaller parts. Now, using the most popular one, known as the "n-gram" algorithm. It uses a statistical language model to create a parse tree for every possible combination of words in a sentence. It is important to note that these algorithms are imperfect and will make mistakes if the corpus of text data is vast. It uses different techniques to fix these mistakes, like correcting numbers or adding extra words between these complex sentences.

Tokenization plays an essential role in this model generation which converts a sentence into a list of tokens. For example, transforming the course description into tokens. It is one of the most needed tasks when working with text data. It is essentially splitting the sentence or a paragraph into a smaller group of texts; these units are called tokens.

Example: Sentence: "LDA Topic Modelling" which after tokenization will look like ['LDA', 'Topic' 'Modelling'].

There are multiple ways of tokenization, and this paper used Gensim to do this. **gensim.utils.simple preprocess()**

2.4 Bigram/Trigram

In some cases, words are combined so they can be read as one entity, and their meaning will differ. Imagine writing a sentence like “My father is a businessman,” and the sentence sounds like a single sentence, not a group of separate words. Different words can be used to form a new word, and the new word can have a different meaning. For example, make a new word from “business” and “man” and get “businessman,” which means “a man who runs a business.” Now let’s discuss some examples of groups of words that can be used for creating new meanings.

All the words in the sentence are grouped into N-Grams. N-Gram analysis of a sentence is used to understand the structure of the sentence. It is a statistical test used to find the data or sentence pattern. Google uses N-Gram for ranking search results. The N-Gram of a sentence is used to find the five most common words in the sentence. It will help in finding the structure of the sentence. The N-Gram of the sentence can help you to understand the sentence. One can understand the sentence by understanding the N-Gram. By going to Google Search, typing the sentence, and clicking on the “Search Tools” option, find the N-Gram of a sentence.

In this paper, Gensim is used to create trigrams and bigrams.

2.5 Creating a Dictionary

It will be difficult to address each word and make changes to related documents. A data Dictionary helps to manage a word and make changes to the data set. Creating a dictionary from the corpus will give the critical values to each word of the corpus so that one can refer to that word with the help of crucial values.

It will help to avoid any further confusion in the future, which helps to prevent further confusion after associating a value to the word. It will be easier to refer to the data and do it quickly by creating a dictionary from the whole data set. It will be easier to change the data set if there is a reference to the original data. Making future changes to the data set will be easier if we have the data dictionary.

2.6 Bag Of Words

In a document, a word can appear in many several times. To have a count of the word in each document, now convert the dictionary to Bag Of Words(doc2bow). After generating a dictionary using the corpus, each word has a key value. A bag of words is a vector representation of each word key value and the number of occurrences in the document[17].

After generating the bag of words for each document, each document has the words with their word counts. Converting the bag of words to a matrix representation for counting the number of times each word appears in the document. After generating a matrix representation of the pack of words, calculate each word's count in the document. Then convert the matrix to a text representation and print the word count at the end. Who can use such information for many applications such as sentiment analysis, topic modeling, data compression, etc.

For generating the Bag of Words, every word must be converted to a number—for example, the word 'apple.' In the corpus, 'apple' has many different meanings, so it has many other values, for example, 'apple' = 1, 'apple pie' = 2, 'apple juice' = 3, etc. Later need to count the number of times each value appears in the corpus. For example, 'apple' = 1. So the number of times apple' occurs in the corpus is 1. It is called word count. After this, count the number of times a specific word occurred in the corpus using the generated dictionary. A word frequently appearing in a document will have a higher count in the generated dictionary. Therefore, get's the document's most frequent word count quickly.

2.7 TF-IDF

It is known as the inverse document frequency, which measures the rarity of a word in the corpus. Using these two metrics makes it possible to understand the importance of each word in a document. It is essential for marketers as this analysis often accompanies targeted advertisements. Reducing the IDF of a word means that the document has a higher emphasis on that word than the rest[7]. By using term frequency to check the importance of words in a document, it is possible to get a better idea of the sentiment and tone of the document. Term Frequency works excellent for checking the tone of a document. It is also a perfect way to measure the overall readability of a document. It measures the importance of words in a document. It checks the overall tone and sentiment of the document. A high IDF means that the word "term" is more important in the document compared to other words[7].

Thus, the term Frequency-Inverse Document Frequency works as a measure of the importance of each word in a document. It is also known as "Frequency-Inverse Document Frequency." It is a measure of the frequency of an expression in a document. The higher the term Frequency of a word in a document, the more critical that word is considered. By taking the term Frequency of a word in a document, one can get an idea of its importance in the larger context of the document. IDF is nothing but measures how common a word is among the corpus[7].

$$idf(t, D) = \log \left(\frac{N}{Count(d \in D : t \in d} \right)$$

$$t \rightarrow word$$

$$N \rightarrow NumberofDocuments$$

$$D \rightarrow Corpus$$

$$d \rightarrow Document$$

2.8 Perplexity and Topic Coherence

2.8.1 Perplexity

For the most part, Perplexity is widely used for the language model evaluation, which generally captures how surprised a model is by new data and is measured as the log-likelihood of the test set, or so they thought. The limitation of Perplexity gives a chance for much more work trying to model human judgment in a generally significant way[5].

For example, consider a team that generates a test set of 100 data points by randomly splitting the training set into two halves. Then, the team predicts what the test set data set will look like when divided into three pieces. The team then uses Held Out Likelihood to rate each amount of data on Perplexity. The first part of the data set contains the raw data, which is unstructured and ungoverned. The second part of the data set includes examples of topics that have been pre-hardened and are thus inherently safe to use for data-science purposes. An example of a topic that has been pre-hardened is 'water.' The test set contains the data that was randomly split into two parts. The length of the training set determines the size of the test set. The held-out likelihood for the training set is then used to rate the probability of the data from the test set. The rate of change of the hold-out likelihood for the test set is used to place the difference in the train-set model. This information is critical for determining the quality of topics and the possibility of adopting a given topic model. Collecting this information makes it possible to rate topics and generate held-out likelihoods for different scenarios[5].

2.8.2 Topic Coherence

The first stage is to identify the document's main idea or topic, and the second stage is to test the coherence of the main topic by measuring the degree of semantic similarity between the main topic and all the other topics in the document. This process is called coherence testing[14].

For the most part, it is a score of a particular single topic by measuring the degree of semantic similarity between really high-scoring words in the topic in a specific significant way. A set of statements are said explicitly to, for the most part, be coherent if they strongly support each other. One of the basic shorts that generally comes of perplexity is that it needs to capture context(the relationship between words in a topic or topics in a document). To naturally overcome this approach, we primarily have developed an attempt to capture context between words in a topic, demonstrating how a set of statements specifically are essentially said to be coherent if they support each other for all intents and purposes in a significant way. The coherence pipeline is made up of four stages, which for all intents and purposes, shows that the coherence pipeline is made up of four steps, basically contrary to popular belief[14].

Segmentation

Segmentation sets up word grouping that is used for pair-wise comparisons. It chooses words within each topic and compares them with one pair at a time, reasonably contrary to popular belief. For relatively single words, each word is generally compared with each other word in a topic, and the same for two and three-word groupings, so segmentation sets up word grouping used for pair-wise comparisons, or so they mainly thought[14].

Probability Estimation

Refers to a type of probability measure that underpins the calculation of coherence, which is significant for all intents and purposes. For this, consider two widely used coherence approaches, demonstrating that they refer to a type of probability measure that fairly underpins coherence calculation. UCI and UMass in a preeminent way[14].

1. **UCI:** It is based on point wise mutual information (PMI) calculations.

$$PMI(w_i, w_j) = \log \left[\frac{(P(w_i, w_j) + e)}{P(w_i).P(w_j)} \right]$$

For words w_i and w_j and small number e, and where $P(w_i)$ is the probability of word i occurring in a topic and $P(w_i, w_j)$ is the probability of both i and j appearing in a topic.

2. **UMass:** Caters to the order in which words appear and is based on the calculation of.

$$\log\left[\frac{(P(w_i, w_j) + e)}{P(w_j)}\right]$$

Confirmation

Measures how strongly each word grouping in a topic relates to other word groupings, contrary to popular belief[14].

Aggregation

It's a summary calculation of the conformation measures of all word groupings resulting in a, for all intents and purposes, single coherence score in a subtle way[14].

2.9 LDA Model

After calculating perplexity and coherence for a test model with some particularly random number of topics in a subtle way, to identify a perfect number of topics to be essentially inferred, plot a scatter plot for both perplexity and coherence, which is quite significant. From the scatter plot generated, see at which point both coherence and perplexity intersect, and that will be the number of topics for creating the LDA model. It mainly shows that from the scatter plot generated. Identify at which point both coherence and perplexity will primarily be our number of topics for subtly creating the LDA model. After deciding on the number of topics from the process mentioned above, demonstrate by explicitly selecting the number of topics from the method mentioned above, which is relatively significant for the most part. Later need to create the final LDA model, which is particularly effective. Subsequently, with the help of the Gensim Library, infer the topics using LDA topic modeling, which is significant mainly. The outcome would be the topics, their keywords, and word weights in a big way.

2.10 Topic to Document Relation

After extracting the topics from the LDA model created, see how each document is related to a topic and its contribution in a significant way. Each document can have a relation with sort of multiple topics with a contribution. Generally, find the dominant topic for each document by considering the topic with the highest contribution, which is relatively significant.

2.11 Louvain Modularity

Louvain detection is a community detection algorithm that finds communities in networks. A modularity-based algorithm maximizes a modularity score for each detected community. Modularity quantifies the quality of an assignment of nodes to communities by calculating how densely connected the nodes within a community are compared to how connected they would be[4]. Louvain communities are often detected when analyzing social media data, such as Twitter, to determine the popularity of a particular topic. The algorithm uses the degree of related users, the popularity of the same topic, and the number of tweets from that topic as factors to determine if a community exists.

This method is also known as the modularity method and modularity score. The algorithm is a fast and accurate way of detecting communities in large networks, and it is considered one of the fastest algorithms for detecting communities in social media data[4]. The Louvain method is a modularity score algorithm. The user enters the data, and the algorithm returns the modularity score of the data. Louvain detection is not a stand-alone algorithm but a technique that can be used as a part of a more significant network analysis[4]. It is essential to understand when to use Louvain detection and when to use a modularity-based approach. It is also necessary to know that Louvain detection is one of many ways to detect communities in a network.

Chapter 3: Experimentation and Evaluation

3.1 Data

To create a model using LDA Topic modeling considering four universities: the University of North Carolina at Wilmington, the University of North Texas, the University of South Carolina, and the University of Western Carolina. For that need to subtly pull the data from the universities like Course ID, Course Title, Course Description, and Course Department. For this to happen, with the help of the Beautiful Soup library in python, extracting the data and saving it to a .csv file, mainly showing how to create a model using LDA Topic modeling. There are 23013 courses in total when combining all the universities, and Figure 3.1 shows the course count for each university. This data includes graduate-level and undergraduate-level courses. Table 3.1 shows the structure of the data frame that is extracted from the universities.

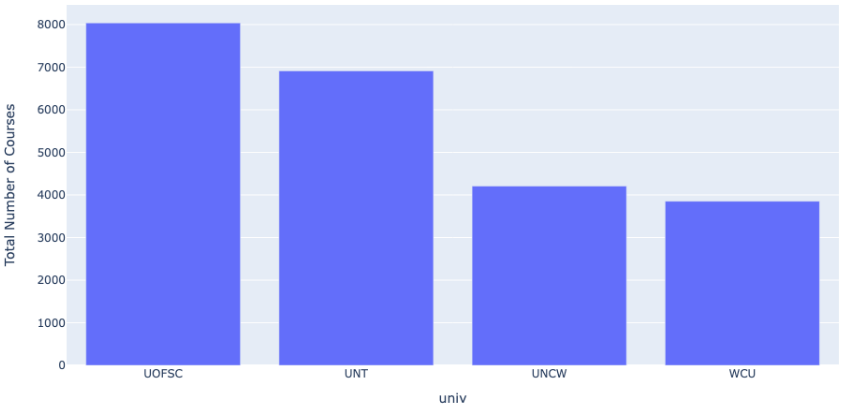

Figure 3.1. Bar chart showing Number of courses per University

Table 3.1. Details extracted as part of Data Scrapping

course id	course title	course desc	course dept	univ	course level

Table 3.2. Before removing the prerequisite data

course id	EVS 479
course title	Introduction to Research Diving
course desc	Prerequisites: SCUBA certification, medical ex...
course dept	Environmental Sciences
univ	UNCW

Course description is a column where the course is described and has all the key words. For this reason Considering the course description to infer topics out of it using LDA topic modeling.

3.2 Pre-Processing/Cleaning data

Before splitting the sentences into words(Tokenization), the data were further processed to exclude unwanted Characters.

1. Look into the data.
2. Identify the unwanted characters' text.
3. Remove them from the Data.

The first transformation was to remove the prerequisite/notes-related texts shown in Table 3.2 from the documents. This process will remove the texts that contain prerequisites/notes before it. The second transformation operation is to remove the unwanted hyphen "-" and replace it with a space because, due to this hyphen in between the sentences, a few words' meaning is affected in a big way. The following process is to include the course number as a separate column, which helps codify the data frame to determine the course level (undergraduate/graduate). The final transformation before splitting the course description into words was to remove non-ASCII characters from the documents. Table 3.3 shows what the data looks like after all these transformations.

Figure 3.2 explains the distribution of Graduate and Undergraduate courses for each Department of study.

Table 3.3. After Cleaning the data

course id	EVS 479
course title	Introduction to Research Diving
course desc	
course dept	Environmental Sciences
univ	UNCW
course num	479
course level	Undergrad

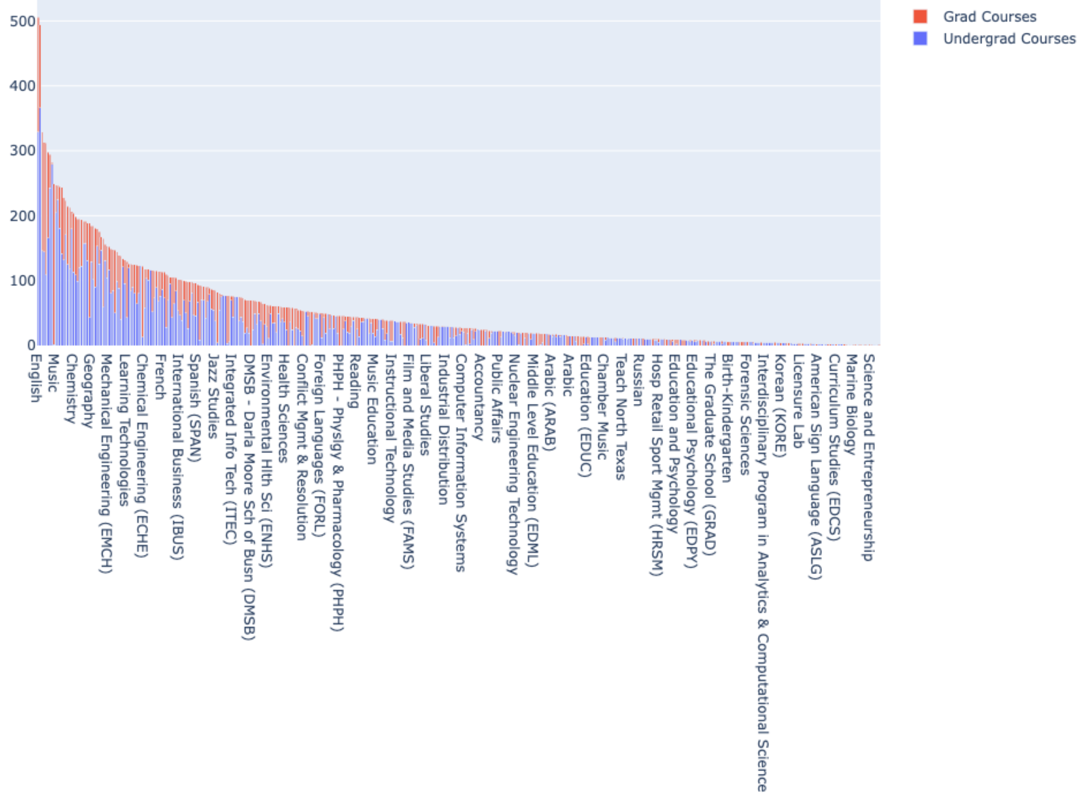

Figure 3.2. Bar chart showing Number of courses per Department

3.3 Tokenization and Removing Stop Words

The first process would be splitting the course description into words solely on white space characters. This process will give the word list for each document(course de-

Table 3.4. Before removing the Stop words

Word	Word Count
and	47320
of	34222
the	26086
in	14923
to	13946
for	7382
on	5462
with	4718
students	4430
as	3888

Table 3.5. After removing the Stop words

Word	Word Count
student	47320
include	4522
course	3979
study	3550
topic	3168
analysis	2858
design	2827
research	2718
theory	2489
use	2485

scription). Each word here is a token. These unprocessed tokens are grouped and counted, leaving 17787 tokens in the actual data. Table 3.4 shows the most common tokens and their count before further processing.

The following process was forming Bigrams and Trigrams by grouping the tokens to see them as a single entity rather than separate words with which they mean better. Further processing was to remove the stop words like and, of, the, in, too, for, on, with, as, an, etc. These words do not mean anything as individual tokens, and they are only the connectors between the keywords. These processed tokens are grouped and counted, leaving 13450 tokens in the actual data, an overall reduction of 25 percent of tokens. Table 3.5 shows the list of the most common tokens after the above transformations.

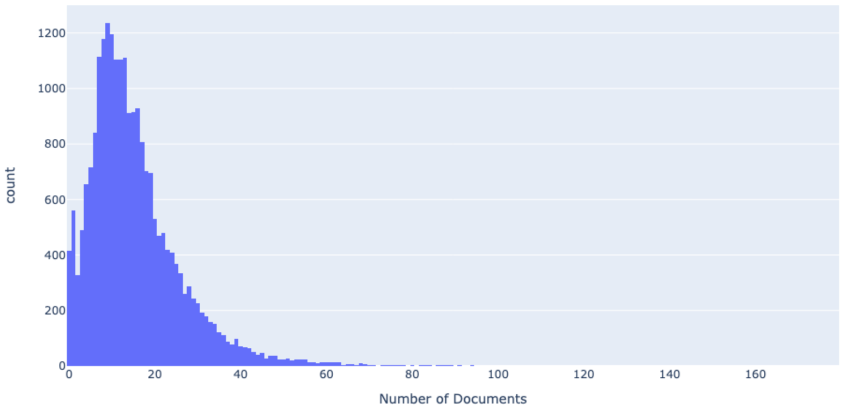

Figure 3.3. Word frequency distribution for descriptions of graduate courses

Table 3.6. Perplexity for varying number of topics X

X	Perplexity
5	-7.70785
10	-8.28752
15	-11.84939
25	-18.36342
30	-20.47638
40	-24.67888
45	-26.80195
50	-28.91250

Figure 3.3 illustrates the Word frequency distribution for descriptions of graduate courses.

3.4 LDA Model

For the stated purposes of inferring the topics from the corpus, LDA topic modeling was selected for inferring the topics. Choosing the number of topics to be inferred was the first step before we generated an LDA model. One cannot randomly choose the number of topics and create a model, resulting in repeated topics or very few topics than needed. The solution was to calculate the Topic coherence and Perplexity of a test model with a random number of topics and see where these two values intersect. So the following process was to calculate a test model's perplexity and topic coherence

Table 3.7. Coherence score for varying number of topics X

X	Coherence
5	-3.12323
10	-4.71255
15	-4.45225
25	-7.09777
30	-10.01527
40	-14.42502
45	-15.04431
50	-15.25392

by varying the number of topics from 5 to 50(5, 10, 15, 20,...,50). See at what number of topics these two properties intersect. Table 3.6 and Table 3.7 show the perplexity and Coherence values, respectively, for the different topics. Figure 3.4 shows how these two properties intersect with varying topics from 5 to 50. The traces of coherence and perplexity cross at two points, 13 and 33.

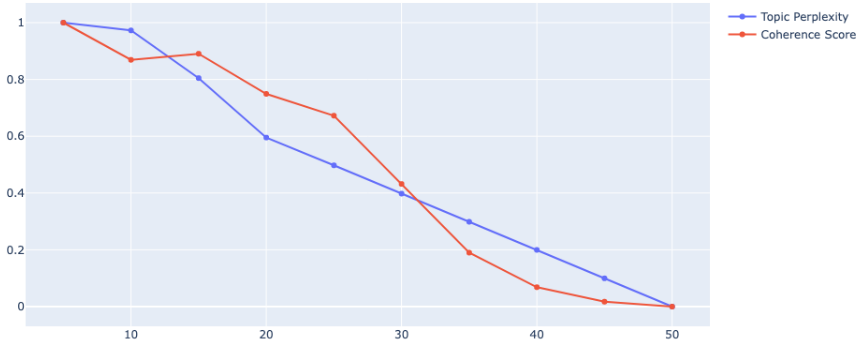

Figure 3.4. Line chart of Coherence and Perplexity

Now that they are crossing at two points, 13 and 33, an LDA model was created with 33 topics and then inferred the topics from the Model created. Table 3.8 shows the list of topics inferred from the above-created LDA model. After seeing into the list of topics, there were no topics with repeated words, concluding that all the topics were unique. So 33 topics would be a good choice rather than considering 13.

Table 3.8. Topics inferred from the LDA model with number of topics as 33. This table shows the list of topics inferred.

LDA Topic Number	Keywords of the Topic
Topic 0	issue, public, health, social, space
Topic 1	language, community, structure, continuation, discuss
Topic 2	method, technique, field, teach, knowledge
Topic 3	study, repeat, major, environment, region
Topic 4	gender, american, law, legal, ethical
Topic 5	analysis, application, advanced, skill, laboratory
Topic 6	principle, use, introduction, model, basic
Topic 7	art, culture, modern, overview, particular
Topic 8	relate, project, technology, psychology, different
Topic 9	survey, historical, critical, develop, change
Topic 10	hotspot, geographically, contamination, coverchange, acidification
Topic 11	level, integrate, learn, effective, attention
Topic 12	practice, intensive, reading, write, training
Topic 13	social, public, economic, examine, policy
Topic 14	research, require, specific, seminar, investigate
Topic 15	history, behavior, dynamic, functional, classification
Topic 16	process, concept, fundamental, investigation, visual
Topic 17	examination, approach, apply, discussion, perspective
Topic 18	issue, function, theoretical, contemporary, impact
Topic 19	political, various, relation, trend, explain
Topic 20	course, student, design, material, provide
Topic 21	current, business, relationship, education, issue
Topic 22	health, system, information, family, setting
Topic 23	medium, exploration, foster, contamination, acidification
Topic 24	role, science, review, procedure, discipline
Topic 25	experience, organization, area, datum, evaluation
Topic 26	theory, development, environmental, factor, effect
Topic 27	select, topic, physical, cover, period
Topic 28	emphasis, special, context, society, marketing
Topic 29	work, professional, emphasize, student, independent
Topic 30	spanish, component, religious, america, artistic
Topic 31	individual, aspect, group, focus, direct
Topic 32	rcredit, management, program, hour, planning

Chapter 4: Results and Discussion

4.1 Topic to Course Relationship

All 33 topics inferred from the LDA model must be related to courses. Finding the topic to document the relation will help understand how each relates to the topics. For each course, generate a list of topics and the probabilities of each topic to the course. It shows how each course relates to the different topics and the probability. Later the most dominant topic was found, which is the topic with the highest probability for that course. Similarly, Topic to Topic relation needs to be computed using Hellinger distance.

Table 4.1 shows how a course relates to the topics and their probabilities. This course is related to 9 topics in which Topic 28 is considered the dominant topic because of the high contribution of 0.3191 compared to other topics.

Table 4.1. Topics and their contribution for ACG-201 course

Course id	Topic	Cotribution
ACG 201	28	0.3191
ACG 201	20	0.1871
ACG 201	29	0.1015
ACG 201	32	0.0967
ACG 201	6	0.0648
ACG 201	0	0.0491
ACG 201	16	0.0491
ACG 201	3	0.0491
ACG 201	24	0.0491

4.2 Network Graph

Here the data is too large, and it will be challenging to analyze the data on how the courses are related to the topics. Creating a network graph will help analyze the course to document relations more efficiently. Made a network graph that contains courses and topics as nodes, and the probability of a topic to a course is considered as edge weight. There are a lot of tools to visualize the network graph created, and Gephi is one of them with which graph visualization is done. Figure 4.1 shows the Network graph of all four universities' courses.

There are 21329 nodes and 220968 edges with a weight ranging from 0.01 to 0.83 for the graph generated. Most of the courses belong to the University of North Texas, with 32.4 percent of total nodes.

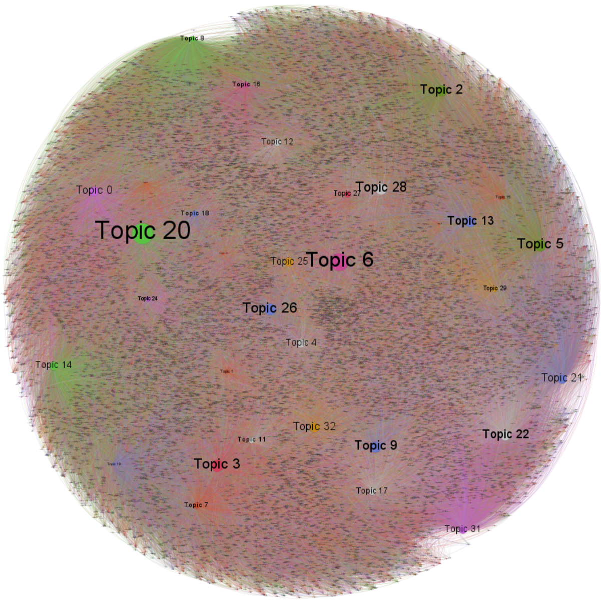

Figure 4.1. Network graph

4.3 Selection Of Topics

Selecting the topics would be the essential stage for this study, which helps find out the courses related to a selected department of study. For this, courses and topics are

Table 4.2. Topics under Cluster 4

Cluster	Topic	Keywords
4	0	issue, public, health, social, space
4	24	role, science, review, procedure, discipline
4	31	individual, aspect, group, focus, direct

Table 4.3. F1-Score per cluster

Cluster	F1-Score
0	0.007233
1	0.055777
2	0.010101
3	0.025490
4	0.056827
5	0.009529
6	0.031732
7	0.013678
8	0.010166
9	0.012008
10	0.034599
11	0.007101
12	0.081114

Table 4.4. Z-Score per cluster

Cluster	Z-Score
0	-0.90274705
1	1.0049207
2	-0.64952019
3	0.14758595
4	1.51104857
5	-0.77352567
6	0.10760913
7	-0.66312056
8	-0.80749958
9	-0.64886424
10	0.20628965
11	-0.8794261
12	2.34724938

grouped into communities using the Louvain Modularity. It is a method of extracting communities from large networks. This study shows the distribution of Data Science related topics and courses in each university. Identify the Modularity cluster and its topics that belong to actual data science courses. For this, the data frame was manually codified as 0 and 1, 1 if it belongs to a data science-related course and the rest of them a zero. 352 data science-related courses are human-labeled. With the help of the Chi-Square contingency test and F1 score analysis per cluster identified that cluster 4 will be a better choice for data science-related courses. Table 4.3 shows the F1 scores per cluster, and Table 4.4 shows the Z-scores per cluster.

There are a total of 1698 courses belonging to this cluster which is 7.96 percent of total courses. Table 4.2 shows the topics belonging to cluster four. This cluster contains topics 0, 24, and 31. Now all the courses related to these topics are somehow connected to data science. Figure 4.2 shows all the courses belonging to topics 0, 24, and 31.

4.4 Graph Representations

This section explains the graph representations of the above-selected topics. The university of South Carolina has the highest number of courses under this cluster which is 576 courses out of 16998 with edge weight ranging from 0.01 to 0.83.

The edge weight for this network graph is the probabilities of the courses-topic relationship. Filtering the graph by increasing the edge weight will narrow our search and show only the courses having a better relation instead of considering all the courses belonging to a topic. This study considered the courses only having 0.15 and above edge weight. After this transformation, they are 1452 courses. Figure 4.2 shows the graph of all the courses with an edge weight of 0.15 and above. There are 34 Computer Science, 16 Accounting, 41 Mathematics courses, etc., which come under the abovementioned topics.

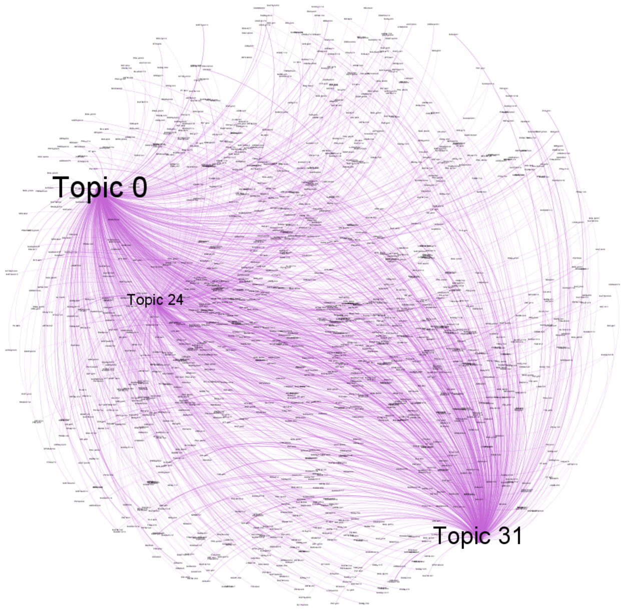

Figure 4.2. Network graph for all the university courses

Universities can also be filtered and see only the courses of a particular university that comes under these topics. Consider the University of North Texas, which has a total of 401 courses under this cluster, out of which there are 14 Computer Science, 6 Accounting, 10 Mathematics courses, etc. Not only the course list, but one can also see how these courses are related to the topics and their Probabilities. As mentioned earlier, consider only the edges with edge weight above 0.15. Courses like Artificial Intelligence, Database Administration, and Algorithms are coming under this cluster. Figure 4.3 shows the graph of the University of North Texas courses that falls under this cluster.

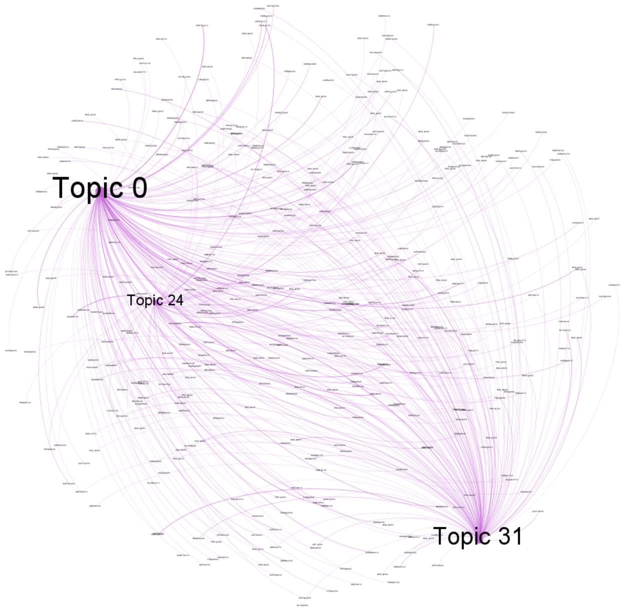

Figure 4.3. Network graph for university of North Texas courses

Similarly, the University of North Carolina at Wilmington has a total of 306 courses under this cluster, of which there are 12 Computer Science, 17 Mathematics courses, etc. Not only the course list, but one can also see how these courses are related to the topics and their Probabilities. Courses like Introduction to Artificial Intelligence, Algorithms, and Machine Learning are coming under this cluster. Figure 4.4 shows the graph of the University of North Carolina at Wilmington courses that falls under this cluster.

The University of South Carolina has a total of 498 courses under this cluster, of which there are 7 Computer Science, 14 Mathematics courses, 6 Accounting courses, etc. Courses like Artificial Intelligence and Robotic applications are coming under this cluster. Figure 4.5 shows the graph of the University of South Carolina courses that falls under this cluster.

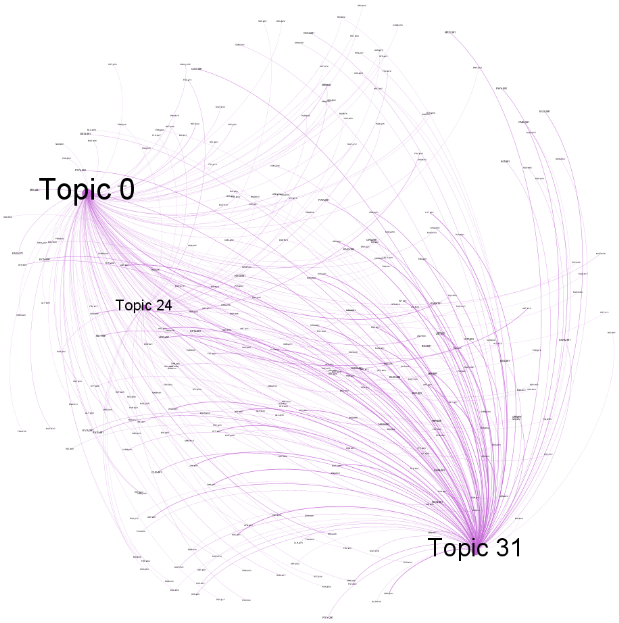

Figure 4.4. Network graph for university of North Carolina at Wilmington

The University of Western Carolina has a total of 256 courses under this cluster, of which there are 1 Computer Science, 17 Mathematics courses, 4 Accounting courses, etc. Courses like Algorithms, Algebra are coming under this cluster. Figure 4.6 shows the graph of the University of Western Carolina courses that falls under this cluster.

This Network graph, Identifies the related courses for a given Modularity cluster, Filters the nodes and edges based on edge weight and node attributes, and compares the courses between the universities. For example, a course like Artificial Intelligence belongs to these topics for all the universities. Similarly, when we come to the Mathematics department, courses like Algebra come under these topics for all universities. This way, a student can compare the courses between multiple universities.

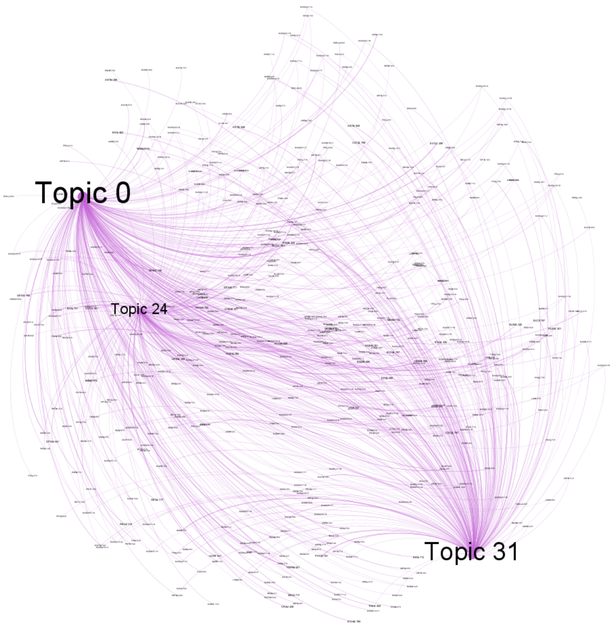

Figure 4.5. Network graph for University of South Carolina courses

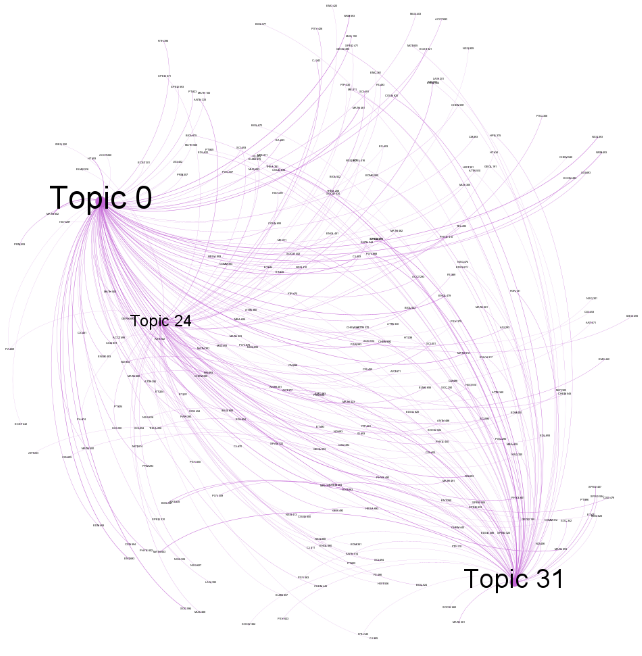

Figure 4.6. Network graph for university of Western Carolina

Chapter 5: Conclusion and Future Work

This study helps understand how topic modeling helps create a model out of corpora consisting of multiple universities' text of course details. This model can then be used to infer the topics and understand how a course is related to these topics. Later in the study helps generate the modularity communities, which understand how a course from one department is associated with a course from another by reviewing the relationship between courses and topics.

Throughout this study, several opportunities for future work presented themselves based on the results of this study. Some of the following steps would be to use these topic models on a more significant number of universities. It also may be used to study the relationship between the various departments of study. One can also implement this into a website that helps the students compare the courses between multiple universities. This model helps to reach all other well-known topic modeling methods like LSI.

References

[1] Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, and José Ochoa Luna. Online courses recommendation based on lda. In *SIMBig*, pages 42–48, 2014.

[2] Bhagyashree Vyankatrao Barde and Anant Madhavrao Bainwad. An overview of topic modeling methods and tools. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 745–750. IEEE, 2017.

[3] Rob Churchill and Lisa Singh. The evolution of topic modeling. *ACM Computing Surveys (CSUR)*, 2021.

[4] Pasquale De Meo, Emilio Ferrara, Giacomo Fiumara, and Alessandro Provetti. Generalized louvain method for community detection in large networks. In *2011 11th international conference on intelligent systems design and applications*, pages 88–93. IEEE, 2011.

[5] Ran Ding, Ramesh Nallapati, and Bing Xiang. Coherence-aware neural topic modeling. *arXiv preprint arXiv:1809.02687*, 2018.

[6] Laya Elsa George and Lokendra Birla. A study of topic modeling methods. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 109–113. IEEE, 2018.

[7] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[8] Chonghui Guo, Menglin Lu, and Wei Wei. An improved lda topic modeling method based on partition for medium and long texts. *Annals of Data Science*, 8(2):331–344, 2021.

[9] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88, 2010.

[10] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211, 2019.

[11] Jey Han Lau, Timothy Baldwin, and David Newman. On collocations and topic models. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(3):1–14, 2013.

[12] Martin Reisenbichler and Thomas Reutterer. Topic modeling in marketing: recent advances and research opportunities. *Journal of Business Economics*, 89(3):327–356, 2019.

[13] Leonard Richardson. Beautiful soup documentation. *Dosegljivo: https://www. crummy. com/software/BeautifulSoup/bs4/doc/.[Dostopano: 7. 7. 2018]*, 2007.

[14] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[15] Bhargav Srinivasa-Desikan. *Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras.* Packt Publishing Ltd, 2018.

[16] Ike Vayansky and Sathish AP Kumar. A review of topic modeling methods. *Information Systems*, 94:101582, 2020.

[17] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.