Using Natural Language Processing to Facilitate Common Student Misconception Analysis

by

Azreen Zaman

B.S. Aerospace Engineering and Electrical Engineering and Computer Science, Massachusetts Institute of Technology (2022)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Azreen Zaman
Department of Electrical Engineering and Computer Science
May 12, 2023

Certified by: Mohamed Abdelhafez
Lecturer, Department of Physics
Thesis Supervisor

Accepted by: Katrina LaCurts
Department of Electrical Engineering and Computer Science
Chair, Master of Engineering Thesis Committee

Using Natural Language Processing to Facilitate Common Student Misconception Analysis

by

Azreen Zaman

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

There is a large variation in the educational background and purpose of incoming university students. To improve the overall learning experience of these students, we can utilize natural language processing such as topic modeling and sentiment analysis to facilitate common student misconception analysis. This project aims to develop an algorithm via natural language processing that extracts specific topics and common errors that students struggle with in class from online feedback semi-automatically to allow instructors to adjust lesson plans and place emphasis on topics of concern. Using these tools, we can conduct study on the effect on student grades when instructors take into account the information extracted by the model in their lesson plans. This project is aimed at MIT freshmen taking two semesters of physics.

Thesis Supervisor: Mohamed Abdelhafez
Title: Lecturer, Department of Physics

Contents

Chapter 1

Introduction

The current educational system is an example of a landscape that is continuously improved by a vast amount of data that is produced every day in many different formats and most frequently conceals vital and useful information. One of the greatest benefits that sentiment analysis [4] and topic modeling approaches [7] may offer is the ability to locate and extract the hidden "pearls" from the sea of educational data. Due to the current pandemic outbreak, which forced many colleges and institutions to switch from on-campus physical classes to online instruction using eLearning platforms and techniques like massive open online courses (MOOCs), the importance has expanded significantly.

At MIT, the intro physics 1 course 8.01 has adopted various online learning tools during the pandemic to help students stay on track with the curriculum during this sudden shift into remote learning. For example, 8.01 has introduced learning sequences before the lecture to introduce topics and give a brief summary of the lecture to the students. The students are then expected to answer problems based on these materials. Students have found it difficult to correctly answer all the problems when they are learning independently during these sequences. The 8.01 teaching team then piloted a “Second Chances” Program that allows the student to receive the points to a problem they incorrectly answered or exhausted all attempts in answering. The student first needs to describe what specifically caused them to answer the problem incorrectly and then describe what the right approach was. This data can help instructors understand specific topics students are struggling with and adjust their lecture to put emphasis on these topics and reinforce student learning. We can utilize latent Dirichlet allocation (LDA) topic model [6] to identify prominent underlying misconceptions in the large collection of student responses. The impact of this model can directly be studied by examining grade averages in the class from before and after the algorithm was used. Additionally, we can compare certain sections' grade averages to one another by deploying the algorithm in some sections and not others.

Chapter 2

Related Work

2.1 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a type of topic modeling algorithm that is used to uncover the underlying topics in a collection of documents. It does this by identifying the words that are most associated with each topic, and then using those words to determine the main themes in the documents. The paper "Analyzing Large Collections of Open-Ended Feedback From MOOC Learners Using LDA Topic Modeling and Qualitative Analysis" discusses the use of natural language processing (NLP) techniques to analyze open-ended feedback from massive open online course (MOOC) learners. The study collected written feedback from MOOC learners on their learning experience and used NLP techniques, specifically topic modeling, to identify the main themes discussed in the feedback. The researchers also conducted a qualitative analysis of the feedback, which involved manually coding and categorizing the comments. The results of the topic modeling showed that the main themes identified in the feedback included the quality of the course content, the usefulness of the course, and the interaction with the instructor and other learners. The results of the qualitative analysis supported the findings of the topic modeling and provided additional insights into the learners' experiences. The study suggests that NLP techniques can be useful for educators in analyzing and using student feedback to improve MOOCs, and highlights the importance of combining NLP techniques with qualitative analysis to provide additional depth and context to the findings. [6]

2.2 Natural Language Processing in Education

The paper "Natural Language Processing and its Use in Education" discusses the use of natural language processing (NLP) techniques in the field of education. NLP involves the use of computational methods to process and analyze natural language data, such as text or speech. The paper explains that NLP can be used in education to analyze and understand student feedback, to automatically grade written assignments, and to assist in language learning. The paper also discusses the challenges and limitations of using NLP in education, including the need for high-quality training data and the potential for biases in the algorithms. The paper concludes by stating that NLP has the potential to significantly impact education, but that it is important to carefully consider the ethical and technical issues involved in its use. [1]

2.3 Sentiment Analysis

Sentiment analysis is the process of identifying and extracting subjective information from text data. It is often used to determine the overall sentiment or emotion of a piece of text, such as whether it is positive, negative, or neutral. The paper "Sentiment Analysis of Students' Feedback with NLP and Deep Learning: A Systematic Mapping Study" examines the use of natural language processing (NLP) and deep learning techniques for sentiment analysis of student feedback. The study conducted a systematic mapping, which involves searching for and reviewing a large number of studies in a systematic and transparent manner. The search identified a total of 26 studies that used NLP and deep learning techniques for sentiment analysis of student feedback, which were published between 2013 and 2019 and found in a variety of databases and conference proceedings. The majority of the studies used NLP techniques, such as sentiment analysis and topic modeling, to analyze student feedback on teaching performance. A smaller number of studies used deep learning techniques, such as convolutional neural networks, for sentiment analysis. The results of the studies showed that NLP and deep learning techniques can effectively identify the sentiment and themes in student feedback. [5] The study highlights the potential of using NLP and deep learning techniques for sentiment analysis of student feedback and suggests that further research is needed to explore the use of these techniques in different contexts and languages. The study also suggests that there is a need for more research on the use of NLP and deep learning techniques to analyze feedback from other stakeholders, such as employers or colleagues. [4]

Chapter 3

Proposed Work

For my thesis work, I aim to create an algorithm to extract common misconceptions from students whilst they complete their pre-lecture learning sequences using topic modeling and sentiment analysis. What we are interested in is finding specific topics from student's open ended responses and the topics' relative distribution among the responses. I would like to implement this algorithm in the spring semester class of physics such that the specific misconceptions can be utilized by professors to reinforce those ideas during their lecture. To analyze if this reinforcement is beneficial to students I would then conduct a study to see if average test scores increase from the previous year and test if some sections score better than others when this program is integrated in some sections and not others. Finally, I would like to adjust the algorithm such that it is effective for any class, not just physics.

3.1 Methods

3.1.1 Data Collection

We collect the data by linking a “second chance” form at the end of each learning sequence problem in which students can describe why they incorrectly answered the problem and what the right approach was. This information is then converted into a csv file that includes the following columns: Problem, What Went Wrong, What is Right Approach, as well as the Request Time. The Request time is the time a student submits a second chance form.

3.1.2 Data Processing

Now that we have the data the next step is to preprocess it. We can do this by deleting any irrelevant or noisy data, such as special characters or stop words. We will also need to stem and lemmatize the data to reduce dimensionality. Next, we will vectorize the data by converting the text data into numerical form. We can do this by using techniques such as term frequency-inverse document frequency (TF-IDF) or count vectorization. The Request Time column in the data can be used to convert to DateTime so we can index a portion of data specific to an assignment. For example, during analysis we can index 7 days worth of data to see the problems and responses from the learning sequence that was assigned that week. This will help specify topics students are struggling with in a given week.[3]

3.1.3 Model Testing

The first thing I will do with the cleaned data is generate a word cloud so that I have a clear visual of what words are reappearing in the responses. This will give us a clearer picture into what topics we want the model to pick up on. Based on this I would like to develop and test multiple models and algorithms for topic modeling on short open ended responses. Some algorithms I have been researching into is Latent Feature LDA [7] which is an extension of LDA that tries to find the latent topic structure that could have generated the observed documents. The difference is in the way words are generated from topics. I then want to perform sentiment analysis on the data to determine what opinion students hold for the topics that were discussed in their responses. At first glance, one may think that all the topics will be "misconceptions" as the prompt is asking the student to describe what they did wrong and why. However, there is potential for students to describe what they did understand well and how that helped them reach closer to the answer. Sentiment

Analysis will help figuring out the polarity of the topics and what can be labeled as misconceptions and what can be labeled as things students more so easily understand. Finally, I would like to test the accuracy of these algorithms before deploying them. One way to test the accuracy of a topic modeling algorithm is to manually evaluate the results. This involves examining the topics that the algorithm has identified and determining whether they are meaningful and coherent. Human Judgement can be used to assess the quality of the topics and determine whether they accurately represent the underlying structure of the data. There are also several evaluation metrics that can be used to quantitatively assess the accuracy of a topic modeling algorithm. Some common metrics include perplexity, coherence, and topic overlap. Perplexity measures how well the model is able to predict the words in a test set based on the learned topics, while coherence measures the degree to which the words in a topic are semantically related. Topic overlap measures the degree to which the topics identified by the model overlap with each other.

3.1.4 Expanded Scope

An extension of this project can be adjusting the algorithm such that it can be fed any list of responses from an arbitrary class and point out misconceptions from that data for instructors to utilize.

Chapter 4

Developing the Algorithm

For the bulk of my research, I focused on two algorithms: K-Means and Latent Dirichlet Allocation (LDA). I will describe each algorithm as follows and how I implemented them to extract common misconceptions from student learning.

4.1 K-Means

4.1.1 Description

K-means clustering is an increasingly popular unsupervised machine learning algorithm that can be effectively utilized in the realm of topic modeling. Topic modeling is an innovative technique that aims to unveil concealed themes within a compilation of documents. While methods such as Latent Dirichlet Allocation (LDA) are frequently employed for this purpose, k-means clustering introduces a distinctive approach that can yield valuable perspectives.

K-means clustering, at its core, groups data points into 'k' distinct clusters based on their likeness. Each cluster is characterized by a centroid, which is calculated as the average of all the data points assigned to that specific cluster. This adaptable process can be modified for topic modeling by considering each document as a data point and clustering them together based on their semantic resemblance.

To implement k-means clustering in topic modeling, several essential steps can be followed:

1. Initial Data Preprocessing: The textual data, typically comprising a collection of documents, necessitates pre-processing. This entails procedures such as tokenization, eliminating common words, stemming, and transforming the text into a numerical representation, such as TF-IDF or word embeddings, to enable compatibility with the clustering algorithm.

2. Determining the Ideal Number of Clusters (k): Establishing the appropriate number of clusters presents a notable challenge in k-means clustering for topic modeling. Various techniques, including the elbow method or silhouette analysis, can be employed to evaluate the quality of clustering based on the internal cohesion and external separation of the clusters.

3. Applying k-means Clustering: Once the number of clusters is determined, the k-means algorithm can be employed on the pre-processed data. This iterative algorithm assigns each document to the nearest centroid and subsequently updates the centroids based on the new assignments. This process continues until the algorithm reaches convergence.

4. Extracting Insights from Clusters: Following the clustering phase, every document will be assigned to one of the 'k' clusters. These clusters can be interpreted as representative topics or themes present within the document collection. Examining the centroids of the clusters allows for a better understanding of the characteristic features or keywords associated with each topic. Furthermore, analyzing the documents within each cluster facilitates gaining insights into the content linked with each topic.

5. Evaluation and Refinement: The quality of the clustering results can be evaluated using evaluation metrics such as silhouette score or coherence measures. In cases where the results are not satisfactory, further refinement can be performed by adjusting preprocessing techniques, modifying the number of clusters, or exploring alternative clustering algorithms.

One of the advantages of utilizing k-means clustering in topic modeling is its simplicity and efficiency. It demonstrates scalability when processing large datasets and effectively handles high-dimensional feature spaces. Nonetheless, it is essential to acknowledge its limitations. K-means clustering assumes spherical clusters and may face challenges with complex or overlapping topics. Additionally, it necessitates the subjective specification of the number of clusters.

In conclusion, k-means clustering serves as a valuable technique for topic modeling by facilitating the discovery of latent topics within a document collection through the grouping of similar documents. Interpreting the resulting clusters offers valuable insights into the underlying themes present in the data. Although k-means clustering has inherent limitations, it provides a straightforward and efficient approach to topic modeling that can significantly enhance the existing toolkit of topic modeling techniques.

4.1.2 Implementation

Package Imports

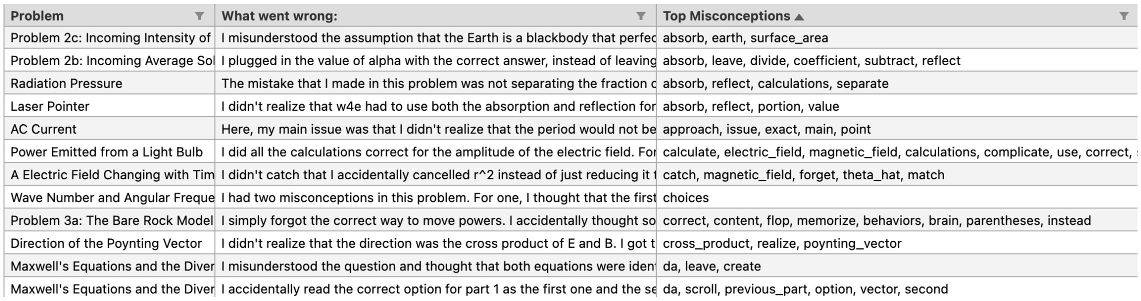
The algorithm starts by importing the necessary packages: pandas, nltk, TfidfVectorizer from sklearn.feature_extraction.text, KMeans from sklearn.cluster, and cosine_similarity from sklearn.metrics.pairwise. Pandas is a data manipulation library that provides a flexible and powerful toolset for working with structured data. It is used to create a data frame from the input CSV file. NLTK (Natural Language Toolkit) is a leading platform for building Python programs to work with human language data. It is used for text preprocessing, which includes tokenization, removing stopwords and punctuation, and lemmatization. The TfidfVectorizer is used to convert the preprocessed text data into a document-term matrix. It also removes stop words and applies other pre-processing steps such as tokenization and lemmatization. KMeans is a clustering algorithm that partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is used to cluster the preprocessed text data into k clusters, where k is the number of unique problems in the input data. Cosine_similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. It is used to find the response from "What went wrong" column that is most similar to the top words.

Data Cleaning

First, we read in the data from the csv file generated from the website that handles the second chances and convert it to a pandas DataFrame. Then we clean up the data such that we drop any irrelevant columns. We then convert the "Request Time" column into a datetime object such that we can index specific dates we want to analyze. We are now left with a pandas DataFrame with columns "Problem", "What went wrong","What is the right approach", and the index column which is the datetime. pr

Data Preprocessing

The algorithm then sets up constants, including the number of clusters which is equal to the number of unique problems in the input data. The algorithm defines a preprocess function that converts the input text to lowercase, tokenizes it, removes stop words and punctuation, lemmatizes the remaining words, and returns the preprocessed text as a string. The input data is then grouped by problem, and the "What went wrong" and "What is the right approach" columns are concatenated into a single column. The preprocess function is applied to this column, and a document-term matrix is created using TfidfVectorizer. We then split the sentences into a list of words and feed it into a bigram model. In the context of natural language processing, a bigram refers to a sequence of two consecutive words in a text. It is a type of n-gram, where 'n' represents the number of consecutive words considered. In a bigram, the order of the words matters, and they are typically used to capture some of the contextual information present in the text.



Problem	What went wrong:	Top Misconceptions ▲
Problem 2c: Incoming Intensity of	I misunderstood the assumption that the Earth is a blackbody that perfec	absorb, earth, surface_area
Problem 2b: Incoming Average Sol	I plugged in the value of alpha with the correct answer, instead of leaving	absorb, leave, divide, coefficient, subtract, reflect
Radiation Pressure	The mistake that I made in this problem was not separating the fraction c	absorb, reflect, calculations, separate
Laser Pointer	I didn't realize that w4e had to use both the absorption and reflection for	absorb, reflect, portion, value
AC Current	Here, my main issue was that I didn't realize that the period would not be	approach, issue, exact, main, point
Power Emitted from a Light Bulb	I did all the calculations correct for the amplitude of the electric field. For	calculate, electric_field, magnetic_field, calculations, complicate, use, correct,
A Electric Field Changing with Tim	I didn't catch that I accidentally cancelled r^2 instead of just reducing it t	catch, magnetic_field, forget, theta_hat, match
Wave Number and Angular Freque	I had two misconceptions in this problem. For one, I thought that the first	choices
Problem 3a: The Bare Rock Model	I simply forgot the correct way to move powers. I accidentally thought so	correct, content, flop, memorize, behaviors, brain, parentheses, instead
Direction of the Poynting Vector	I didn't realize that the direction was the cross product of E and B. I got t	cross_product, realize, poynting_vector
Maxwell's Equations and the Diver	I misunderstood the question and thought that both equations were iden	da, leave, create
Maxwell's Equations and the Diver	I accidentally read the correct option for part 1 as the first one and the se	da, scroll, previous_part, option, vector, second

Figure 4-1: sample of kmeans_results.csv for second chance student response data

Clustering

The KMeans clustering algorithm is then applied to the document-term matrix to create k clusters. The top words for each cluster are extracted by finding the most important words in each cluster using the argsort() method. The algorithm then filters out the correct approach words from the top words for each cluster and stores them in a list.

Results

For each cluster, the algorithm finds the response from the "What went wrong" column that is most similar to the top words by calculating the cosine similarity between each document in the cluster and the preprocessed text data. The top 5 most similar documents are retrieved, and the response with the highest cosine similarity to the original document is selected as the best response. Finally, the algorithm stores the problem, response, and top misconceptions for each cluster in a pandas data frame, which is saved as a CSV file. Overall, this algorithm uses a combination of text preprocessing, clustering, and similarity measures to extract common misconceptions from a set of problems and associated student responses. It can be easily adapted to handle different input data and optimize the clustering and similarity measures for different use cases. Figure 7-1 showcases a small sample of the top words for each problem and the associated misconception from the list of responses.

4.1.3 Evaluation

Silhouette Score

The Silhouette Score is a widely employed metric utilized for assessing the quality of clustering outcomes. It quantifies the similarity between data points within a cluster relative to data points in other clusters. The Silhouette Score is a bounded value ranging from -1 to 1, where higher scores indicate better clustering performance.

To compute the Silhouette Score, the following steps are undertaken:

1. For each data point, two values, denoted as $a(i)$ and $b(i)$, are calculated.
    - $a(i)$ represents the average dissimilarity between the data point indexed as "i" and all other points within the same cluster.
    - $b(i)$ represents the average dissimilarity between the data point indexed as "i" and all points in the nearest neighboring cluster, which refers to the cluster that exhibits the highest dissimilarity with the data point.
2. The Silhouette Coefficient for each data point indexed as "i" is computed employing the following formula:

$$silhouette(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{4.1}$$

3. The average Silhouette Coefficient across all data points is calculated to obtain the Silhouette Score, which serves as an evaluation measure for the clustering algorithm.

The Silhouette Score can be interpreted as follows:

- When the score approximates 1, it indicates well-clustered data points that are distant from neighboring clusters.
- A score close to 0 suggests the presence of overlapping or poorly separated clusters.

- If the score approaches -1, it signifies misassignment of data points to incorrect clusters.

[8]

Within-Cluster Sum of Squares (WCSS)

The Within-Cluster Sum of Squares (WCSS) is a widely used metric for assessing the compactness or tightness of clusters in clustering algorithms. It quantifies the sum of squared distances between each data point and the centroid of its assigned cluster, providing a measure of the clustering quality.

To compute the WCSS, the following steps are followed:

1. For each cluster, the sum of squared distances between each data point and the centroid of the cluster is calculated. Let n denote the number of data points in the cluster, $\mathbf{x}_i$ represent the coordinates of the i-th data point, and $\mathbf{c}$ denote the centroid of the cluster.

2. The squared Euclidean distance between each data point $\mathbf{x}_i$ and the centroid $\mathbf{c}$ is computed using the formula:

$$dist_i = \|\mathbf{x}_i - \mathbf{c}\|^2 \tag{4.2}$$

3. The squared distances are summed across all data points in the cluster to obtain the WCSS value for that particular cluster.

4. The WCSS values for all clusters are summed to obtain the total WCSS value for the clustering algorithm.

Mathematically, the calculation of the WCSS can be expressed as:

$$WCSS = \sum_{j=1}^{k} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_j\|^2 \tag{4.3}$$

where k represents the number of clusters, and n represents the total number of data points.

Table 4.1: Silhouette Score and WCSS value based on number of problems and number of clusters

Number of Problems	**Number of Clusters**	**Silhouette Score**	**WCSS value**
20	10	0.101	6.405
20	5	0.0705	11.750
40	20	0.0387	12.243
40	10	0.0267	24.243

The WCSS serves as an indicator of the compactness of clusters. A lower WCSS value indicates that the data points within each cluster are closer to their respective centroid, implying more well-defined and compact clusters. [2]

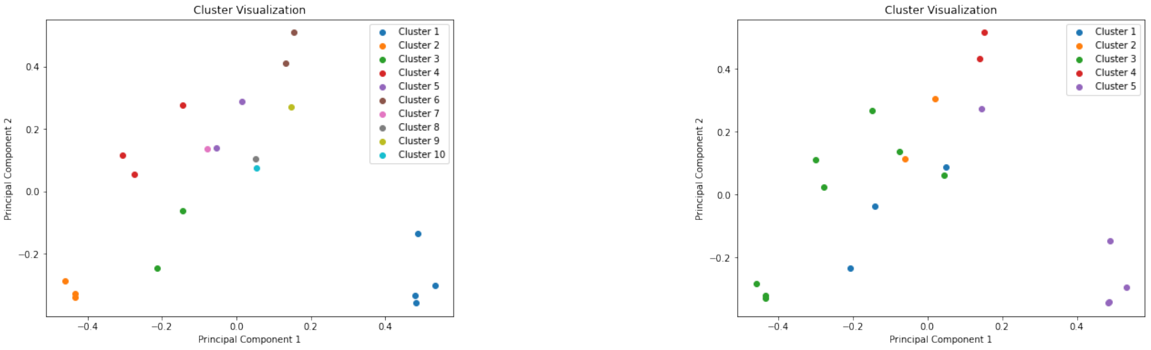
Results

We use these metric to determine if the clusters that were chosen based on the problems closely relate to eachother. The corresponding scores are shown in Table: 6.1 based on the number of problem responses we are analyzing and how many clusters we obtain from them. As we increase the number of problems the clusters become less defined and represents an overlapping of clusters. This is also seen when we decrease the cluster number as well. This suggests that minimizing the data to only a week worth of responses can give a better description of what the common misconceptions are for any given cluster.

4.1.4 Visualization

Principal Component Analysis (PCA)

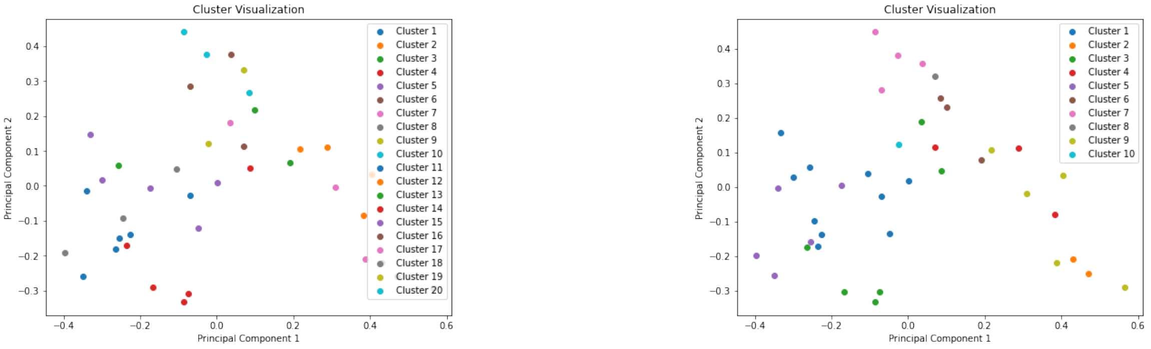
One common approach to visualize the clusters created by the K-means algorithm is Principal Component Analysis (PCA) which reduces the high-dimensional feature space to 2 or 3 dimensions, and then plots the data points with different colors representing different clusters. Each data point represents a problem response, and the colors indicate the assigned cluster. By visualizing the clusters, we can gain insights into the distribution and separability of the data points in the reduced feature



(a) 20 problems and 10 clusters.

(b) 20 problems and 5 clusters.

Figure 4-2: Cluster Visual for 20 Problem Responses.



(a) 40 problems and 20 clusters.

(b) 40 problems and 10 clusters.

Figure 4-3: Cluster Visual for 40 Problem Responses.

space. The visualisations can be seen in Figures 7-2 and 7-3

4.2 Latent Dirichlet Allocation (LDA)

4.2.1 Description

The Latent Dirichlet Allocation (LDA) model is a widely adopted method for topic modeling, aiming to uncover concealed thematic structures within a set of documents. LDA employs a probabilistic framework that facilitates the identification and interpretation of topics based on word distribution across documents.

Utilizing LDA for topic modeling involves the following steps:

1. Data Preprocessing: The textual data, comprising documents, undergoes pre-

processing to eliminate noise and irrelevant information. This typically encompasses tokenization, stop word removal, stemming, and possibly lemmatization. The resultant preprocessed documents serve as input for the LDA model.

2. Constructing the LDA Model: The LDA model assumes that each document is a blend of multiple topics, with each topic represented by a distribution of words. The model estimates these latent distributions through iterative word-to-topic assignments, adjusting probabilities based on statistical inference. The number of topics, a hyperparameter, can be predetermined or determined using methods like cross-validation or coherence measures.

3. Topic Inference: After training the LDA model, it can infer the underlying topics for new documents. This involves analyzing word distributions in the document and estimating the probability of each topic's presence. Based on these probabilities, the document can be assigned the most probable topics.

4. Topic Interpretation: The resulting topics from the LDA model can be interpreted by examining the most likely words associated with each topic. These words shed light on the main themes represented by the topics. Additionally, examining topic proportions within each document allows for understanding the document's content in terms of the identified topics.

5. Evaluation and Refinement: The LDA model's output can be evaluated using metrics such as coherence scores, which assess the semantic consistency of generated topics. If the results are unsatisfactory, the model can be refined by adjusting hyperparameters, modifying preprocessing steps, or incorporating additional text features.

A notable strength of the LDA model is its capability to uncover latent topics within a document collection. By considering statistical dependencies between words and documents, LDA identifies underlying thematic structures that may not be immediately apparent. Furthermore, LDA allows for soft assignment of topics, acknowledging that documents can relate to multiple topics with varying degrees of importance.

LDA proves particularly valuable for exploratory analysis and information retrieval tasks. It aids in tasks such as document clustering, content recommendation, text summarization, and other applications where comprehending the main themes within a large corpus is critical.

However, it is worth noting that the LDA model possesses certain limitations. It assumes documents are mixtures of topics, disregarding hierarchical relationships between topics. Additionally, it treats words as independent entities, overlooking their contextual dependencies within documents. These limitations can be addressed by incorporating advanced models like Hierarchical Dirichlet Processes (HDP) [10] or integrating contextual information through techniques like LDA2Vec.

To summarize, the LDA model serves as a powerful tool for topic modeling, offering a probabilistic framework to unearth latent thematic structures within a document collection. Through estimating topic-word and document-topic distributions, LDA facilitates topic interpretation and inference, benefiting tasks such as content analysis, recommendation systems, and exploratory analysis.

4.2.2 Implementation

Package Imports

The purpose of this method is to uncover hidden themes within the misunderstandings and offer insights into the primary challenges faced by students. It relies on various software packages and libraries, including nltk, sklearn, gensim, and spacy, to preprocess the data, construct the LDA model, and visualize the themes. The method is applied to a dataset comprising student responses, with each response associated with a specific problem. The outcomes are assessed using coherence scores and presented visually using pyLDAvis.

Data Cleaning

The initial steps involve loading the dataset, eliminating irrelevant columns, and transforming the "request time" column into a datetime object. Subsequently, the data is sliced to select a specific timeframe. The sliced data is then grouped based on the problem and combined to form a textual corpus.

Data Preprocessing

Text preprocessing encompasses tokenization, stopword removal, lemmatization, and additional preprocessing procedures. Tokenization, stopword removal, and lemmatization are carried out using the nltk package. The data is converted to lowercase, punctuation marks are removed, and stopwords are eliminated. Additionally, bigrams and trigrams are generated to capture relevant phrases.

Theme Modeling with LDA

The method constructs an LDA model employing the gensim library. Initially, a dictionary is created to map unique words to numeric identifiers. Subsequently, the corpus is transformed into a bag-of-words representation. The LDA model is trained using the corpus, dictionary, and configurable parameters such as the number of themes, passes, and chunk size.

4.2.3 Evaluation

Coherence Score

In the realm of Latent Dirichlet Allocation (LDA), coherence score is a measure employed to assess the interpretability of the generated topics. It gauges the extent to which the prominent words within a topic are logically connected. A higher coherence score indicates more meaningful and coherent topics.

To calculate coherence score in the LDA context, the following steps are commonly followed:

1.

2. Topic modeling: LDA is applied to a given corpus to derive a set of topics. LDA represents documents as blends of topics, where each topic is a distribution of

words.

3. Selection of top words: For each topic, the most probable ‘n‘ words are selected based on their occurrence probability within that topic.

4. Co-occurrence matrix: A matrix is constructed to capture the frequencies of word pairs appearing together within a sliding window of size ‘w‘. This matrix provides a measure of how frequently words co-occur in the same context.

5. Pointwise Mutual Information (PMI): PMI is computed for each word pair based on their co-occurrence frequencies. PMI quantifies the strength of association between two words in a given context and is mathematically defined as:

$$PMI(w_i, w_j) = \log\left(\frac{P(w_i, w_j)}{P(w_i) \cdot P(w_j)}\right) \tag{4.4}$$

   Here, w_i and w_j denote two words, $P(w_i, w_j)$ represents the probability of their co-occurrence, and $P(w_i)$ and $P(w_j)$ are the probabilities of their individual occurrences.

6. Computation of coherence score: The coherence score is then calculated by aggregating the PMI values for all word pairs within a topic and taking their average. Various methods such as C_v, C_p, C_uci, and C_npmi exist for computing the coherence score, employing different weighting and normalization techniques.

The formula for calculating coherence score, specifically using the C_v method, can be expressed as:

$$Coherence = \frac{2}{Comb(n, 2)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} PMI(w_i, w_j) \tag{4.5}$$

Here, n denotes the number of top words selected for each topic, and $Comb(n, 2)$ represents the number of combinations of n elements taken 2 at a time.

It is important to note that specific formulas and methods for coherence score calculation may vary depending on the implementation or research papers. Different methods may take into account factors such as word frequency, normalization techniques, or the integration of background corpus statistics to enhance the coherence evaluation. [9]

4.2.4 Visualization

To visualize the themes, the method utilizes the pyLDAvis package. The prepared LDAvis data is generated, saved, and subsequently loaded for visualization. The resulting visualization facilitates interactive exploration of the themes, their associated keywords, and their distribution.

Chapter 5

Conclusion and Future Work

5.1 Deployment

Once the Algorithm is refined and tested I would like to deploy it and integrate it fully with the intro physics curriculum. Students will complete learning sequencing and we would run the algorithm on the second chances responses. Lecturers will then be given the results of the algorithm which would be a list of common misconceptions students faced when first introduced to the new material. Lecturers will then be advised to put emphasis on these misconceptions during the lecture to help students unlearn these mistakes. This makes way for several tests that can be used to determine how beneficial this line of work can be for student learning.

5.2 Grade Trend Analysis

There are several statistical tests that we can use to compare exam averages from different years of a class or different sections of a class. One option is the two-sample t-test, which is used to determine whether there is a significant difference between the means of two groups. To use the two-sample t-test, I would first collect the exam averages for each year or section of the class and organize them into two groups: one for the first year and one for the second year. Alternatively we can create two groups where one is a section that has the algorithm implemented in their curriculum the other group is a section that does not have this algorithm implemented. I would then calculate the means and standard deviations of each group. Next, I would use the two-sample t-test to determine whether there is a significant difference between the means of the two groups. The t-test will calculate a t-statistic and a p-value, which can be used to determine whether the difference between the means is statistically significant. If the p-value is less than a predetermined significance level (usually 0.05), I can conclude that there is a significant difference between the means of the two groups. This means integrating the program into the curriculum for the intro physics classes is potentially beneficial to student learning.

5.3 Conclusion

In summary, both the K-means algorithm and the Latent Dirichlet Allocation (LDA) algorithm have demonstrated their effectiveness in uncovering misconceptions present in student responses. These algorithms provide valuable insights into the challenges faced by students and enable educators to gain a deeper understanding of these misconceptions. Although further testing is required to validate and improve the algorithms, the initial results indicate their efficiency in discovering hidden patterns and themes within student misconceptions.

The K-means algorithm, a widely used clustering technique, effectively groups student responses based on their similarity, allowing for the identification of common misconceptions. By automatically categorizing responses into distinct clusters, educators can gain valuable insights into prevailing misconceptions and devise targeted interventions to address them. This algorithm significantly reduces the manual effort required to analyze large datasets, making it a time-efficient approach for educators and researchers.

On the other hand, the LDA algorithm, a powerful topic modeling technique, enables a comprehensive exploration of the underlying themes within student misconceptions. By extracting latent topics, the LDA algorithm reveals the primary issues students face and provides a nuanced overview of misconceptions prevalent in the dataset. This algorithm's ability to uncover hidden connections between words and concepts enhances educators' understanding of student misconceptions, facilitating the development of tailored instructional strategies to effectively address them.

The implementation of these algorithms in educational settings can greatly facilitate student learning. By gaining insights into common misconceptions, educators can proactively tailor their teaching strategies to address these misconceptions, thereby improving student comprehension and minimizing persistent misconceptions. Identifying and intervening in misconceptions leads to more effective learning experiences, as students receive personalized support in overcoming their difficulties and acquiring accurate knowledge.

Furthermore, employing these algorithms can contribute to the broader educational community by fostering the sharing of insights and best practices in addressing common misconceptions. The systematic analysis of student responses using these algorithms enables educators to identify overarching patterns and themes that extend beyond individual students or classrooms. Through collaborative efforts and knowledge exchange, educators can develop evidence-based instructional approaches that have a far-reaching impact on student learning outcomes.

While the K-means and LDA algorithms have demonstrated their potential in identifying and understanding misconceptions in student responses, it is crucial to acknowledge the need for further testing and refinement. Ongoing research and evaluation will enhance the accuracy, reliability, and applicability of these algorithms across diverse educational contexts and domains. Through continuous development and validation, these algorithms have the potential to become indispensable tools for educators, revolutionizing the identification, mitigation, and resolution of misconceptions in pursuit of effective education.

Chapter 6

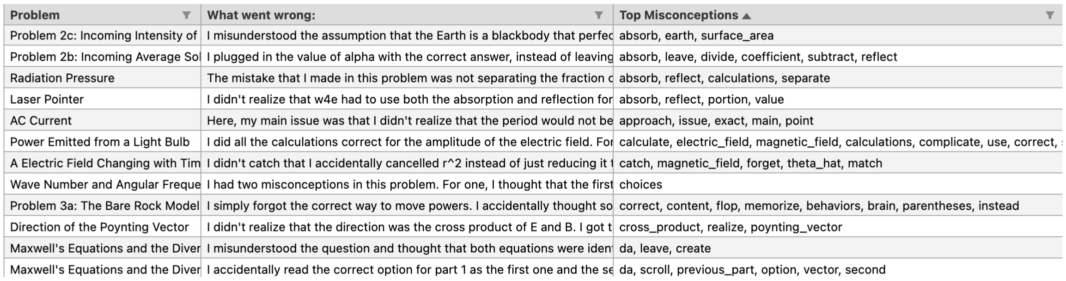
Tables

Table 6.1: Silhouette Score and WCSS value based on number of problems and number of clusters

Number of Problems	**Number of Clusters**	**Silhouette Score**	**WCSS value**
20	10	0.101	6.405
20	5	0.0705	11.750
40	20	0.0387	12.243
40	10	0.0267	24.243

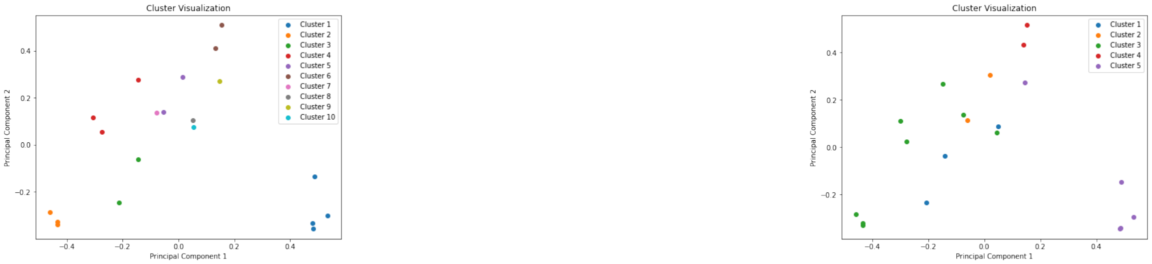
Chapter 7

Figures



Problem	What went wrong:	Top Misconceptions ▲
Problem 2c: Incoming Intensity of	I misunderstood the assumption that the Earth is a blackbody that perfec	absorb, earth, surface_area
Problem 2b: Incoming Average Sol	I plugged in the value of alpha with the correct answer, instead of leaving	absorb, leave, divide, coefficient, subtract, reflect
Radiation Pressure	The mistake that I made in this problem was not separating the fraction c	absorb, reflect, calculations, separate
Laser Pointer	I didn't realize that w4e had to use both the absorption and reflection for	absorb, reflect, portion, value
AC Current	Here, my main issue was that I didn't realize that the period would not be	approach, issue, exact, main, point
Power Emitted from a Light Bulb	I did all the calculations correct for the amplitude of the electric field. For	calculate, electric_field, magnetic_field, calculations, complicate, use, correct,
A Electric Field Changing with Tim	I didn't catch that I accidentally cancelled r^2 instead of just reducing it t	catch, magnetic_field, forget, theta_hat, match
Wave Number and Angular Freque	I had two misconceptions in this problem. For one, I thought that the first	choices
Problem 3a: The Bare Rock Model	I simply forgot the correct way to move powers. I accidentally thought so	correct, content, flop, memorize, behaviors, brain, parentheses, instead
Direction of the Poynting Vector	I didn't realize that the direction was the cross product of E and B. I got t	cross_product, realize, poynting_vector
Maxwell's Equations and the Diver	I misunderstood the question and thought that both equations were iden	da, leave, create
Maxwell's Equations and the Diver	I accidentally read the correct option for part 1 as the first one and the se	da, scroll, previous_part, option, vector, second

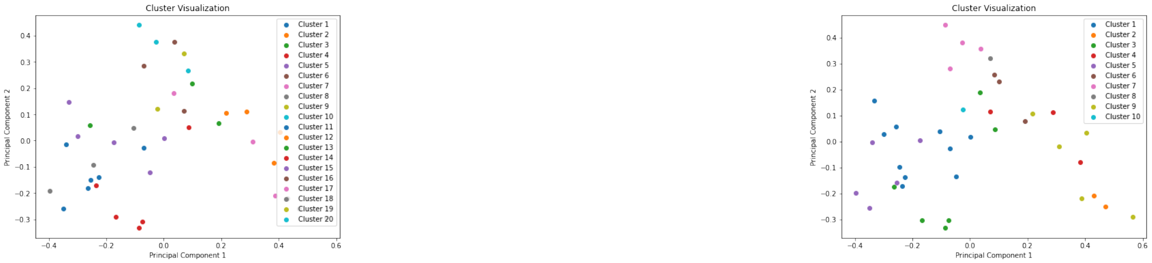
Figure 7-1: sample of kmeans_results.csv for second chance student response data



(a) 20 problems and 10 clusters.

(b) 20 problems and 5 clusters.

Figure 7-2: Cluster Visual for 20 Problem Responses.



(a) 40 problems and 20 clusters.

(b) 40 problems and 10 clusters.

Figure 7-3: Cluster Visual for 40 Problem Responses.

Bibliography

[1] Khaled M Alhawiti. Natural language processing and its use in education. *International Journal of Advanced Computer Science and Applications*, 5(12), 2014.

[2] Michael J Brusco and Douglas Steinley. A comparison of heuristic procedures for minimum within-cluster sums of squares partitioning. *Psychometrika*, 72:583–600, 2007.

[3] Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. Single document automatic text summarization using term frequency-inverse document frequency (tf-idf). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4):285–294, 2016.

[4] Zenun Kastrati, Fisnik Dalipi, Ali Shariq Imran, Krenare Pireva Nuci, and Mudasir Ahmad Wani. Sentiment analysis of students' feedback with nlp and deep learning: A systematic mapping study. *Applied Sciences*, 11(9):3986, 2021.

[5] Andrew Katz, Matthew Norris, Abdulrahman M Alsharif, Michelle D Klopfer, David B Knight, and Jacob R Grohs. Using natural language processing to facilitate student feedback analysis. In *2021 ASEE Virtual Annual Conference Content Access*, 2021.

[6] Gaurav Nanda, Kerrie A Douglas, David R Waller, Hillary E Merzdorf, and Dan Goldwasser. Analyzing large collections of open-ended feedback from mooc learners using lda topic modeling and qualitative analysis. *IEEE Transactions on Learning Technologies*, 14(2):146–160, 2021.

[7] Andra-Selina Pietsch and Stefan Lessmann. Topic modeling for analyzing open-ended survey responses. *Journal of Business Analytics*, 1(2):93–116, 2018.

[8] Ketan Rajshekhar Shahapure and Charles Nicholas. Cluster quality analysis using silhouette score. In *2020 IEEE 7th international conference on data science and advanced analytics (DSAA)*, pages 747–748. IEEE, 2020.

[9] Shaheen Syed and Marco Spruit. Full-text or abstract? examining topic coherence scores using latent dirichlet allocation. In *2017 IEEE International conference on data science and advanced analytics (DSAA)*, pages 165–174. IEEE, 2017.

[10] Yee Teh, Michael Jordan, Matthew Beal, and David Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in neural information processing systems*, 17, 2004.