WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Survey on Large Language Models

M.Ananthavignesh[1], Dr. D. Shanmuga Priyaa[2]

[1]Student, Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Kovaipudur, Coimbatore.
[2]Faculty, Assistant Professor, Department of Computer Science, Sri Krishna Adithya College of Arts and Science, Kovaipudur, Coimbatore.

ABSTRACT

The quantity of survey studies in this topic has significantly expanded in recent years due to the growing popularity of Large Language Models (LLMs). For novice researchers, navigating this vast body of knowledge might be challenging. In this work, I examine the metadata of these LLM survey articles, classifying them according to study focus areas and examining publishing trends. I use straightforward techniques like TFIDF vectorization and one-hot encoding to create a feature matrix. In addition, I utilize a logistic regression model to forecast each paper's category based on its metadata. I use class weighting to boost performance for less prevalent categories in order to overcome the class imbalance in the data. The accuracy increased from 43% to 47%, according to the data, demonstrating the value of class weighting.

1 Introduction

AI methods have been extensively used in a variety of fields, including graphs [Kipf and Welling, 2016; Zhuang and Al Hasan, 2022], texts [Vaswani et al., 2017; Devlin et al., 2018], and photos [He et al., 2016; Dosovitskiy, 2020]. Large Language Models (LLMs) have drawn a lot of attention lately as a crucial subset of AI techniques [Radford et al., 2018, 2019, Brown et al., 2020, Achiam et al., 2023, Bai et al., 2022, Team et al., 2023]. In particular, the study areas pertaining to LLMs are attracting the attention of an increasing number of novices. Survey papers regarding LLMs are frequently read by newcomers to gain knowledge about the latest advancements in this discipline. As a result, several survey papers on LLMs have been released in the previous two years to aid in their learning. However, a lot of these survey studies can be intimidating, making it difficult for inexperienced readers to read them effectively. In order to meet this problem, my project's goal is to investigate and evaluate the LLM survey papers' metadata, offering suggestions to improve their readability and comprehension [Zhuang and Kennington, 2024]. In particular, I want to classify study themes, look at publishing trends, and find gaps in the body of literature in order to methodically examine the metadata of these LLM survey papers.

In general, the following succinctly describes my contributions:

• I categorized research focus areas and found important trends in publication dates by performing a comprehensive analysis of the metadata from LLM survey articles.

• To enable machine learning analysis, I created a feature matrix by applying onehot encoding to categorical data and Term Frequency-Inverse Document Frequency (TFIDF) vectorization to textual data.

• Using classweighted approaches, I addressed class imbalance by using a logistic regression model to categorize survey articles according to their metadata.
• Using metrics like accuracy, precision, recall, and F1-score, I assessed the logistic regression model's performance. I also used a confusion matrix to depict classification performance.

2 Related Work

In recent years, large language models, or LLMs, have advanced quickly. This development is demonstrated by GPT-4 by [Achiam et al., 2023], which shows advancements in text production and managing challenging jobs. Constitutional AI was introduced by [Bai et al., 2022], with an emphasis on leveraging human feedback to make AI systems safer. Few-shot learning was transformed by language models like GPT-3, which allowed models to generalize from sparse data (Brown et al., 2020). For many NLP jobs, the introduction of BERT by Devlin et al. (2018) has also become essential. Relatively few studies have explicitly concentrated on classifying or evaluating survey papers within the LLM field, despite the fact that these papers indicate developments in LLM development. Typically, survey studies give summaries of LLM developments while addressing application domains, ethical issues, and technical difficulties.

Unlike previous works that concentrate on ethical frameworks or technical contributions, this paper tackles the dearth of organized analysis of LLM survey publications. This paper offers a meta-analysis of the survey literature itself, whereas previous studies typically concentrate on the performance and functionality of particular models. By concentrating on taxonomy classification, publication trends, and finding unexplored study topics, my analysis adds a fresh perspective to the corpus of LLM literature.

3 Methodology

I started the study by importing the Pandas package, a robust Python tool for data analysis and manipulation. For simpler processing and manipulation of the structured data, I loaded the dataset—which was saved as a CSV file—into a Pandas DataFrame.

This first step helped confirm that the data had been loaded correctly and gave insight into the structure of the dataset, which had columns such as "Taxonomy," "Title," "Authors," "Release Date," and "Summary." Figure 1 provides an illustration of this analysis. The groundwork for the following phases of data exploration and reporting was established by this preparation. Afterwards, the exploration phase included studying the distribution of research across different taxonomy categories and looking at patterns in survey paper publications over time, with an emphasis on the "Release Date" field. Descriptive statistics were calculated for important fields like "Release Year" and "Taxonomy" in order to better comprehend the data. Term Frequency-Inverse Document Frequency (TF-IDF) vectorization was used to convert textual data from the "Title" and "Summary" fields into numerical representations for the data manipulation phase, while one-hot encoding was used to handle the categorical "Categories" field. A feature matrix was created by combining these changes, and it was subsequently normalized to guarantee that every feature was on the same scale. In the last stage, a logistic regression model was used to forecast each paper's taxonomy category. Metrics including accuracy, precision, recall, and F1-score were used to assess the model's performance. Class-weighting approaches were used to rectify the class imbalance and enhance the model's capacity to categorize underrepresented This method made it easier to create a predictive model for categorizing survey articles on Large Language Models (LLMs) and gave a thorough grasp of the dataset.

3.1 Data Exploration

3.1.1 Survey Paper Trends Over Time

I looked at the dataset's "Release Date" column to see how the number of published papers changed over time in order to assess trends in survey paper publications**.**

Methodology: To facilitate the manipulation of time-based data, the "Release Date" column was first transformed into a datetime format using Pandas. I was able to sort the data by month and determine how many survey articles were published each month by extracting the "YearMonth" component from the converted dates. Visualizing trends in publishing rates required the use of this grouping technique. I created a graphic using Matplotlib, with the number of survey articles published over that time period shown on the Y-axis and the publication year and month represented on the X-axis. Figure 2 provides an illustration of this analysis.

Findings: The plot showed that, beginning in early 2022, there was a slow increase in the number of survey articles released in 2021. Nearly 18 survey papers were published in November 2023, marking a notable peak that happened in late 2023.

This dramatic rise is indicative of the increased interest in and research into LLMs, which is probably being fueled by developments in AI technology. Particularly in the second half of 2023, the trend shows a sharp increase in the number of publications, indicating a surge in research attention on LLMs.

3.1.2 Taxonomy Distribution

In order to determine which study topics were underrepresented and which were dominant, I examined the distribution of survey papers among the different categories within the suggested taxonomy.

	Taxonomy	Title	Authors	Release Date	Links	Paper ID	Categories	Summary
0	Comprehensive	A Comprehensive Survey of AI-Generated Content...	Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yu...	7-Mar-23	https://arxiv.org/abs/2303.04226	2303.04226	cs.AI, cs.CL, cs.LG	Recently, ChatGPT, along with DALL-E-2 and Cod...
1	Comprehensive	Language Model Behavior: A Comprehensive Survey	Tyler A. Chang, Benjamin K. Bergen	20-Mar-23	https://arxiv.org/abs/2303.11504	2303.11504	cs.CL	Transformer language models have received wide...
2	Comprehensive	A Survey of Large Language Models	Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tan...	31-Mar-23	https://arxiv.org/abs/2303.18223	2303.18223	cs.CL, cs.AI	Language is essentially a complex, intricate s...
3	Comprehensive	One Small Step for Generative AI, One Giant Le...	Chaoning Zhang, Chenshuang Zhang, Chenghao Li,...	4-Apr-23	https://arxiv.org/abs/2304.06488	2304.06488	cs.CV, cs.AI, cs.CL, cs.CV, cs.LG	OpenAI has recently released GPT-4 (a.k.a. Cha...
4	Comprehensive	Summary of ChatGPT-Related Research and Perspe...	Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhan...	4-Apr-23	https://arxiv.org/abs/2304.01852	2304.01852	cs.CL	This paper presents a comprehensive survey of ...
...	...	...	...	...	...	...	...	...
139	Others	The Life Cycle of Knowledge in Big Language Mo...	Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun	14-Mar-23	https://arxiv.org/abs/2303.07616	2303.07616	cs.CL	Knowledge plays a critical role in artificial ...
140	Others	Topics, Authors, and Networks in Large Languag...	Rajiv Movva, Sidhika Balachandar, Kenny Peng, ...	20-Jul-23	https://arxiv.org/abs/2307.10700	2307.10700	cs.DL, cs.CL, cs.CY	Large language model (LLM) research is dramati...
141	Others	Document Automation Architectures: Updated Sur...	Mohammad Ahmadi Achachlouei, Omkar Patil, Taru...	18-Aug-23	https://arxiv.org/abs/2308.09341	2308.09341	cs.CL, cs.LG	This paper surveys the current state of the ar...
142	Others	When Large Language Models Meet Citation: A Su...	Yang Zhang, Yufei Wang, Kai Wang, Quan Z. Shen...	18-Sep-23	https://arxiv.org/abs/2309.09727	2309.09727	cs.DL, cs.CL	Citations in scholarly work serve the essentia...
143	Others	A Survey of Large Language Models Attribution	Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Li...	7-Nov-23	https://arxiv.org/abs/2311.03731	2311.03731	cs.CL	Open-domain generative systems have gained sig...

144 rows × 8 columns

Figure 1: Dataset in a tabular format

Methodology:

I used the value_counts() function on the "Taxonomy" column to determine the frequency of papers attributed to each category. A clear picture of the number of papers classified under each taxonomy term was provided by the generated counts. To display the distribution, a bar chart was made with Matplotlib. Taxonomy categories were represented by the X-axis, and the quantity of papers in each category was displayed on the Y-axis. This analysis is illustrated in Figure 3.

Results: With 26 articles, the "Trustworthy" category was the most prevalent, according to the bar chart. "Comprehensive" and "Prompting," which both displayed a comparatively large number of papers, were other commonly represented categories. Nonetheless, the underrepresentation of categories like "Law," "Finance," and "Education" indicates that fewer survey articles addressed these topics. The findings indicated possible inadequacies in other domains, such as banking and education, where fewer studies were published, while highlighting the research focus on prompting and credibility in LLMs.

3.1.3 Descriptive Statistics on Release Years

I calculated descriptive statistics on the "Release Year" column in order to gain a better understanding of the survey articles' temporal distribution. This analysis shed light on the data's distribution and central patterns.

Methodology:

The "Release Year" was retrieved for every paper after the "Release Date" column was first transformed from string format into a datetime format using Pandas. For the "Release Year" data, I calculated a number of important statistical measures:

Mean: A measure of the central tendency that provides information on the average year of publication.

The data's concentration is made clearer by the median, which is the midpoint year of publication.

The distribution of publishing years around the mean is described by variance and standard deviation.

Range and Interquartile Range (IQR): These figures show the distribution of the middle 50% of the data as well as the earliest and latest years of publication.

Quantiles: To determine the distribution of the data, the 25th, 50th, and 75th percentiles were computed.

Results:

This analysis is illustrated in Figure 4. The computed descriptive statistics showed a mean release year of 2023, confirming that the majority of survey papers were published recently. The skewness value suggested a slight rightward skew, indicating that most publications were concentrated in the last few years, especially in 2023 and 2024. This analysis provided a clearer picture of the publication trends over time and helped interpret the temporal aspects of LLM-related research.

```
Mean Release Year: 2023.0555555555557
Standard Deviation: 0.40586224845896185
Variance: 0.1647241647241641
Median Release Year: 2023.0
Mode of Release Year: 2023
Min Release Year: 2021, Max Release Year: 2024
Range of Release Years: 3
Interquartile Range (IQR): 0.0
Skewness of Release Year: -0.20690653476862192
Count of Release Years: 144
Sum of Release Years: 291320
Quantiles of Release Years (25%, 50%, 75%):
0.25    2023.0
0.50    2023.0
0.75    2023.0
Name: Release Year, dtype: float64
```

Figure 4: Descriptive Statistics of Survey Paper Release years

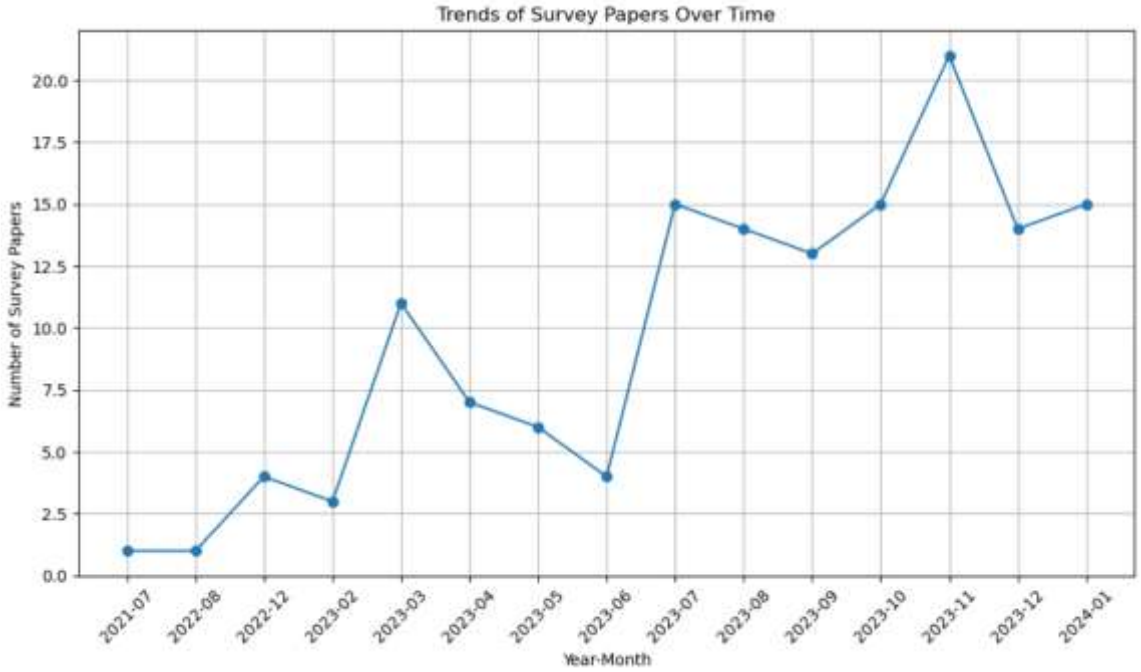


Figure 2Trends in Survey Paper Publications Over Time

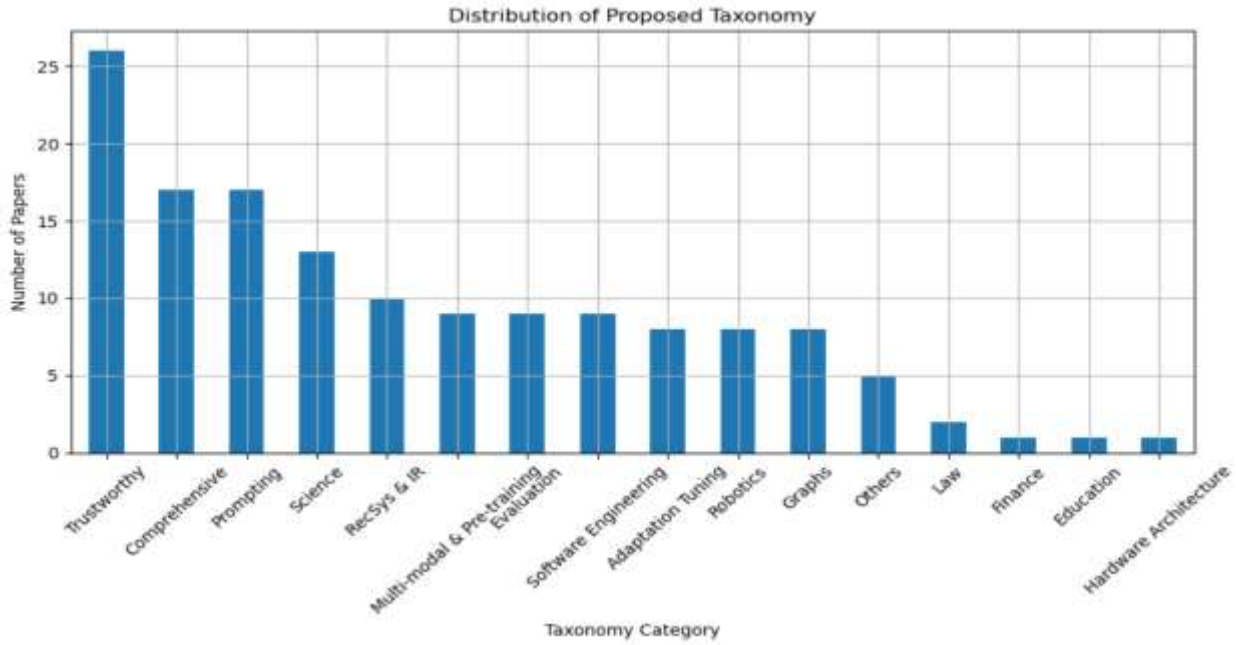


Figure 3: Distribution of Survey Papers by Taxonomy Category

3.1.4 Visualizing Categorical and Temporal Data

I looked at the categories' frequency distribution and the survey papers' release years' temporal distribution in the last section of the data exploration process. Figures 6, 7, and 8 provide illustrations of this analysis.

Methodology:

In order to determine which categories were most represented in the categorical data, I calculated frequency counts for the "Categories" column.

With 28 and 27 papers, respectively, this count showed that categories like "cs.CL" and "cs.AI" were well-represented.

I made a bar chart to show the distribution of categories in order to visualize the results. According to the figure, the majority of the papers fell into a small number of categories, with only a few others being somewhat represented.I created a cumulative distribution plot and a histogram of release years to examine the temporal data. The frequency of articles published in each year was represented by the histogram, and the cumulative distribution plot provided an overall view of how the number of publications increased over time.

Findings: According to the histogram and cumulative distribution plot, the majority of survey papers were released in 2023, with only a small number appearing in 2021 and 2022. Beginning in 2023, the number of publications increased quickly, indicating a notable growth in the field's research activities.

The integration of temporal and categorical visualizations provided insightful informatioabout the timing and areas of focus of study in studies pertaining to LLM.

```
Categories Frequency Distribution:
Categories
cs.CL, cs.AI                          28
cs.CL                                 27
cs.CL, cs.AI, cs.LG                   11
cs.SE                                  7
cs.AI                                  6
cs.CL, cs.LG                           4
cs.IR, cs.AI, cs.CL                    4
cs.AI, cs.CL                           4
cs.IR, cs.AI                           3
cs.AI, cs.CL, cs.LG                    2
cs.CL, cs.AI, cs.CY, cs.LG             2
cs.AI, cs.LG                           2
cs.LG, cs.AI, cs.SI                    1
cs.CL, cs.AI, cs.CR                    1
cs.AI, cs.CL, cs.CY, cs.MA             1
cs.RO, cs.AI                           1
cs.LG, cs.CL, cs.SI                    1
cs.SE, cs.AI, cs.CL, cs.PL             1
cs.CL, cs.IR                           1
cs.SE, cs.AI                           1
cs.SE, cs.HC                           1
cs.AR, cs.CL, cs.LG                    1
cs.LG, cs.AI, cs.CL                    1
cs.DL, cs.CL, cs.CY                    1
cs.LG, cs.AI                           1
cs.LG                                  1
cs.LG, cs.AI, cs.DC, cs.PF             1
cs.CL, cs.CR, cs.LG                    1
cs.IR, cs.AI, cs.SE                    1
cs.CR                                  1
cs.CY, cs.AI, cs.CL, cs.CV, cs.LG      1
cs.DL, cs.CL, cs.CY, cs.SI             1
cs.CL, cs.AI, cs.CV                    1
cs.CV, cs.AI, cs.CL, cs.LG             1
cs.CL, cs.AI, cs.CV, cs.MM             1
cs.CV, cs.AI                           1
cs.CV, cs.CL                           1
cs.CV                                  1
cs.CL, cs.AI, cs.CV, cs.HC, cs.MA      1
cs.NE, cs.AI, cs.CL                    1
cs.CL, cs.AI, cs.CY                    1
cs.CY, cs.AI, cs.CL, cs.LG             1
cs.AI, cs.CL, cs.IR                    1
cs.HC                                  1
cs.CY                                  1
cs.DL, cs.CL                           1
Name: count, dtype: int64
```

Figure 5: Categories Frequency Distribution

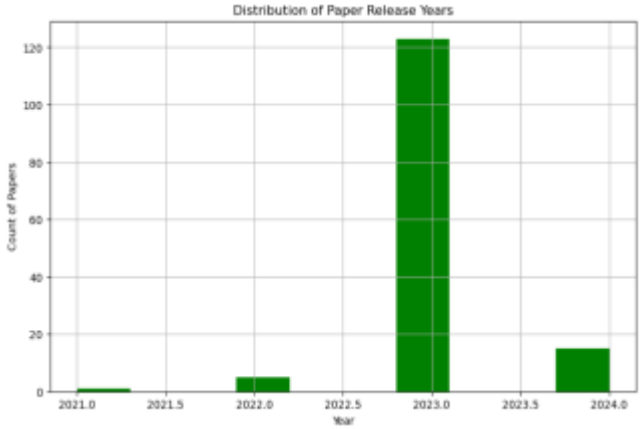


Figure 6: Cumulative Distribution of Survey Paper Publications Over Time

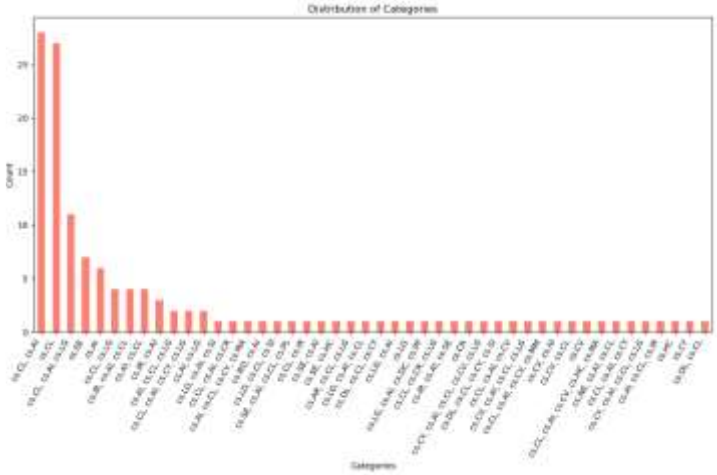


Figure 7: Bar Chart of Categories Frequency Distribution

3.2 Data Manipulation

3.2.1 Building a Feature Matrix

During this phase of the analysis, I converted textual and categorical data into numerical representations for the dataset's feature matrix, which might be utilized for additional modeling or analysis. Techniques like one-hot encoding and Term Frequency-Inverse Document Frequency (TF-IDF) vectorization were used in this process.

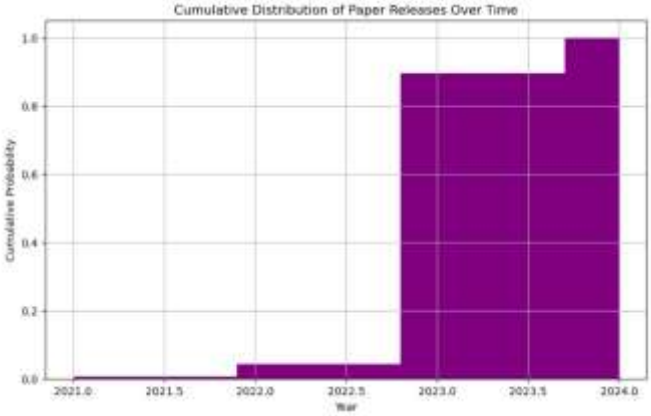


Figure 8: Cumulative Distribution of Paper ReleaseYears Over Time

Methodology:

The Dataset is being loaded: In order to manipulate and analyze the dataset, I first loaded it from the CSV file into a Pandas DataFrame. Constructing the Matrix of Features: I used the following transformations to produce a strong feature matrix:

Text Data Vectorization with TF-IDF: The TF-IDF vectorizer was used to change the textual data in the "Title" and "Summary" columns.

Using this procedure, the textual data is transformed into numerical vectors that represent each word's significance in relation to the other words in the dataset. Following that, each paper title and summary's TF-IDF scores were saved in a dense matrix format. One-Hot Encoding of Categorical Data: The one-hot encoding approach was used to analyze the categorical data in the "Categories" column. Each distinct category is converted into a binary feature using this procedure, where a value of "1" denotes the existence of a category for a given paper and a value of "0" denotes its absence. The resulting one-hot encoded matrix was then merged with the TF-IDF matrices.

Combining characteristics I created a single feature matrix by concatenating the vectorized text and categorical data into a single matrix. This final matrix produced a thorough numerical representation of the dataset by incorporating the one-hot encoded "Categories" column, the TF-IDF representations of the "Title" and "Summary" columns, and more.

Feature Matrix Display: Lastly, I showed the combined feature matrix (figure 9), which contained 3,748 columns that represented every feature that was taken out of the categories and textual data. The dataset's complexity is reflected in its huge number of columns, each of which represents a binary category label or a distinct word in the text.

3.2.2 Normalization of Feature Matrix

To make sure all the features were on the same scale and ready for additional machine learning algorithms, I normalized the feature matrix in the next stage of data processing. Since many machine learning models are sensitive to the magnitude of several features, normalization is crucial. This procedure makes sure that no feature's scale causes it to dominate the others.

Methodology:

Boolean Column Conversion: To start, I changed the feature matrix's boolean columns to represent numeric values (1s and 0s). To do this, the pd.to_numeric() method was used to make sure that every matrix column was numeric and prepared for scaling. To ensure a clean dataset, any conversion problems were fixed by appending zeros to the missing values.

Using MinMaxScaler for normalization: I used Scikit-learn's MinMaxScaler to normalize the numerical data, which scales all features to a range of 0 to 1. To maintain the structure of the feature matrix, the MinMaxScaler was only applied to the DataFrame's numerical values. Regardless of the features' initial scale, this scaling guarantees that they are equivalent**.**

Rebuilding the DataFrame: The column names and structure of the original feature matrix were preserved when the scaled values were transformed back into a Pandas DataFrame following normalization. This guaranteed compatibility with subsequent processes and simplified the interpretation of the scaled values.

Seeing the Normalized Matrix: Figure 10 showed the normalized matrix that was produced, which indicated that all values in the 3,748 columns (features) were now scaled from 0 to 1. This stage verified that the dataset had been correctly standardized by the normalization procedure, guaranteeing that every feature made an equal contribution to any further modeling or analysis.

In order to guarantee that no one characteristic would disproportionately impact the output of machine learning models, this normalization step was essential.

The data was prepared for a range of machine learning approaches, which frequently presume normalized input features for maximum performance, by scaling all characteristics to the same range.

3.2.3 Encoding Labels Using Label Encoder

I converted the categorical "Taxonomy" labels into numerical values using Scikit-learn's LabelEncoder in the following data manipulation step. In order to feed non-numeric data into machine learning algorithms—which normally demand numerical input—this modification is necessary. Figure 11 provides an illustration of this analysis**.**

Methodology:

Label Encoder Initialization: To transform the categorical labels in the "Taxonomy" column, I first initialized the Label Encoder. I set the variable y to the "Taxonomy" column, which listed the various categories to which each paper was assigned.

Label Fitting and Transformation: The taxonomy labels were then fitted and transformed using the Label Encoder. Each distinct taxonomy category was encoded as an integer by this process. For instance, the label "Comprehensive" had the code "1," whereas the label "Trustworthy" had the value "7." I was able to numerically express the category labels thanks to this modification, which improved the dataset's suitability for machine learning models.

Checking Encoded Labels: In order to confirm that the encoding procedure was accurate, I printed the original labels as well as the encoded labels after the change. This stage made sure that the original taxonomy categories and their related integers were accurately mapped. Inverse Transforming Labels: I also mapped the encoded numbers back to their original categories labels using the Label Encoder's inverse_ transform method in order to verify the encoding.

This enabled me to verify that, in the event that it became required, the encoded labels could be accurately restored to their original taxonomy categories.

In order to convert the categorical taxonomy data into a numerical format—a prerequisite for many machine learning algorithms—this encoding step was essential. I was able to make the data more computationally manageable for further analysis or modeling jobs while preserving the crucial categorical information by using Label Encoder.

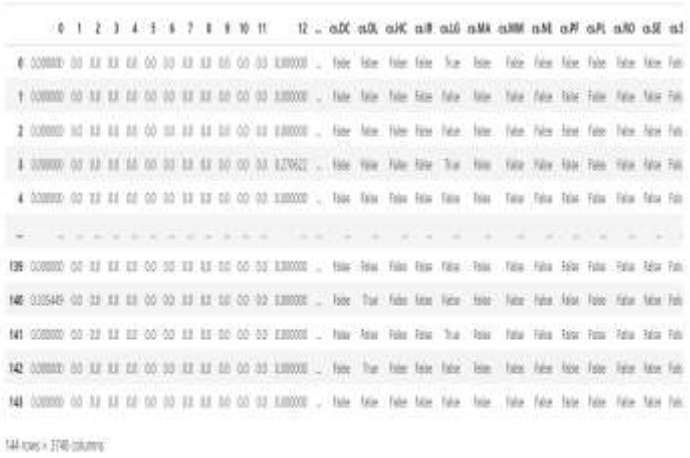


FIGURE 9 Featured Martix

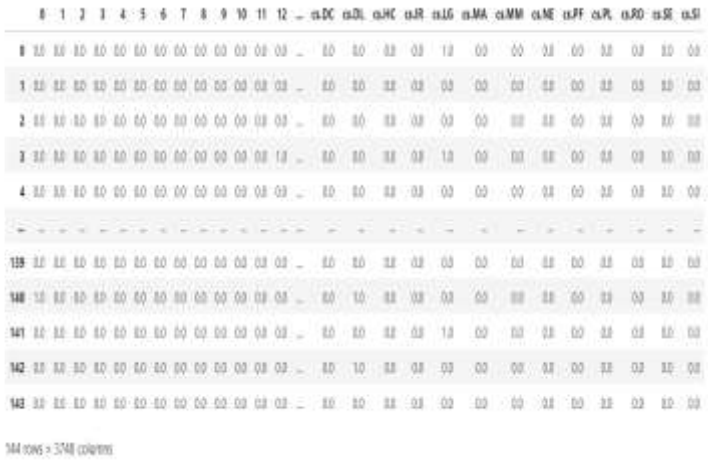


Figure 10: Normalized matrix

```
Encoded labels: [ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  8  8  8  8  8  8  8
  8  8  0  0  0  0  0  0  0  0 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
 15 15 15 15 15  3  3  3  3  3  3  3  3 13 13 13 13 13 13 13 13 13 13
 13 13 13 11 11 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12  5  5  5
  5  5  5  5  5 14 14 14 14 14 14 14 14 14  7  7  4  2  6  9  9  9  9  9]
Original labels: ['Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Adaptation Tuning' 'Adaptation Tuning'
 'Adaptation Tuning' 'Adaptation Tuning' 'Adaptation Tuning'
 'Adaptation Tuning' 'Adaptation Tuning' 'Adaptation Tuning' 'Prompting'
 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting'
 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting'
 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Evaluation' 'Evaluation' 'Evaluation' 'Evaluation' 'Evaluation'
 'Evaluation' 'Evaluation' 'Evaluation' 'Evaluation' 'Science' 'Science'
 'Science' 'Science' 'Science' 'Science' 'Science' 'Science' 'Science'
 'Science' 'Science' 'Science' 'Science' 'RecSys & IR' 'RecSys & IR'
 'RecSys & IR' 'RecSys & IR' 'RecSys & IR' 'RecSys & IR' 'RecSys & IR'
 'RecSys & IR' 'RecSys & IR' 'RecSys & IR' 'Robotics' 'Robotics'
 'Robotics' 'Robotics' 'Robotics' 'Robotics' 'Robotics' 'Robotics'
 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs'
 'Software Engineering' 'Software Engineering' 'Software Engineering'
 'Software Engineering' 'Software Engineering' 'Software Engineering'
 'Software Engineering' 'Software Engineering' 'Software Engineering'
 'Law' 'Law' 'Finance' 'Education' 'Hardware Architecture' 'Others'
 'Others' 'Others' 'Others' 'Others']
```

Figure 11: Visualization of Encoded and Original Taxonomy Labels for Survey Papers

```
Training data shape: (86, 3748), Training labels shape: (86,)
Test data shape: (58, 3748), Test labels shape: (58,)
```

Figure 12: Dataset Overview: Training and Testing Data Shapes

3.2.4 Splitting the Data into Training and Testing Sets

In this step, I prepared the dataset for model training by splitting it into training and testing sets. This process is crucial for evaluating how well a machine learning model generalizes to unseen data by using one portion of the data for training and another for testing. I used the train_test_split function from Scikit-learn to carry out this operation.This analysis is illustrated in Figure 12.

Methodology:

Defining the Test Ratio: I set the test ratio to 0.4, meaning that 40% of the data would be reserved for testing, and the remaining 60% would be used for training. This split ratio ensures that a significant portion of the data is set aside to evaluate the model's performance on unseen data.

Splitting the Data: The train_test_split function was used to split the normalized feature matrix (X) and the encoded labels (y_encoded). The function randomly split the data according to the specified ratio while maintaining the structure and shape of the data. A random seed

(random_state=42) was set to ensure reproducibility of the split.

- X_train and y_train: These represent the training set features and labels. - X_test and y_test: These represent the testing set features and labels.

Verifying the Split: After splitting the data, I checked the shapes of both the training and testing sets to ensure that the split was done correctly. The training data had 86 samples, while the testing data had 58 samples, consistent with the 60/40 split.

- Training Data Shape: (86, 3748), meaning 86 samples with 3,748 features each. - Testing Data Shape: (58, 3748), meaning 58 samples with 3,748 features each.

This step was essential for preparing the data for machine learning, ensuring that the model would be trained on one portion of the data and validated on another to avoid overfitting and assess its generalization performance.

3.3 Data Evaluation

3.3.1 Logistic Regression Model

To evaluate the model's performance, a logistic regression algorithm was implemented to predict the taxonomy category for each paper in the dataset. Logistic regression is suitable for multi-class classification problems and provides insight into the relationships between features and labels. This analysis is illustrated in Figure 13. Methodology:

- *Converting Column Names:* The column names in the feature matrix were converted to strings to ensure compatibility with Scikitlearn.
- *Model Initialization:* The logistic regression model was initialized using Scikit-learn's LogisticRegression function. The max parameter was set to 1000 to allow sufficient iterations for model convergence.
- *Training the Model:* The model was trained on the training dataset using the fit() method to establish patterns between features and taxonomy labels.
- *Making Predictions:* After training, predictions were made on the test dataset using the predict() method, aiming to classify each paper into its respective taxonomy category.
- *Evaluating Performance:* Model performance was evaluated using the accuracy_score function, with the overall accuracy measured at 43%. A classification report was generated, providing precision, recall, and F1-score for each category.
- *Addressing Class Imbalance:* The label distribution in both the training and testing
- datasets was reviewed to assess class imbalance, which can negatively impact model performance, especially for underrepresented categories.

The model's performance highlighted areas for improvement, particularly regarding class imbalance and the need for more complex models to enhance predictive accuracy.

3.3.2 Confusion Matrix and Mode

l Visualization To further assess the logistic regression model, a confusion matrix was generated. This matrix provided detailed insights into the number of correct and incorrect classifications for each taxonomy category. This is illustrated in Figure 14 and Figure 15. Methodology: • Classification Report: A classification report was created to summarize precision, recall,Figure 13: Classification Report and Label Distributionand F1-scores for each class, offering a more comprehensive view of the model's performance.

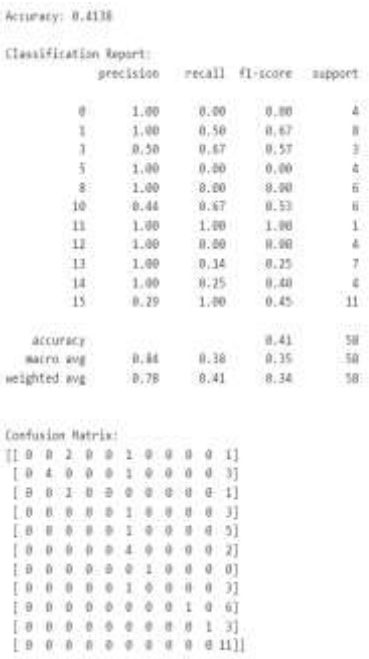
• Confusion Matrix: A confusion matrix was computed using the confusion matrix unction to compare true labels with predicted labels across all classes.

• Confusion Matrix Visualization: The confusion matrix was visualized as a heatmap using the Seaborn library. The darker colors along the diagonal of the heatmap indicated correctly predicted classes, while off-diagonal elements reflected misclassifications.

• Model Accuracy: The overall model accuracy remained consistent at 43 The confusion matrix and visual representation helped identify which classes were misclassified more frequently, highlighting potential areas where the model could be improved

```
Accuracy: 0.4138

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.00      0.00         4
           1       1.00      0.50      0.67         8
           3       0.50      0.67      0.57         3
           5       1.00      0.00      0.00         4
           8       1.00      0.00      0.00         6
          10       0.44      0.67      0.53         6
          11       1.00      1.00      1.00         1
          12       1.00      0.00      0.00         4
          13       1.00      0.14      0.25         7
          14       1.00      0.25      0.40         4
          15       0.29      1.00      0.45        11

    accuracy                           0.41        58
   macro avg       0.84      0.38      0.35        58
weighted avg       0.78      0.41      0.34        58


Training set label distribution:
15    15
10    11
1      9
11     9
3      6
13     6
9      5
14     5
0      4
12     4
5      4
8      3
7      2
2      1
4      1
6      1
Name: count, dtype: int64

Test set label distribution:
15    11
1      8
13     7
8      6
10     6
5      4
14     4
0      4
12     4
3      3
11     1
Name: count, dtype: int64
```

Figure 13: Classification Report and Label Distribution



```
Accuracy: 0.4138

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.00      0.00         4
           1       1.00      0.50      0.67         8
           3       0.50      0.67      0.57         3
           5       1.00      0.00      0.00         4
           8       1.00      0.00      0.00         6
          10       0.44      0.67      0.53         6
          11       1.00      1.00      1.00         1
          12       1.00      0.00      0.00         4
          13       1.00      0.14      0.25         7
          14       1.00      0.25      0.40         4
          15       0.29      1.00      0.45        11

    accuracy                           0.41        58
   macro avg       0.84      0.38      0.35        58
weighted avg       0.78      0.41      0.34        58


Confusion Matrix:
[[ 0  0  2  0  0  1  0  0  0  0  1]
 [ 0  4  0  0  0  1  0  0  0  0  3]
 [ 0  0  1  0  0  0  0  0  0  0  1]
 [ 0  0  0  0  0  1  0  0  0  0  3]
 [ 0  0  0  0  0  1  0  0  0  0  5]
 [ 0  0  0  0  0  4  0  0  0  0  2]
 [ 0  0  0  0  0  0  1  0  0  0  0]
 [ 0  0  0  0  0  1  0  0  0  0  3]
 [ 0  0  0  0  0  0  0  0  1  0  6]
 [ 0  0  0  0  0  0  0  0  0  1  3]
 [ 0  0  0  0  0  0  0  0  0  0 11]]
```

Figure 14: Classification Report with Confusion Matrix

3.3.3 Bonus Question: Logistic

Regression with Class Weighting

Given the class imbalance in the dataset, class

weighting was applied to the logistic regresssion model to improve performance, particularly for underrepresented categories. This isillustrated in Figure 16.

Methodology:

– Installing Imbalanced-Learn: The imbalanced-learn library was installed to manage class imbalances during training.

– Initializing Logistic Regression with Class Weighting: The logistic regression model was re-initialized with the class_weight=’balanced’ parameter, ensuring that classes were weighted inversely proportional to their frequency in the dataset.

– Training the Model: The model was retrained on the dataset with the adjusted class weights to address the imbalance and improve classification of underrepresented categories.

– Making Predictions and Evaluating Performance: Predictions were made on the test dataset, and the accuracy improved to 47%. Another classification report was generated to evaluate improvements in the precision, recall, and F1-score for underrepresented classes.

Results: The adjusted logistic regression model showed a modest improvement in overall accuracy, from 43% to 47%, and improved performance on underrepresented classes, although certain categories still exhibited lower performance.

```
Accuracy with class weighting: 0.4655

Classification Report with class weighting:
              precision    recall  f1-score   support

           0       1.00      0.00      0.00         4
           1       1.00      0.62      0.77         8
           3       0.40      0.67      0.50         3
           5       1.00      0.00      0.00         4
           8       1.00      0.00      0.00         6
          10       0.31      0.67      0.42         6
          11       1.00      1.00      1.00         1
          12       1.00      0.50      0.67         4
          13       1.00      0.14      0.25         7
          14       1.00      0.25      0.40         4
          15       0.37      1.00      0.54        11

    accuracy                           0.47        58
   macro avg       0.82      0.44      0.41        58
weighted avg       0.78      0.47      0.40        58


Confusion Matrix with class weighting:
[[ 0  0  2  0  0  1  0  0  0  0  1]
 [ 0  5  0  0  0  2  0  0  0  0  1]
 [ 0  0  2  0  0  0  0  0  0  0  1]
 [ 0  0  0  0  0  1  0  0  0  0  3]
 [ 0  0  0  0  0  3  0  0  0  0  3]
 [ 0  0  0  0  0  4  0  0  0  0  2]
 [ 0  0  0  0  0  0  1  0  0  0  0]
 [ 0  0  0  0  0  1  0  2  0  0  1]
 [ 0  0  1  0  0  1  0  0  1  0  4]
 [ 0  0  0  0  0  0  0  0  0  1  3]
 [ 0  0  0  0  0  0  0  0  0  0 11]]
```

Figure 16: Class weighing

4 Conclusion

In this report, I analyzed survey papers on Large Language Models (LLMs) by exploring metadata, categorizing research areas, and using machine learning to predict taxonomy categories. The analysis highlighted publication trends, with significant research growth in 2023, particularly in "Trustworthy" and "Prompting" areas.

I implemented a logistic regression model, initially achieving 43% accuracy, which improved to 47% after applying class-weighting techniques to address class imbalance. Although this improved classification for underrepresented categories, further refinement is needed to enhance performance in some areas. This study provides a foundation for understanding LLM survey papers and offers a machine learning-based approach to help researchers identify key trends and gaps in the field. Future work could involve more advanced models or additional metadata to improve classification and insight into LLM research

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. arXiv preprint arXiv:2212.08073, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.

In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

Thomas N Kipf and Max Welling. Semi supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. OpenAI blog, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.