# Navigating the Taxonomy of Large Language Models: A Comparative Exploration through Data Manipulation and Evaluation Techniques

**Sharadha Kasiviswanathan**
Department of Computer Science
Boise State University
sharadhakasivisw@u.boisestate.edu

## Abstract

As the landscape of research on Large Language Models (LLMs) rapidly evolves, understanding the taxonomy of these models becomes increasingly crucial for researchers. This study explores the automatic classification of survey papers related to LLMs, utilizing graph representation learning and various data manipulation techniques. By collecting and analyzing metadata from 144 literature reviews, we construct co-category graphs to evaluate and compare the effectiveness of different classification paradigms, including pre-trained language models and graph neural networks. Our findings indicate that leveraging graph structures significantly enhances taxonomy classification performance compared to traditional language models. Additionally, we demonstrate the advantages of using weak labels generated from smaller models, revealing new insights into weak-to-strong generalization. This research not only contributes to the understanding of LLM taxonomy but also provides a framework for future explorations in the field, highlighting the importance of innovative evaluation methods in navigating the complexities of LLM research.

## 1 Introduction

AI techniques have been widely applied to various domains, such as images (He et al., 2016; Dosovitskiy, 2020), texts (Vaswani et al., 2017; Devlin et al., 2018), and graphs (Kipf and Welling, 2016; Zhuang and Al Hasan, 2022). As a critical subset of AI techniques, Large Language Models (LLMs) have gained significant attention in recent years (Radford et al., 2018, 2019; Brown et al., 2020; Achiam et al., 2023; Bai et al., 2022; Team et al., 2023). Especially, more and more new beginners are interested in the research topics about LLMs. To learn the recent progress in this field, new beginners commonly will read survey papers about LLMs. Therefore, to facilitate their learning, numerous survey papers on LLMs have been published in the last two years. However, a large amount of these survey papers can be overwhelming, making it challenging for new beginners to read them efficiently. To embrace this challenge, in this project, we aim to explore and analyze the metadata of LLMs survey papers, providing insights to enhance their accessibility and understanding (Zhuang and Kennington, 2024). Specifically, we aim to develop a comprehensive framework for analyzing and exploring the data to uncover underlying patterns, manipulating the data to enhance its quality and relevance, and evaluating the results to ensure robust and actionable insights. After analyzing the data, our plan includes the following key components:

**Exploration:** Conducting an in-depth analysis to identify trends and anomalies within the dataset.

**Manipulation:** Applying techniques to clean, transform, and prepare the data for analysis, ensuring its integrity and suitability for our required objectives.

**Evaluation:** Assessing the effectiveness of our methods and results through statistical measures and validation techniques to confirm their reliability and significance.

Overall, our contributions can be summarized as follows which will be covered more in detail under the methodology section:

- Data Exploration
- Data Manipulation
- Data Evaluation

## 2 Related Work

This study builds on the emerging body of research focused on automating the classification of survey papers in the rapidly evolving field of Large Language Models, leveraging graph representation learning to enhance taxonomy assignment (Zhuang and Kennington, 2024).

## 3 Methodology

### 3.1 Data Exploration

In this section, we delve into the data exploration process conducted on the survey dataset to uncover insights into research trends, taxonomy distribution, author contributions, and thematic content of papers. The dataset comprises various attributes, including release dates, taxonomy categories, authors, and summaries, which facilitate a comprehensive analysis.

**A. Dataset Loading and Overview:** The dataset was loaded using the Pandas library. The initial step involved importing the CSV file containing survey data, which included critical columns necessary for subsequent analyses. This dataset serves as the foundation for understanding trends in survey papers, categorization, and author contributions over time.

**B. Trends Over Time:** To analyze the release trends of survey papers, we converted the 'Release Date' column to a datetime format and subsequently grouped the data by year and month. This process allowed us to visualize the number of papers released each month. As a result, a line plot was generated to illustrate the trends in survey papers over time. Each data point was marked for clarity, enabling easy identification of peaks and troughs in the publication rate. The findings revealed interesting patterns in research activity, which may correlate with emerging topics or advancements in the field as shown in Figure 1. Additionally, we calculated the mean number of surveys released per month, which provided a quantitative measure of research activity during the study period. The analysis indicated that the average number of surveys published monthly was approximately 9.6, suggesting sustained interest and productivity in the field.

**C. Taxonomy Distribution:** The distribution of taxonomy categories assigned to survey papers was evaluated by counting the frequency of each category. A bar chart was created to visually represent this distribution, highlighting the diversity of topics explored in the surveys as shown in Figure 2. Among the categories, the "Trustworthy" classification was particularly noteworthy, with a total of 26 papers assigned to this category, showcasing a significant focus on reliability in research findings.

**D. Further Exploration:**

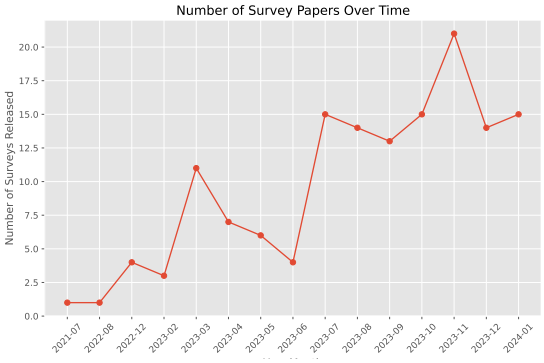


Figure 1: Survey Trends

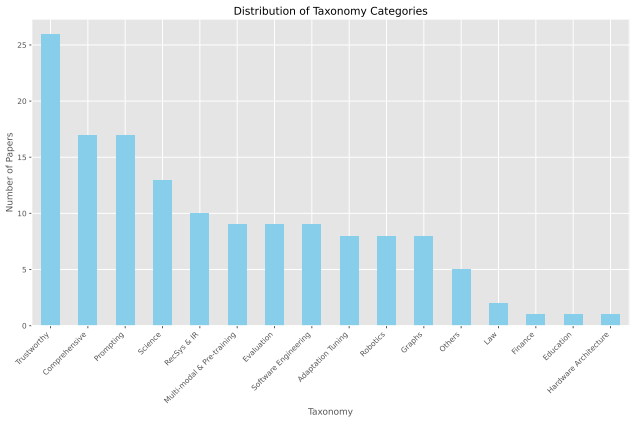


Figure 2: Taxonomy Distribution Bar Chart

- **Histogram of Release Dates**

  To gain insights into the annual distribution of survey papers, a histogram of the release years was plotted. This visualization demonstrated the concentration of papers published in certain years, providing context to research trends which can be observed from the Figure 3.

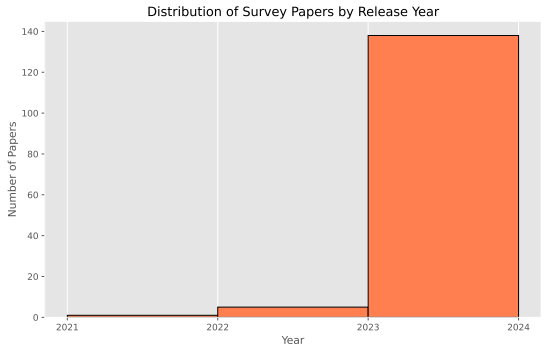


Figure 3: Release Year Histogram

- **Category Distribution Pie**

To further explore research topics, a pie chart was generated to illustrate the distribution of research categories. Each category was represented proportionally, with external labels detailing the percentage share of each category within the dataset as shown in Figure 4.

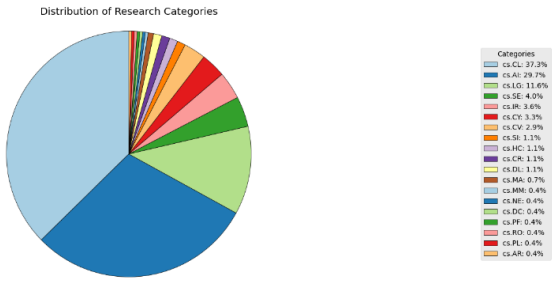


Figure 4: Category Distribution Pie Chart

- **Author Contributions**

  A violin plot was used to visualize the contributions of the top authors in terms of the number of papers published. This plot highlighted disparities in author output, revealing key contributors in the field as depicted from the Figure 5.

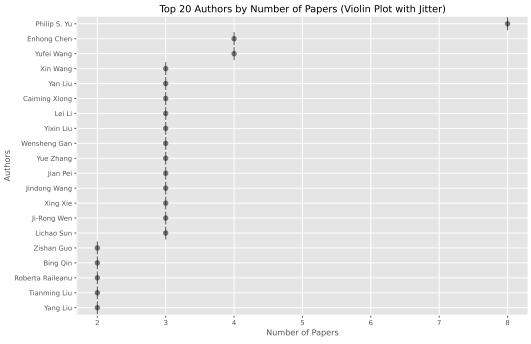


Figure 5: Contribution of Authors

- **Word Cloud Analysis**

  Finally, a word cloud was generated from the summaries of the papers, providing a visual representation of the most frequently mentioned terms. This technique helps in understanding prevalent themes and topics in the research literature which could be observed from the Figure 6.

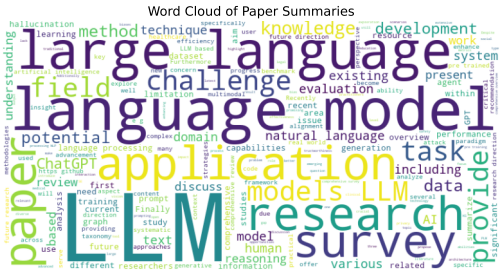


Figure 6: Word Cloud

### 3.2 Data Manipulation

1. **Building the Feature Matrix:** The first step in data manipulation was to create a feature matrix from the dataset using the appropriate function. This function transforms three key features: Title, Summary, and Categories.

   - **Title and Summary Vectorization:** The `TfidfVectorizer` from the `sklearn.feature_extraction.text` module was used to convert the text data in the `'Title'` and `'Summary'` columns into numerical format. This transformation creates a Term Frequency-Inverse Document Frequency (TF-IDF) representation, where each unique word is represented by a vector. The `stop_words='english'` parameter is employed to exclude common English words that do not contribute significant meaning (e.g., "the", "is").

   - **One-Hot Encoding of Categories:** The 'Categories' column was processed by splitting the string of categories into separate columns using the `str.split` method. Following that, the `pd.get_dummies()` function was utilized to perform one-hot encoding, generating binary columns for each category. This allows the model to recognize the presence of each category as a separate feature.

   - **Combining Features:**
     The feature matrices from the TF-IDF transformations and the one-hot encoded categories were combined into a single feature matrix using `pd.concat()`. This

results in a comprehensive feature set that encapsulates information from both textual data and categorical labels.

2. **Normalizing the Data:**

   Normalization was applied to the feature matrix using the `MinMaxScaler` from the `sklearn.preprocessing` module. This process rescales the feature values to a range of [0, 1], ensuring that all features contribute equally to the model training. Normalization is particularly important in machine learning as it improves convergence during training and mitigates the influence of features with larger ranges.

3. **Encoding the Labels:**

   The categorical labels in the 'Categories' column were encoded using the `LabelEncoder`. This encoding converts the string labels into integers, making them suitable for classification tasks. Each unique category is assigned a distinct integer value, which facilitates the training of machine learning algorithms.

4. **Splitting the Dataset:** The dataset was divided into training and testing sets using the `train_test_split` function from the `sklearn.model_selection` module. A test ratio of 0.4 was specified, meaning that 40% of the data would be allocated for testing while the remaining 60% would be used for training the model. The random state was set to 42 in order to ensure reproducibility of the split.

5. **Verification of Dataset Shapes:** After splitting the dataset, the shapes of the resulting training and testing feature matrices and labels were printed to verify the integrity of the operation. The shapes confirm that the dataset was correctly divided, with 86 samples in the training set and 58 samples in the testing set.

   **Output:**

   Training features shape: (86, 3547); Testing features shape: (58, 3547); Training labels shape: (86,); Testing labels shape: (58,)

   The data manipulation steps outlined above effectively prepare the dataset for machine learning modeling by creating a robust feature matrix, normalizing features, encoding labels, and splitting the dataset into training and testing sets. This preparation is crucial for ensuring that the subsequent modeling process is accurate and efficient.

### 3.3 Data Evaluation

In this section, we evaluate and compare the performance of two classification models: Logistic Regression and Support Vector Machine (SVM). The models were applied to a specific dataset, and their performances were measured using key evaluation metrics, including accuracy, precision, recall, and F1 score.

1. **Model Training and Predictions:**

   - **Logistic Regression:** LR is a statistical method used for binary classification that models the probability of a categorical dependent variable based on one or more predictor variables using the logistic function. The model was trained using the training data, and predictions were made on the test set.
   - **Support Vector Machine (SVM):** SVM is a supervised machine learning algorithm used for classification and regression tasks that finds the optimal hyperplane to separate data points of different classes in high-dimensional space. This model was trained in a similar manner, using the training data and making predictions on the test set.

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| Logistic Regression | 0.4655 | 0.2790 | 0.4655 | 0.3209 |
| SVM | 0.4310 | 0.2179 | 0.4310 | 0.2744 |

Table 1: Comparison of Evaluation Metrics for Logistic Regression and SVM models.

2. **Confusion Matrix Visualization:** Confusion matrices were plotted for both models to provide a visual representation of their performance. The confusion matrices show the distribution of true positives, true negatives, false positives, and false negatives, which helps identify specific areas where each model may struggle. Both the confusion matrices can be observed through Figures 7 and 8.

3. **Performance Comparison:** A bar graph was created to compare the evaluation metrics of both models visually. The analysis revealed that - Logistic Regression outperformed SVM in all evaluation metrics, including accuracy,

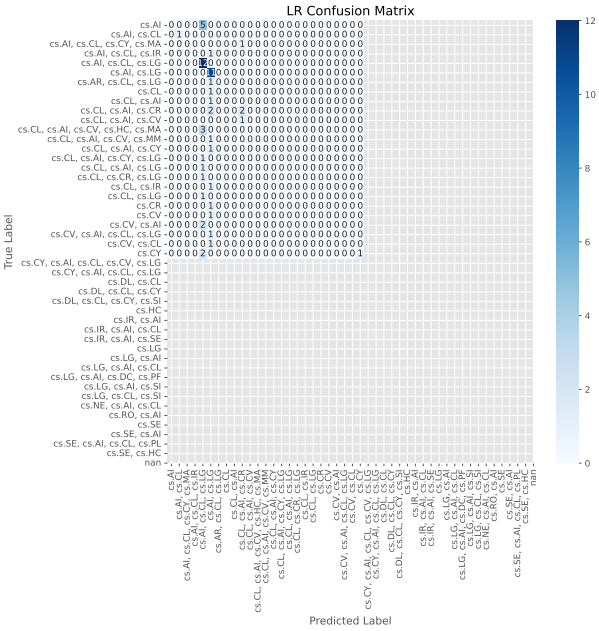


Figure 7: LR Confusion Matrix

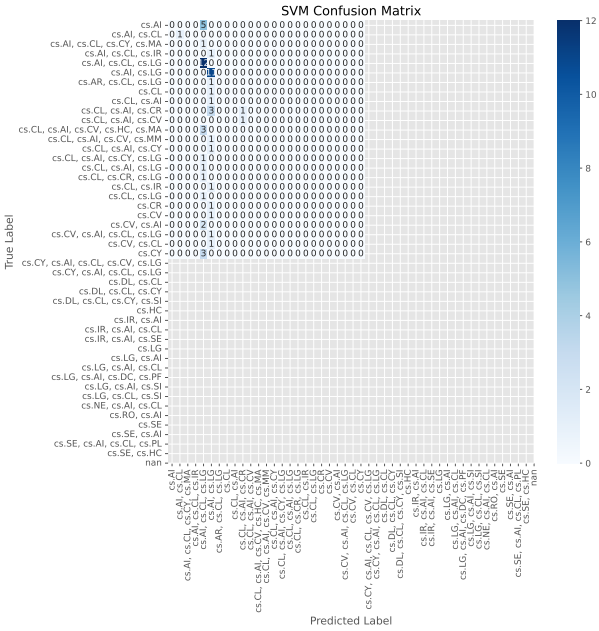


Figure 8: SVM Confusion Matrix

precision, recall, and F1 score which can be observed from the Figure 9.

4. **Overall Performance:** The average scores for each model were calculated to determine overall performance which showed that Logistic Regression's average ccore was 0.3828 and that of SVM's was 0.3740. The results indicate that Logistic Regression performed better overall, suggesting it may be a more suitable model for this dataset and classification task. This evaluation underscores the importance of using appropriate metrics to assess model performance and the potential for further improvements through hyperparameter tuning and exploring additional algorithms.

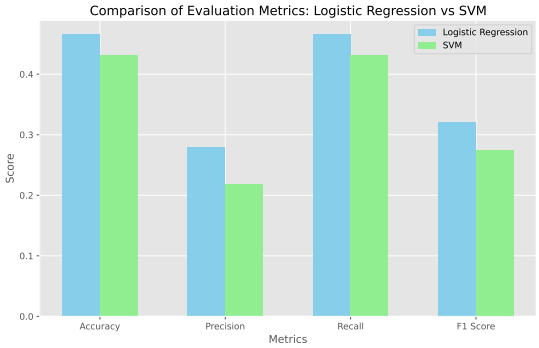


Figure 9: LR vs SVM Model Comparison

## 4 Conclusion

In this analysis, we thoroughly explored and manipulated the dataset to derive meaningful insights for classification tasks. We employed two machine learning models: Logistic Regression and Support Vector Machine (SVM), to evaluate their performance in predicting outcomes based on textual features extracted from the data.

The data pre-processing steps included transforming the text data into numerical representations using TF-IDF, effectively capturing the significance of terms within the documents. Our evaluation metrics — accuracy, precision, recall, and F1 score provided a comprehensive assessment of model performance. The results indicated that Logistic Regression outperformed SVM across all metrics, achieving an accuracy of 0.4655, precision of 0.2790, and recall of 0.4655, thereby highlighting its suitability for this particular classification problem.

The insights gained from this analysis underscore the importance of selecting the appropriate model based on performance metrics tailored to the specific dataset and task. Future work may focus on optimizing model parameters, experimenting with additional features, and exploring advanced algorithms to further enhance predictive accuracy. Overall, this exploration and evaluation process illustrates the critical role of data manipulation and modeling in deriving actionable insights from complex datasets.

# A APPENDIX

## References


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

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

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

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

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

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

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

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jun Zhuang and Mohammad Al Hasan. 2022. Defending graph convolutional networks against dynamic graph perturbations via bayesian self-supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4405–4413.

Jun Zhuang and Casey Kennington. 2024. Understanding survey paper taxonomy about large language models via graph representation learning. *arXiv preprint arXiv:2402.10409*.