Classifying Survey Papers on Large Language Models Using Machine Learning Techniques

Chiaying Wu
Department of Computer Science
Boise State University
chiayingwu@u.boisestate.edu

Abstract

With the rapid development of research on large language models (LLMs), the number of related survey papers is continuously increasing, posing challenges for scholars and researchers navigating this field. This study aims to apply machine learning techniques to effectively classify survey papers on LLMs, thereby providing researchers with better literature retrieval and analysis tools. I first constructed a diverse feature matrix by integrating text data and class labels from different datasets. Using preprocessing methods such as TF-IDF vectorization and one-hot encoding, I prepared for subsequent model training. In the experiments, I implemented a random forest classifier to analyze the relationship between features and classification labels. Preliminary results indicate that my classification model achieved an accuracy of 26% without parameter tuning, which improved to 31% after hyperparameter optimization. Despite challenges such as class imbalance and feature correlation, my research provides an effective method for the automatic classification of survey papers. Future work will focus on further improving classification accuracy and exploring other machine learning algorithms to expand the applicability of such tasks.

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 build a robust feature matrix from a diverse dataset, preprocess the data, and utilize a Random Forest classifier to analyze the relationships between various features and their corresponding labels. This involves vectorizing textual data using TF-IDF, applying one-hot encoding for categorical features, and normalizing the feature matrix to ensure uniformity in model training. We will evaluate the model's performance through various metrics to assess its effectiveness and identify potential areas for improvement.

Overall, our contributions can be summarized as follows:

- Construct a feature matrix that integrates both textual and categorical data, enhancing the model's input representation.
- Preprocess the data through normalization and encoding, preparing it for effective machine learning model training.
- Implement and evaluate a Random Forest classifier, providing insights into the classification performance and identifying challenges, such as class imbalance, that may affect the results.

2 Related Work

Data preprocessing is a crucial step in the machine learning pipeline, as it directly impacts the quality and effectiveness of the models. In this project, I leverage various techniques from the pandas (pandas) library to preprocess our dataset effectively.

The use of pandas for data manipulation and cleaning is well-documented in its official documentation.

In this study, I utilized key tools from `scikit-learn` (scikit-learn). for preprocessing, feature extraction, model training, and evaluation. The `TfidfVectorizer` from the `feature_extraction` module was used to convert paper titles and summaries into numerical TF-IDF vectors. I normalized the feature matrix using `MinMaxScaler` from the `preprocessing` module and split the dataset with `train_test_split` from the `model_selection` module.

I applied `RandomForestClassifier` from the `ensemble` module for classification and used `GridSearchCV` to fine-tune the model's hyperparameters. For evaluation, `accuracy_score`, `classification_report`, and `confusion_matrix` from the `metrics` module were employed to assess accuracy and classification performance. (scikit-learn).

In terms of data visualization, I utilized the Matplotlib (Matplotlib, 2021). library. Specifically, I used line plots and bar charts to illustrate the trends in survey data and the distribution of proposed taxonomies. The library's customizable features enhanced the clarity of the visual results.

Additionally, I used the NumPy (NumPy, 2024). and SciPy (SciPy, 2024). libraries for numerical computations and data processing. NumPy provides essential functionalities for handling arrays and performing mathematical operations, while the sparse module of SciPy allows me to efficiently manage and combine sparse matrices. These libraries facilitated the creation of a comprehensive feature matrix for my classification model. The detailed documentation of NumPy and SciPy offers valuable insights into their respective methods and applications, aiding me in effectively implementing my analysis.

3 Methodology

3.1 Data Exploration

The dataset used in this project was loaded using the pandas library, which enabled efficient manipulation of the data for further analysis. The dataset consisted of survey papers with relevant features such as release dates and taxonomy classifications.

First, I examined the temporal trends in the dataset. The "Release Date" field was converted into a datetime format, which allowed me to extract monthly and yearly trends. By grouping the data by month, I plotted the number of surveys released over time. The figure revealed distinct publishing trends across different months, highlighting periods of increased survey paper production, shown in figure 1.

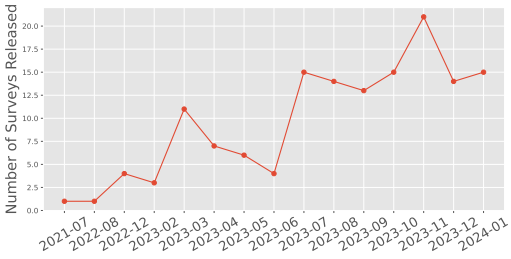

Figure 1: The number of survey papers published each month

Next, I explored the distribution of paper classifications, referred to as "Taxonomy." Using a bar chart, I visualized the number of papers belonging to each taxonomy category. This provided insight into which categories had the most significant focus within the dataset, shown in figure 2.

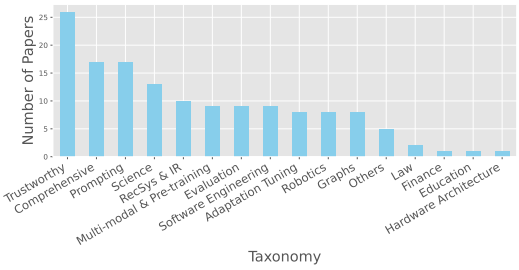

Figure 2: The distribution of the proposed taxonomy

In addition to these foundational analyses, further exploration involved grouping the data by both "Year" and "Taxonomy" to examine trends across different taxonomies over time. The results were visualized in a multi-line plot, revealing how the prominence of each taxonomy evolved across years, shown in figure 3.

Through these explorations, I gained a better understanding of the dataset's structure and trends, which laid the groundwork for subsequent modeling and analysis.

3.2 Data Manipulation

In this section, I focused on building and preprocessing a feature matrix from the dataset. The process started with text-based features, such as the "Title" and "Summary" columns, which were

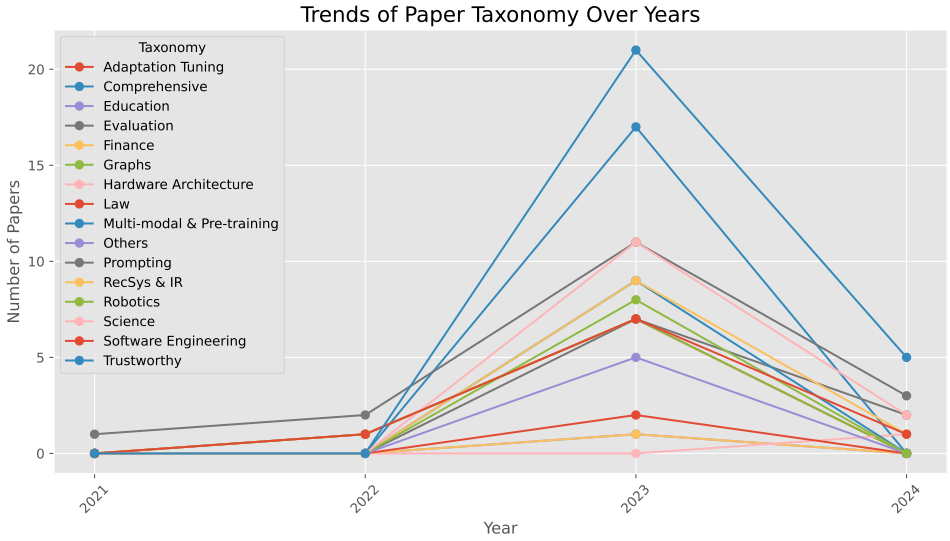

Figure 3: Trends of Paper Taxonomy Over Years

vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) method. This allowed me to represent the textual data as numerical values, capturing the importance of specific words across documents. The TF-IDF vectorizer was applied separately to both the "Title" and "Summary" columns, producing sparse matrices that mapped words to their corresponding TF-IDF scores.

Next, I applied one-hot encoding to the "Categories" column. Since each paper could belong to multiple categories, I split the categories into individual values and created binary indicators for each unique category. This transformation resulted in a matrix where each category was represented as a separate feature with a value of 1 or 0, indicating its presence or absence.

After obtaining these individual feature matrices, I concatenated them to form a comprehensive feature matrix that incorporated both the TF-IDF features and the one-hot encoded categories. This matrix represented all the necessary features for further analysis.

To preprocess the feature matrix, I normalized the data using the MinMaxScaler, ensuring that all features were scaled to a common range. This step was crucial for training machine learning models, as it prevented any feature from disproportionately influencing the model due to its scale. After that I encoded the "Taxonomy" labels using a label encoder, transforming the categorical labels into numerical values for classification tasks.

Finally, the dataset was split into training and testing sets using a 60/40 ratio, ensuring that the model could be evaluated on unseen data. This step prepared the dataset for model training and evaluation.

3.3 Data Evaluation

To evaluate the effectiveness of the feature matrix in classifying papers into their respective taxonomies, I employed a Random Forest Classifier as our machine learning model. In my first attempt, the model was trained using default hyperparameters, resulting in an accuracy of 0.26. This initial accuracy indicates that the model's performance was relatively low, suggesting that it struggled to effectively classify the test set.

To enhance the model's performance, I used hyperparameter tuning using Grid Search in my second attempt. This process involved defining a parameter grid that varied the number of estimators, maximum depth, minimum samples split, and minimum samples per leaf. After optimizing the model with these configurations, the accuracy improved to 0.31, marking a 5% increase from the first attempt, shown in table 1. The increase in accuracy can be attributed to the model's better tuning, which allowed it to learn from the training data more effectively.

Despite this improvement, the accuracy of 0.31 still reflects a relatively low performance. This indicates that there are still significant areas for enhancement in my model's ability to classify the papers correctly.

To assess the model's performance, I employed standard evaluation metrics, including precision,

recall, and F1-score. Additionally, I generated a classification report to provide detailed insights into the model's performance across various categories. A confusion matrix was also created to visualize the distribution of correct and incorrect predictions, helping to identify specific categories with higher misclassification rates.

Future work will focus on further refining the model, potentially exploring different algorithms, additional feature engineering, and employing more advanced techniques to improve classification accuracy.

Attempt	Accuracy
First attempt	0.26
Second attempt	0.31

Table 1: Model Accuracy Comparison

3.4 Additional Contributions of Sentiment Analysis

In this study, I further assessed sentiment trends to complement the analysis of the paper dataset. To achieve this, I utilized the text classification pipeline from Hugging Face's Transformers library (HuggingFace, 2024) to perform sentiment classification on the titles of the papers, shown in figure 4. By analyzing each title, I was able to quickly capture the overall sentiment trends in the research field, providing valuable insights for subsequent research. The classification results were organized into a dataframe, and a pie chart was created to display the proportions of positive and negative labels. This approach not only improved the efficiency of the analysis but also added more dimensions to the research.

4 Conclusion

In this report, I described the process of applying machine learning techniques to classify survey papers, emphasizing the importance of data preprocessing, feature engineering, and model evaluation. By constructing a comprehensive feature matrix that integrates text and categorical data, I laid the foundation for effective model training. Preliminary results showed that the accuracy of the random forest classifier reached 26%, highlighting the need for further improvement.

By implementing hyperparameter tuning, I improved the model's accuracy to 31%. While this is a step forward, it still underscores the challenges

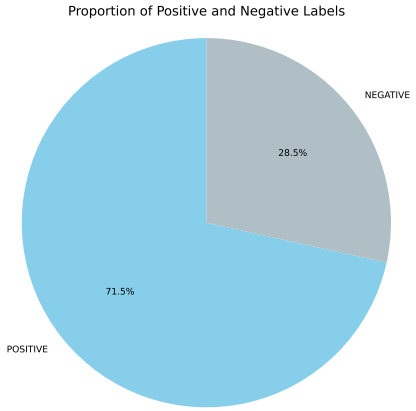

Figure 4: Sentiment Distribution of Paper Titles

of class imbalance and feature relevance. My evaluation metrics, including precision, recall, and F1 score, provided valuable insights into the model's performance across different classification categories.

Future work will focus on addressing the identified challenges, including updating datasets, evaluating new methods (such as prompting techniques in large language models), and solving class imbalance problems. By exploring other machine learning algorithms and incorporating more complex preprocessing methods, I hope to further enhance the model's effectiveness and reliability for better performance in classifying survey papers. Ultimately, this research contributes to the growing field of data science, providing insights and methods applicable to similar classification tasks.

A APPENDIX

A.1 Experimental Environment Setup

- Operating System: Windows 11 Home, 64-bit, x64-based processor
- Processor: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz
- Memory: 16.0 GB (15.7 GB usable)
- Python Version: 3.12.5
- Library Versions Used:
  - pandas 2.2.2
  - scikit-learn 1.5.2
  - numpy 2.1.1

- matplotlib 3.5.3
- SciPy 1.14.1

A.2 Hyperparameter Settings

The following hyperparameter settings were used during model training and tuning:

- Random Forest Classifier Hyperparameters:
  - `n_estimators`: 100, 200
  - `max_depth`: None, 10, 20
  - `min_samples_split`: 2, 5
  - `min_samples_leaf`: 1, 2
- Cross-Validation: 5-fold cross-validation was used for model tuning.

A.3 Feature Matrix Construction and Preprocessing

The following preprocessing steps were performed during feature matrix construction:

- TF-IDF was applied to vectorize the "Title" and "Summary" fields, using up to 1000 feature words per document.
- One-hot encoding was applied to the "Categories" field and integrated into the feature matrix.
- The feature matrix was normalized using `MinMaxScaler`, mapping feature values to the [0, 1] range.

A.4 Dataset Splitting

The dataset was split as follows:

- Training Set Proportion: 60%
- Test Set Proportion: 40%
- Random Seed: `random_state=None`

A.5 Model Evaluation

The following metrics were used to evaluate the model's performance:

- Accuracy
- Precision, Recall, and F1-score
- Confusion Matrix

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.

HuggingFace. 2024. Transformers.

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

Matplotlib. 2021. Matplotlib documentation.

NumPy. 2024. Numpy v1.24.0 documentation.

pandas. *pandas Documentation*.

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.

scikit-learn. scikit-learn: Machine learning in python - api documentation.

SciPy. 2024. Scipy v1.10.0 documentation.

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*.