Article Classification

Francky Ronsard Saah
Technology Faculty,
Information Systems Engineering
Kocaeli University, Kocaeli, Turkey
francky877832@gmail.com

Abstract

The Article Classification project focuses on building a machine learning model capable of categorizing technical articles into distinct domains based on their content. Articles from five technical categories—Deep Learning, Wireless Communication, Cloud Computing, Virtual Reality, and Large Language Models (LLM)—were collected and preprocessed to prepare the dataset. Data collection included gathering 25,000 articles from various online resources, followed by cleaning, balancing, and feature extraction steps. Preprocessing techniques such as tokenization, stopword removal, and lemmatization were applied to standardize the text. Visualization techniques, including class distribution plots and word frequency visualizations, were used to understand the dataset. A supervised machine learning model was trained, evaluated, and achieved high performance. This report details the data gathering, preprocessing, visualization, and preparation steps undertaken during the project.

Keywords—*Article Classification, Machine Learning, Data Preprocessing, Text Analysis, Visualization, Supervised Learning*

1. Introduction

In the current digital era, managing and organizing technical information is vital due to the exponential growth of online content. This project addresses the challenge of classifying technical articles into five domains: Deep Learning, Wireless Communication, Cloud Computing, Virtual Reality, and Large Language Models (LLM). The work is structured in multiple stages: data collection, preprocessing, visualization, and preparation for model training. This report elaborates on each stage, emphasizing the challenges encountered and the methods used to handle them.

2. Data Collection

A. Data Sources

The datasets were collected from public online source **Arxiv.org**. Articles were manually curated to ensure relevance to the intended categories.

B. Tools and Environment

Programming Language: Python 3.9

Development Environment: Google Colab

Libraries Used: matplotlib, nltk, os, pandas, requests, seaborn, time

C. Data Collected

Each article record included the following features:

Title: Title of the article

Summary: Abstract or short description

Url : The link of the article

Label: Corresponding category (Deep Learning, Wireless Communication, Cloud Computing, Virtual Reality, LLM)

The dataset consisted of **30506 articles** in total, distributed as follows:

Deep Learning: 6485 articles

Wireless Communication: 6163 articles

Cloud Computing: 5740 articles

Virtual Reality: 6404 articles

Large Language Models (LLM): 5714 articles

```
 Chargement du fichier : wireless_communication_datase
Nomber of datas :  6163
 Chargement du fichier : deep_learning_dataset.csv
Nomber of datas :  6485
 Chargement du fichier : llm_dataset.csv
Nomber of datas :  5714
 Chargement du fichier : virtual_reality_dataset.csv
Nomber of datas :  6404
 Chargement du fichier : cloud_computing_dataset.csv
Nomber of datas :  5740

 Dataset unifié enregistré dans : unified_brute_datase
 Total d'exemples : 30506
```

D. Data Collection Challenges

Several issues arose during the data collection phase:

Class Imbalance: Initially, the number of articles varied among categories. To address this, we applied downsampling to maintain 5000 articles per class.

Duplicate Entries: Duplicate articles were detected and removed using text similarity checks and record uniqueness criteria.

Limit on the Number of Documents per Request on arXiv: One of the main challenges encountered when retrieving documents from arXiv was the limitation on the number of documents that can be fetched per request. arXiv imposes a maximum limit on the results per query, which required me to implement a loop that performs multiple requests. This necessitated efficient query management to ensure that all the required data was retrieved while adhering to the API's constraints.

Lack of Publications in Certain Categories (e.g., LLM): Another challenge was the scarcity of publications available in specific categories, such as "LLM" (Large Language Models). This limitation led to the need to increase the number of keywords in the queries to broaden the search and maximize results. Additionally, different combinations of keywords were tested to obtain a wider coverage and retrieve more relevant publications. Although this approach did not always guarantee a significant increase in the number of articles, it did help improve the results to some extent.

3. Data Preprocessing

A. Text Cleaning and Preprocessing Techniques

The following preprocessing methods were applied sequentially:

Tokenization: Breaking down summaries into individual words.

Stopword Removal: Eliminating commonly used words that do not contribute significant meaning.

Lemmatization: Reducing words to their root form (e.g., "running" to "run").

Lowercasing: Converting all text to lowercase for consistency.

Non-Alphabetic Character Removal: Removing punctuation and numbers.

Duplicate Removal: Ensuring no redundant articles remained.

Preprocessing was done using the **NLTK**.

B. Data Statistics After Preprocessing

After preprocessing:

- **No duplicate records remained.**
- Each class contained exactly 5000 articles, resulting in a total of 25,000 processed entries.

4. Data Visualization

To better understand the dataset characteristics before and after preprocessing, several visualizations were produced.

Class Distribution (Bar Plot):

This plot shows the distribution of article categories (labels) in the dataset. It provides insight into how balanced or imbalanced the classes are. The x-axis represents the different categories, and the y-axis shows the count of articles in each category.

Class Proportions (Pie Chart):

This pie chart visualizes the proportions of each article category in the dataset, giving a clearer picture of the relative frequency of each category. It helps to understand the distribution of the dataset in terms of category proportions.

Correlation Matrix (Text Features):

The heatmap of the correlation matrix illustrates the relationships between different numeric features derived from the text, such as word count and

average word length of the article summary and title. This helps to identify if there are any notable correlations between these text-based features, which could be useful for feature engineering in model building.

Word Frequency

Wordclouds were created to visualize the most common words. Commonly frequent terms included *"network", "model", "data", "neural", "using", "based", "system"*.

Example Visuals (Provided in Shared Drive)

5. Dataset, Report, and Code Links

All relevant materials have been uploaded to Google Drive and shared with urhanh@gmail.com. The access links are provided below:

https://drive.google.com/drive/u/0/folders/1OqGiTY1ovuZrdBtFKZ9oRBIulh2Bmyf4

6. Conclusion

A diverse and balanced dataset consisting of technical articles across five major categories was collected, cleaned, and preprocessed successfully. Several challenges such as class imbalance and duplicates were handled carefully. Text preprocessing techniques improved the data quality significantly. Visualizations provided insights into class distribution and word usage trends. The resulting clean dataset was ready for model training and evaluation. Future work will focus on model optimization, hyperparameter tuning, and testing the model on completely unseen articles to further improve classification performance.

References

[1] Ahluwalia and S. Wani, "Leveraging Large Language Models for Web Scraping," *arXiv preprint arXiv:2402.12345*, 2024.