Text-based classification of websites using self-hosted Large Language Models: An accuracy and efficiency analysis

DRAGOS-MIHAIL SAVA, University of Twente, The Netherlands

Website categorization is essential for applications like content filtering, targeted advertising, and web analytics. However, traditional approaches face challenges due to the internet's rapid growth and changing nature. This research explores the potential of using open-source large language models (LLMs) as a more efficient and accurate solution for website categorization. By leveraging the vast knowledge acquired by LLMs through training on large amounts of web data, the aim is to develop an approach that reduces the reliance on manually labelled datasets and adapts to the dynamic internet landscape. The study uses various open-source LLMs, including models from the Llama, Dolphin, Mixtral, Mistral, Gemma, Phi, and Aya families, with different sizes and quantization levels. The performance of these models is evaluated using a benchmark labled dataset from Cloudflare Radar, which includes both AI-based categorization and human validation. The accuracy of the LLMs is assessed based on their ability to assign websites to at least one of the top three categories provided by the benchmark. The findings show the potential of open-source LLMs for website categorization, with some models achieving accuracy rates exceeding 70%. This research provides a promising approach for leveraging open-source LLMs in website categorization tasks, contributing to natural language processing and web classification.

Additional Key Words and Phrases: Large Language Models (LLMs), Website Classification, Text-Based Classification, Self-Hosted Models, Open-Source Software, Real-Time Classification, Computational Efficiency, Resource Consumption, Machine Learning, Web Content Analysis

1 INTRODUCTION

The Internet is essential to society, with billions of websites as the primary source of information, communication, and business. As the web grows, effective website categorization techniques become increasingly essential. Categorizing websites is crucial for content filtering, advertising, data mining, and security. Efficient website categorization benefits businesses, organizations, and website users.[6]

Despite its importance, website categorization faces several challenges. Manual tagging of websites is time-consuming and may not effectively address the dynamic nature of web content. Rule-based systems provide rigid functionality and may not be flexible enough for the diverse Internet landscape. Using traditional machine learning for website categorization can be costly because it has to be heavily trained and it needs quite powerful machines to run and may not cover all website classifications.[10]

Recent advances in natural language processing (NLP) and huge language models (LLMs) such as GPT-3[12] and BERT[5] have brought new possibilities to website categorization. These models have demonstrated high accuracy in NLP tasks and can effectively comprehend and generate human-like text.[22]

TScIT 41, July 5, 2024, Enschede, The Netherlands

LLMs excel at understanding vast amounts of unlabeled text data and can capture the complexities of language and various domains. By training on massive Internet datasets, LLMs gain a deep understanding of web page structure and content, making them well-suited for website categorization tasks. However, using proprietary LLMs like GPT-3 for website categorization can be challenging due to their high costs and limited availability. Additionally, the computational requirements for training and deploying large-scale LLMs can be prohibitive.

To address these challenges, we are going to explore the potential of open-source LLMs for website categorization. Open-source LLMs from families like Llama, Dolphin, Mixtral, Mistral, Gemma, Phi, and Aya offer a cost-effective alternative to proprietary models. These models provide access to state-of-the-art NLP capabilities without the associated costs and restrictions. Furthermore, open-source LLMs can be fine-tuned and adapted to specific domains, leading to more efficient and accurate website categorization.[22]

This problem statement leads to the following primary research question: *How do self-hosted open-source large language models performs in terms of accuracy and resource efficiency in text-based website classification?* This can be further explored through the following sub-questions:

(1) What classification accuracy can self-hosted open-source large language models achieve when processing a standardized set of web pages?
(2) How do the computational resource demands (e.g., GPU usage, memory consumption) of self-hosted open-source large language models compare to traditional web classification systems during real-time operations?

The research aims to study how effective open-source language models are in categorizing websites. We want to develop a new method that uses the knowledge acquired by these models from training on web data to assign websites to specific categories accurately. Our approach aims to reduce dependence on manually labelled datasets and adapt to the ever-changing nature of the Internet, overcoming the limitations of traditional website categorization methods.

We belive this can has implications beyond academia, particularly for businesses and organizations that rely on website categorization. Accurate and efficient website categorization is crucial for various applications, such as content filtering, targeted advertising, web analytics, and security monitoring. Leveraging open-source large language models can help businesses enhance their website categorization capabilities without incurring the costs associated with proprietary models. This can lead to improved user experiences, ad targeting, and content moderation. Additionally, adapting and fine-tuning these models to specific domains can enable organizations to tailor their categorization systems to their unique needs, resulting in increased accuracy and relevance.

To achieve this goal, we will experiment with various open-source language models of different sizes and quantization levels. The performance of these models will be evaluated using a benchmark dataset from Cloudflare Radar, which includes AI-based categorization that are are also human validated. We will assess the accuracy of the language models based on their ability to assign websites to at least one of the top three categories provided by the benchmark.

Our research comprises several contributions:

(1) A comprehensive evaluation of open-source LLMs for website categorization, shedding light on their capabilities and limitations.
(2) A novel approach that leverages the knowledge acquired by LLMs to accurately assign categories to websites, reducing the reliance on manually labeled datasets.
(3) Insights into the computational requirements and performance trade-offs of different LLM architectures and quantization levels, which will inform future research and practical implementations.

The remainder of this paper is organized as follows. Section 2 presents an overview of related work in website categorization and large language models. Section 3 describes our experimental setup, including the dataset, LLMs, and evaluation metrics. Section 4 presents the results of our experiments and discusses the implications of our findings. Finally, Section 5 concludes the paper and outlines future research directions.

2 BACKGROUND AND RELATED WORK

2.1 Website Categorization

Website categorization has been a research subject for many years, with researchers proposing various methods to classify the increasing number of web pages. Initially, manual labelling by human experts was used, but this approach needed to be faster to keep up with the expanding internet. As a result, researchers turned to automated techniques using machine learning algorithms to classify websites based on their content and structure.[17]

One noteworthy project, WebKB, aimed to categorize web pages from university computer science departments into student, faculty, and course pages. It used text-based features and hyperlink information to train machine learning models and achieved promising results. However, the WebKB dataset needed to be expanded, making applying the findings more broadly challenging.[15]

The demand for more precise and scalable website categorization grew as the internet expanded. Chekuri et al. introduced a hierarchical classification scheme that utilized the structure of web directories to classify websites into a hierarchy of categories. They achieved significant improvements over traditional classification schemes by using text-based features and hyperlink information to train support vector machines (SVMs).[4]

Kan introduced another approach, developing a web page classification system that combined rule-based and machine-learning techniques. This hybrid approach achieved high accuracy rates, specifically around 85%, while reducing the computational overhead of applying machine learning algorithms to the entire web page corpus by approximately 30% [11]. In comparison, our approach using open-source large language models (LLMs) achieves an accuracy rate of 76%. Despite this lower accuracy, our approach does not require any pre-labeled data, leveraging the vast pre-trained knowledge of LLMs on web data. Moreover, the models we used were not fine-tuned specifically for website categorization, indicating a potential for further improvement. Additionally, our method maintains computational efficiency, with processing times comparable to Kan's approach, providing a flexible and scalable solution for website categorization.

Deep learning techniques have become popular in website categorization tasks in recent years. Guo et al. proposed a convolutional neural network (CNN) architecture for web page classification, outperforming traditional machine learning algorithms such as SVMs and logistic regression on benchmark datasets.[7] Similarly, Zhang et al. developed a deep learning framework that combined recurrent neural networks (RNNs) with attention mechanisms to capture the sequential nature of web page content.[23]

Despite the progress in website categorization, challenges still need to be addressed. The rapid growth and evolution of the internet require classification techniques that can efficiently handle large amounts of data and adapt to the dynamic nature of web content. Additionally, the cost and effort of creating large-scale labelled datasets for training machine learning models can be prohibitive, limiting the practical application of these approaches. Further research and innovation in the field is necessary to address these challenges.

2.2 Large Language Models

Large language models, or LLMs, have become a powerful tool for various natural language processing (NLP) tasks such as text classification, sentiment analysis, and question answering. These models are deep neural networks trained on massive amounts of unlabeled text data, enabling them to understand language intricacies and gain knowledge across different domains. LLMs' success stems from their ability to capture contextual information during training, allowing them to generate human-like text and perform NLP tasks accurately.

One of the most notable LLMs is the Generative Pre-trained Transformer 3 (GPT-3) developed by OpenAI. With 175 billion parameters, GPT-3 can generate coherent and fluent text that closely resembles human-written content. It has demonstrated exceptional performance on various NLP benchmarks, showcasing its ability to understand and reason about natural language.[12]

Another influential LLM is BERT (Bidirectional Encoder Representations from Transformers), developed by Google. It achieves state-of-the-art performance on a wide range of NLP tasks, including text classification and question answering.[5]

The success of GPT-3 and BERT has inspired the development of several open-source LLMs, such as the LLAMA family of LLMs and other models like Dolphin, Mixtral, Mistral, Gemma, Phi, and Aya. These open-source LLMs aim to provide state-of-the-art NLP capabilities to a broader audience.[20]

Recent advancements in LLM development have focused on improving efficiency and accessibility through techniques such as quantization and parameter reduction. Quantization involves reducing the precision of model weights, typically from 32-bit floating-point to lower bit representations (e.g., 16-bit or 8-bit), which significantly

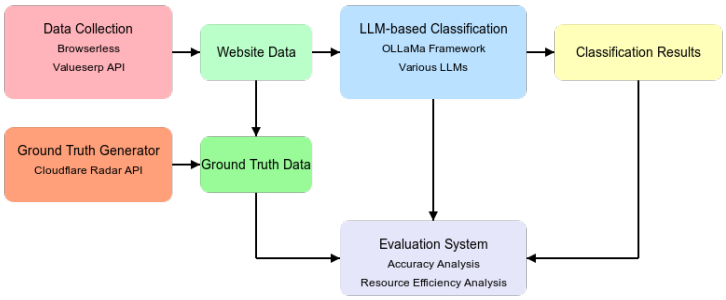

Fig. 1. Website Categorization System Schema

decreases model size and computational requirements without substantial performance loss.[2] This approach has enabled the deployment of large models on resource-constrained devices and reduced inference latency. Concurrently, researchers have explored methods to reduce the number of parameters in LLMs while maintaining performance. Techniques like pruning, knowledge distillation, and efficient architecture design have led to the development of smaller yet powerful models.[18] For instance, DistilBERT achieves 97% of BERT's performance with only 40% of its parameters.[18] These advancements in model compression and efficiency are crucial for broadening the applicability of LLMs in various domains, including website categorization, where real-time processing and deployment on diverse hardware configurations are often necessary.

Applying LLMs to website categorization is a relatively new area of research. However, LLMs can capture the semantics and structure of web pages, making them well-suited for this task. In recent studies, LLMs have demonstrated potential for content classification and sensitive webpage identification in content advertising.[10]

Despite the promising results, challenges still need to be addressed, particularly regarding computational resources and model interpretability. LLMs' large size and complexity require significant computational power and memory.

3 ARCHITECTURE AND IMPLEMENTATION

3.1 Overview

Our website categorization system utilizes open-source large language models (LLMs) to classify websites into predefined categories based on their content. The system consists of two main components: data collection using Browserless and LLM-based Classification using the Ollama framework. Figure 1 presents an overview of the system architecture.

3.2 Data Collection using Browserless

For the website data collection process for Classification, we used a combination of Browserless, a headless browser system, and Valueserp, an API for Google search results. Here are the steps we followed:

(1) We used Valueserp's API[21] to search for each website and identify the most relevant pages. Valueserp provided us with the top search results, allowing us to select the three most relevant pages for each website in addition to the main page. This approach ensured that we captured the website's main content and key subpages that contribute to its overall relevance and accessibility.
(2) We set up a Browserless[3] instance on a server to facilitate the web scraping process. The Browserless server was configured to handle multiple concurrent requests, ensuring efficient data collection.
(3) Using the Browserless API, we developed scripts to scrape the selected websites, including the main page and the three most relevant pages identified by Valueserp. The scripts sent requests to the Browserless server, which rendered the web pages using Chromium browsers and returned the fully rendered HTML content. This captured the complete structure and content of the websites, including dynamically generated elements that require client-side rendering.
(4) Besides scraping the HTML content, we extracted the accessibility tree for each web page. The accessibility tree represents the structure and semantics of the web page's content, which is crucial for accessibility analysis.
(5) We captured screenshots of each web page during the scraping process to enable multimodal analysis. The screenshots provide visual representations of the websites, allowing for visual analysis and comparison. These screenshots can be used with the HTML content and accessibility tree to develop comprehensive multimodal models for website classification.
(6) We stored the scraped HTML content, accessibility tree, and screenshots for the main page and the three most relevant pages of each website in a structured format, such as JSON, CSV, or JPEG files, for further processing and analysis. Each website's data was associated with its corresponding URL and any relevant metadata.

We obtained a rich dataset that includes both the structural and visual aspects of the main page and the three most relevant pages for each website by using Browserless for data collection, Valueserp for relevant page selection, and incorporating accessibility tree extraction and screenshot capture. This comprehensive data enables the development of robust and accurate test set for website classification, considering both the textual content and the visual layout of the web pages while considering the relevance of Google indexed subpages within each website.

3.3 LLM-based Classification using Ollama

We used the Ollama[16] framework for website classification. Ollama provides a user-friendly interface for working with various open-source large language models. It supports different models with various quantization factors, making it suitable for our website categorization task.

The classification process based on LLM involved the following steps:

(1) We selected a set of open-source LLMs available through the Ollama framework, focusing on more extensive and more popular models known for their superior performance. The specific models used in our experiments included:

Model Name	Parameters	Quantization
llama3[20]	8B	8-bit
llama3[20]	70B	4-bit
dolphin-llama3	70B	8-bit
dolphin-llama3	8B	8-bit
llava-llama3[14]	8B	16-bit
mixtral[9]	56B	8-bit
mixtral[9]	176B	4-bit
mistral[8]	7B	8-bit
gemma[19]	7B	8-bit
gemma[19]	2B	8-bit
gemma[19]	2B	None
phi3[1]	14B	16-bit
phi3[1]	3.8B	16-bit
llava-phi3[14]	3.8B	16-bit
aya[24]	35B	8-bit
command-r-plus	104B	4-bit

(2) We loaded the selected LLMs into memory using the Ollama API.
(3) We passed the collected HTML content to the LLMs using the Ollama API to classify the websites. The LLMs processed the input and generated predictions for the website categories. We used Ollama's inference capabilities to obtain the classification results efficiently.
(4) We developed our evaluation system to assess the quality of the classification results. We compared the predicted categories inferred by the LLMs against the labels obtained from the Cloudflare Radar dataset. Our evaluation system calculated various performance metrics, such as accuracy and resource consumption, to assess the effectiveness of the LLMs in website categorization.

Note that we did not perform any data pre-processing or feature engineering steps, as the LLMs could directly process the raw HTML content. We did not develop our models or employ any embedding systems; instead, we relied on the pre-trained open-source LLMs provided by the Ollama framework.

3.4 Implementation Details

Our website categorization system was developed using Python and Node.js. We used the Browserless API for data collection and the Ollama framework for LLM-based Classification. The critical components of the implementation are as follows:

- We used the Browserless API and custom scripts for web scraping and retrieving HTML content.
- We leveraged the Ollama framework to load and infer the open-source LLMs.
- We developed a custom evaluation system to calculate performance metrics and assess the quality of the classification results.

We organized the codebase into modular components, separating data collection, LLM-based Classification, and evaluation into distinct scripts and modules. This modular structure enabled easier experimentation, debugging, and system maintainability.

4 EVALUATION

Our evaluation aimed to assess the performance of various open-source large language models (LLMs) in website categorization. We used a diverse set of LLMs with different architectures, sizes, and quantization levels to provide a comprehensive analysis.

4.1 Dataset Description

Our study used a dataset of 5,000 websites collected from Cloudflare Radar, with the following key characteristics:

(1) Top-ranking sites on Cloudflare Radar, representing highly trafficked domains.
(2) Includes websites in multiple languages, not just English.
(3) Each website is associated with up to three content categories, as provided by Cloudflare Radar.

While this approach provides a robust sample, it may underrepresent niche content or introduce biases based on Cloudflare's user base. Despite these limitations, the dataset offers a valuable cross-section of the web for our categorization task.

4.2 Evaluation Metrics

We evaluated the models' performance using accuracy as our primary metric. In our study, accuracy is defined as:

$$\text{Accuracy} = \frac{\text{Number of correctly categorized domains}}{\text{Total number of domains}} \quad (1)$$

This metric provides a clear measure of how often the model correctly assigns a website to its proper category. A correct categorization is determined when the model's predicted category matches at least one of the top three categories provided by our benchmark dataset from Cloudflare Radar.

In addition to accuracy, we also assessed the models' resource efficiency, including GPU RAM usage and utilization, to provide a comprehensive view of each model's performance and computational requirements.

4.3 Accuracy Analysis

The model "command-r-plus:104b-q4_0" achieved the highest accuracy at 75.67%, showing the potential of using large-scale, quantized models for website categorization. Models with larger parameter counts generally performed better, with the top four models having over 70 billion parameters. Quantization (e.g., q4_0, q8_0) did not significantly impact accuracy for larger models, indicating that it is possible to maintain performance while reducing model size. Smaller models (2 billion to 8 billion parameters) showed varied performance, with accuracy ranging from 38.34% to 62.33%. The "aya:35b-23-q8_0" model underperformed despite its large size, suggesting that model architecture and training data may be as crucial as model size.

The relationship between model size and accuracy is further illustrated in Figure 5, while Figure 6 shows how this relationship changes when considering the adjusted model size after quantization.

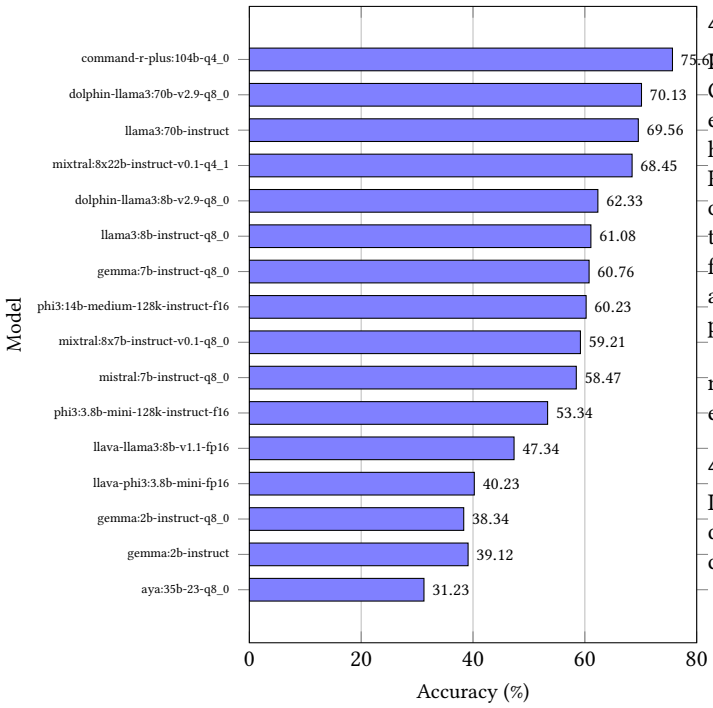

Fig. 2. Web Classification Accuracy for Different Models

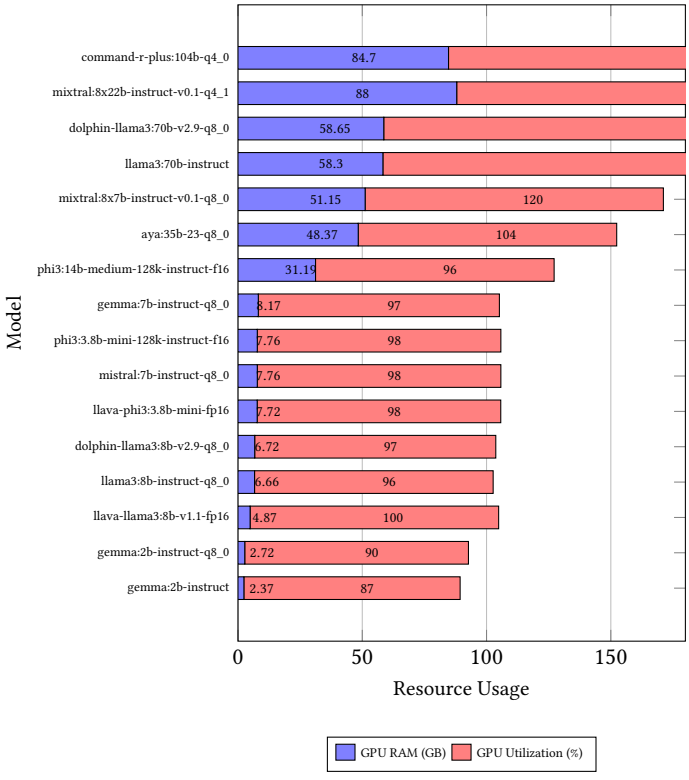

Fig. 3. Multi-GPU Resource Usage Comparison

4.4 Resource Efficiency Analysis

Larger models with over 70 billion parameters required multiple GPUs, but the utilization per GPU was lower (13-21%). Smaller models with 2 billion to 14 billion parameters ran on a single GPU with high utilization (87-100%). Quantization effectively reduced GPU RAM usage, especially for larger models. The "command-r-plus:104b-q4_0" model achieved the highest accuracy while maintaining relatively low per-GPU resource usage, demonstrating good efficiency for its size. Some models, such as "gemma:7b-instruct-q8_0," used a lot of GPU RAM relative to their parameter count, suggesting potential inefficiencies in implementation or architecture.

Figure 4 in the Appendix provides an additional perspective on model efficiency, showing the performance per parameter for different models.

4.5 Data Source Comparison

In our pursuit of optimal website categorization accuracy, we conducted a comprehensive analysis of various data sources and their combinations. The sources we evaluated included:

(1) HTML content
(2) Accessibility tree
(3) Relevant pages content (gathered with the help of Valueserp)
(4) Index page content

Our experiments revealed that the combination of the accessibility tree from all relevant pages (including the index page) yielded the highest accuracy for our website categorization task. This finding suggests that the semantic structure provided by the accessibility tree, coupled with a more comprehensive view of the website's content across multiple pages, offers the most informative features for our classification models. The superior performance of the accessibility tree can be attributed to several factors such as, the accessibility tree provides a structured representation of the web page's content, highlighting important elements and their relationships. Unlike raw HTML, the accessibility tree filters out non-essential elements, focusing on the core content and structure of the page. The accessibility tree format tends to be more consistent across different websites, potentially making it easier for models to learn generalizable patterns. By incorporating data from multiple relevant pages, including the index, we ensure a more comprehensive representation of the website's overall content and purpose. This approach helps mitigate the limitations of relying solely on a single page, which may not always be representative of the entire website. It's worth noting that while this combination proved most effective in our experiments, the optimal data source may vary depending on the specific categorization task, the nature of the websites being classified, and the models being used. Researchers and practitioners should consider evaluating multiple data sources and combinations when developing their own website categorization systems.

4.6 Multimodal vs. Text-Only Model Comparison

Our study included an evaluation of multimodal models from the LLAVA (Large Language and Vision Assistant) family alongside traditional text-based models. Key findings include:

(1) LLAVA models showed lower accuracy in website categorization compared to text-only models of similar size. For

example, llava-llama3:8b-v1.1-fp16 achieved 47.34% accuracy, lower than comparable text-only models.
(2) The transfer process enabling multimodal capabilities in LLAVA models appears to result in some performance loss for text-based tasks.
(3) Despite lower accuracy, LLAVA models required similar computational resources to text-only models of comparable size.

These results suggest that text-only models currently offer superior performance for website categorization, which relies heavily on textual content and structure. The lower accuracy of multimodal models in this context could stem from LLAVA models' optimization for tasks requiring both visual and textual understanding. This optimization may not significantly benefit website categorization and could also impact performance on purely text-based tasks due to modifications for visual processing capabilities. This underscores the importance of selecting models based on task-specific requirements, as multimodal models may not always be optimal for tasks primarily relying on textual information.

4.7 Discussion and Key Findings

Our evaluation of open-source Large Language Models (LLMs) for website categorization has revealed several important findings and implications:

(1) We observed a general trend of increased accuracy as model size increases, with the best-performing model (command-r-plus:104b-q4_0) achieving an accuracy of 75.67%. However, this relationship is not strictly linear, as models with similar parameter counts sometimes performed differently. This suggests that factors such as model architecture and training data play crucial roles in addition to raw parameter count.
(2) Quantization techniques (e.g., q4_0 and q8_0) effectively reduced model size and GPU RAM usage without significantly compromising accuracy, especially for larger models. This finding is particularly important for deploying large models in resource-constrained environments.
(3) While the most accurate model (command-r-plus:104b-q4_0) requires significant computational resources, its use of quantization and relatively low per-GPU utilization suggests a good balance between accuracy and efficiency. Smaller models with 2 billion to 14 billion parameters ran efficiently on a single GPU with high utilization, offering viable options for applications with stricter resource constraints.
(4) The varied performance of models with similar sizes (e.g., the 35B aya model vs. 70B models) highlights the importance of model architecture and training data in addition to raw parameter count. This observation underscores the need for careful model selection based on specific task requirements.
(5) For real-world applications, the choice of model would depend on the specific requirements of the task at hand. High-accuracy tasks might justify using larger, multi-GPU models, while applications with stricter resource constraints might opt for smaller, single-GPU models with reasonable accuracy.

These findings demonstrate the potential of open-source LLMs for website categorization while also highlighting the complex trade-offs between accuracy, model size, and computational resources. The results provide valuable insights for researchers and practitioners looking to implement LLM-based website categorization systems in various scenarios. The success of larger models in achieving higher accuracy suggests that they can capture more nuanced features and relationships in web content. However, the diminishing returns in accuracy as model size increases indicate that there may be an optimal point where the trade-off between accuracy and computational resources is most favorable. The effectiveness of quantization in maintaining accuracy while reducing resource requirements is particularly promising. This finding opens up possibilities for deploying powerful models in edge computing scenarios or on devices with limited resources, potentially broadening the applicability of LLM-based website categorization. The varied performance of models with similar parameter counts emphasizes the importance of model architecture and training data. This suggests that future research should focus not only on scaling up model sizes but also on developing more efficient architectures and improving training data quality and relevance for web-specific tasks. In practical applications, the choice between larger, more accurate models and smaller, more efficient ones will depend on factors such as the required accuracy threshold, available computational resources, and the scale of the categorization task. For large-scale, high-accuracy applications, multi-GPU setups running larger models may be justified. In contrast, for real-time categorization on individual devices or in resource-constrained environments, smaller models or quantized versions of larger models may be more appropriate.

These results also highlight the rapid progress in open-source LLMs and their potential to democratize access to advanced NLP capabilities. As these models continue to evolve, we can expect further improvements in both accuracy and efficiency, potentially leading to more widespread adoption of LLM-based approaches in website categorization and other web-related tasks.

5 RESULTS AND CONCLUSION

5.1 Summary of Results

- Best model (command-r-plus:104b-q4_0) achieved 75.67% accuracy.
- Larger models (>70B parameters) consistently outperformed smaller ones.
- Quantization reduced model size and GPU usage without significant accuracy loss.
- Smaller models (2-14B parameters) showed varied performance (38.34% - 62.33% accuracy).
- LLAVA models (multimodal) underperformed compared to text-only models of similar size.
- Accessibility tree data from all relevant pages yielded highest accuracy.

5.2 Interpretation and Conclusions

(1) LLMs show significant potential for automated website categorization, potentially reducing manual labeling needs.
(2) Quantization enables deployment of powerful models in resource-constrained environments.
(3) Model architecture and training data are as crucial as model size for performance.

(4) Task-specific model selection is important; text-only models outperformed LLAVA models (multimodal ones) for this task.
(5) Semantic structure (accessibility tree) is crucial for accurate website categorization.

As shown in Figures 2 and 3, larger models generally achieved higher accuracy but at the cost of increased resource usage.

These findings highlight the potential of open-source LLMs for web-related tasks while identifying areas for future research, such as improving large model efficiency and developing specialized web task architectures.

5.3 Implications and Applications

The results of this research have several implications for website categorization and the broader application of LLMs. LLMs offer a promising approach for automating website categorization at scale, which could enhance the efficiency and accuracy of content filtering, targeted advertising, and web analytics applications. LLMs' capability to process raw HTML content without extensive preprocessing suggests that they may adapt better to the dynamic nature of web content compared to traditional classification methods. Quantization techniques' effectiveness in maintaining accuracy while reducing computational requirements creates opportunities to deploy powerful models in resource-constrained environments, such as edge devices or shared cloud infrastructure. The study highlights the complex trade-offs between accuracy, model size, and resource usage, providing valuable guidance for practitioners in selecting appropriate models for specific use cases and deployment scenarios. The high accuracy achieved by the Gemma model indicates that smaller, specialized models can be highly effective for tasks like web classification. This opens up possibilities for deploying efficient, task-specific models in resource-constrained environments.

5.4 Alternative Approaches and Future Directions

While this research focused on the use of LLMs for website categorization, our findings suggest several alternative approaches and directions for future work. An additional approach is using embedding models for web classification tasks, which could be more practical than language models. Embedding models can succinctly capture the semantic meaning of web content, potentially leading to improved accuracy and efficiency. Considering the Gemma model's remarkably high accuracy despite its smaller size, fine-tuning it with domain-specific data could be a promising direction. This approach may result in a highly accurate and resource-efficient model for web classification. The exceptional performance of the Gemma model, likely attributed to Google training it on extensive web data, indicates that models with access to comprehensive web datasets may have a significant advantage in web classification tasks. Future research could explore leveraging or replicating this knowledge in other models. Broadening the evaluation to encompass a broader range of websites and categories would offer a more comprehensive assessment of model performance across domains and content types. DSPy[13] presents a promising direction for enhancing our website classification system. Its modular structure and optimization capabilities could improve both accuracy and efficiency. We could use DSPy to develop a more advanced pipeline, potentially incorporating retrieval-augmented generation (RAG) for better context before classification. The framework's automatic prompt optimization could lead to more effective LLM instructions, especially for challenging cases. DSPy's support for fine-tuning smaller models aligns with our goal of creating resource-efficient classifiers. Future work could explore how DSPy's techniques can be applied to our task, potentially achieving state-of-the-art results in website categorization while reducing manual prompt engineering and dataset curation efforts. Developing methods to interpret and explain the categorization decisions made by models would contribute to building trust and easing their adoption in sensitive applications.

5.5 Limitations

It is essential to recognize this study's limitations, especially regarding the available computing resources. Due to the university's limited computing power, we could not conduct more extensive experiments involving a more comprehensive range of models and datasets. The available GPU resources restricted the experiments, limiting the number of models we could assess and the size of the datasets we could work with. With access to a more powerful computing infrastructure, future research could explore a more comprehensive set of models, including more diverse architectures, and evaluate their performance on larger, more representative datasets. Additionally, increased computing power would allow for more detailed model performance analysis, such as examining the impact of different quantization levels and settings on classification accuracy and resource utilization. Despite these limitations, the current study offers valuable insights into the potential of open-source LLMs for website categorization and sets the stage for future research in this area.

The research concludes that open-source language models (LLMs), especially specialized models like Gemma, show promise for categorizing websites. Combining embedding models and fine-tuning efficient, web-knowledgeable models may produce the best results for analyzing and classifying web content. As models evolve, they will play an increasingly important role in managing and understanding the vast and dynamic landscape of the World Wide Web. Future work in this area has the potential to enhance further the accuracy, efficiency, and applicability of web classification systems, contributing to more effective and adaptive solutions for a wide range of web-based applications.

6 APPENDIX

During the preparation of this work, we used Grammarly AI, Grammarly, and Quillbot to correct grammatical errors, improve phrasing and word choice, and rephrase sentences in a more academic style. Additionally, we used ChatGPT and Claude AI to help us developing code for data gathering and analysis, as well as to convert Python charts to LaTeX graphs. After using these tools, we thoroughly reviewed and edited all content, including the generated code and converted graphs, taking full responsibility for the final outcome and ensuring the accuracy and appropriateness of all elements in this work.

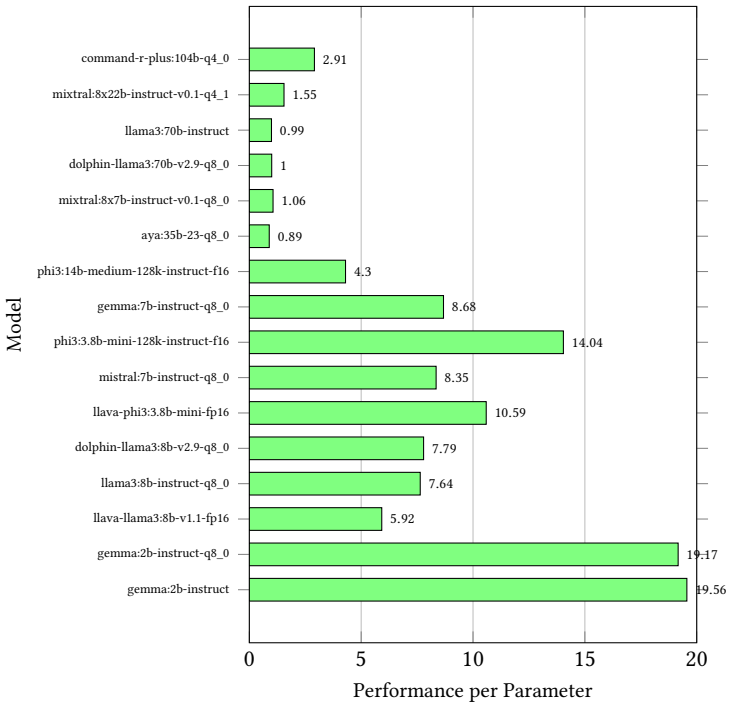

Fig. 4. Performance per Parameter for Different Models

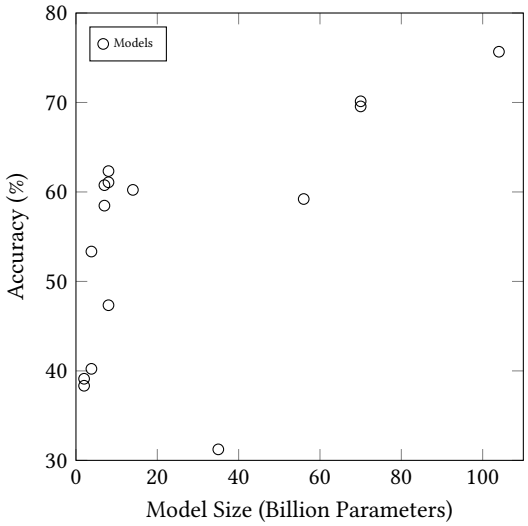

Fig. 5. Accuracy vs Model Size

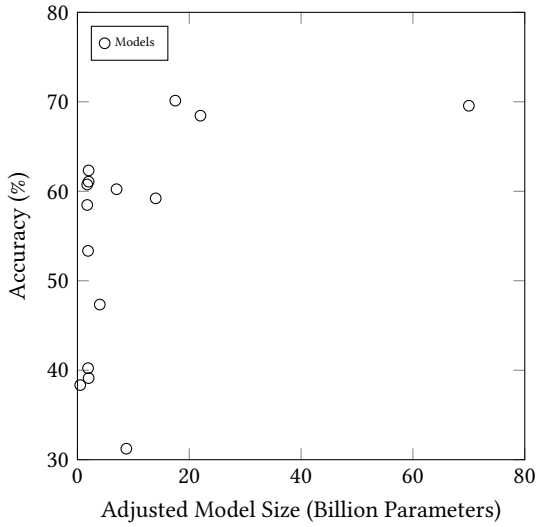

Fig. 6. Accuracy vs Adjusted Model Size (Quantization)

REFERENCES

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219

[2] Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. BinaryBERT: Pushing the Limit of BERT Quantization. arXiv:2012.15701 [cs.CL] https://arxiv.org/abs/2012.15701

[3] Browserless. 2024. Browserless.io: Headless Browser Automation. https://browserless.io. Accessed: 2024-06-27.

[4] Ben Choi and Xiaogang Peng. 2004. Dynamic and hierarchical classification of Web pages. *Online Information Review* 28 (04 2004), 139–147. https://doi.org/10.1108/14684520410531673

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[6] Katerina Goseva-Popstojanova, Goce Anastasovski, Ana Dimitrijevikj, Risto Pantev, and Brandon Miller. 2014. Characterization and classification of malicious Web traffic. *Computers Security* 42 (2014), 92–115. https://doi.org/10.1016/j.cose.2014.01.006

[7] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. 2016. Deep learning for visual understanding: A review. *Neurocomputing* 187 (2016), 27–48. https://doi.org/10.1016/j.neucom.2015.09.116 Recent Developments on Deep Big Vision.

[8] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825

[9] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. arXiv:2401.04088

[10] Xin Jin, Ying Li, Teresa Mah, and Jie Tong. 2007. Sensitive webpage classification for content advertising. In *Proceedings of the 1st International Workshop on Data Mining and Audience Intelligence for Advertising* (San Jose, California) *(ADKDD '07)*. Association for Computing Machinery, New York, NY, USA, 28–33. https://doi.org/10.1145/1348599.1348604

[11] Min-Yen Kan and Hoang Oanh Nguyen Thi. 2005. Fast webpage classification using URL features. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management* (Bremen, Germany) *(CIKM '05)*. Association for Computing Machinery, New York, NY, USA, 325–326. https://doi.org/10.1145/1099554.1099649

[12] Oğuzhan Katar, Dilek Ozkan, GPT, Özal Yildirim, and Rajendra Acharya. 2022. Evaluation of GPT-3 AI language model in research paper writing. https://doi.org/10.13140/RG.2.2.11949.15844

[13] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. *arXiv preprint arXiv:2310.03714* (2023).

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual Instruction Tuning. arXiv:2304.08485

[15] Philippe Martin. 1997. The WebKB set of tools: A common scheme for shared WWW annotations, shared knowledge bases and information retrieval. In *Conceptual Structures: Fulfilling Peirce's Dream*, Dickson Lukose, Harry Delugach, Mary Keeler, Leroy Searle, and John Sowa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 585–588.

[16] Ollama. 2024. Ollama: Local Large Language Model Runner. https://github.com/ollama/ollama. Accessed: 2024-06-27.

[17] Xiaoguang Qi and Brian Davison. 2009. Web page classification: Features and algorithms. *ACM Comput. Surv.* 41 (01 2009).

[18] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL] https://arxiv.org/abs/1910.01108

[19] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295

[20] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971

[21] ValueSERP. 2024. ValueSERP: Affordable SERP Data API. https://www.valueserp.com. Accessed: 2024-06-27.

[22] Tamás Vörös, Sean Paul Bergeron, and Konstantin Berlin. 2023. Web Content Filtering Through Knowledge Distillation of Large Language Models. In *2023 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*. 357–361. https://doi.org/10.1109/WI-IAT59888.2023.00058

[23] Dongxu Zhang and Dong Wang. 2015. Relation Classification via Recurrent Neural Network. arXiv:1508.01006

[24] Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya Model: An Instruction Finetuned Open-Access Multilingual Language Model. arXiv:2402.07827