# Automating Benchmarking Process for Multimodal Large Language Models (MLLMs) in the Context of Waste Disposal

Sundus Hammoud
Institute for Software and Systems Engineering
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
e-mail: sundus.hammoud@tu-clausthal.de

Robert Werner
Institute for Software and Systems Engineering
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
e-mail: robert.werner@tu-clausthal.de

***Abstract*—Multimodal Large Language Models (MLLMs) are systems that can handle both text and non-text input by the user. They can also be prompted to follow certain instructions that can influence their behavior. These capabilities make them an excellent candidate for waste disposal classification. However, these models are trained on general knowledge, and they fail to answer simple questions about recycling because local recycling rules vary across regions. In addition, language models tend to respond in long and detailed text, which makes it very daunting for a human to go through thousands of lines of text while benchmarking such models to evaluate their answers. We propose an approach to automate the benchmarking process in the context of waste disposal and minimize human intervention by introducing a Large Language Model (LLM) to evaluate the answers of another LLM. We also leverage the prompting strategies to achieve this and to resolve the region-based recycling rules problem. We achieved promising results and sped up the benchmarking process significantly by saving researchers from hours of manual evaluation.**

***Keywords-Large Language Model; Multimodal Large Language Model; Benchmarking, LLM-as-a-judge.***

## I. Introduction

Germany has always been one of the leading countries in the field of sustainability. Having a successful recycling system allows us to push the circular economy forward and decrease the dependency on raw materials, saving us from exploiting some of the non-renewable materials like plastic. Recycling also plays a significant role in reducing the waste landfill sizes and therefore protecting the environment from the emissions of toxic greenhouse gases [1].

However, Germany faces a common issue: people often dispose of garbage in the wrong bins. This is often due to confusing recycling rules and limited access to reliable information. A survey in Germany [2] shows that 60% of German citizens lack detailed information on the correct disposal and separation of packaging and household waste. Wrongly disposed waste cannot be used for recycling or even hinders the recycling of materials in the same bin. While many disposal companies offer guidelines and sometimes even human support for waste separation, a low-barrier support bot would be more accessible to users and more affordable.

Businesses have been searching for ways to tackle this situation with the rise of Large Language Models (LLMs) that can interact with only text-based input from a user, and Multimodal Large Language Models (MLLMs) that can handle both text and non-text input. This means that they can interact with images and answer questions about them. This has influenced many businesses to introduce chatbots based on such models to interact with users where they can ask recycling questions and upload images.

Large foundational models currently available are trained on general knowledge regarding waste disposal like Qwen [3] or Llama [4] herds of models. Unfortunately, this is not enough for a model to be useful in Germany, because every region has its own recycling rules, for example, organic waste must be disposed of in the black bin in the city of Goslar but in the green bin in the city of Wolfenbüttel [5]. So, the model must be provided with regional-specific disposal information, according to the region, in which its being used.

Benchmarking a chatbot-based assistant is challenging. Evaluating text answers manually can be a very time-consuming task, because the generated answers are often long and detailed, which makes it extremely difficult to go through thousands of text paragraphs.

Lastly, the research process usually contains many iterations to enhance the algorithm or tweak the model's parameters. So, it is imperative to have a quick evaluation to speed up the process and let the researcher focus more effectively on other aspects of the research.

For these reasons, an automatic benchmarking system is necessary. However, benchmarking a chatbot that has no machine-readable interface but is rather designed to use human language is not possible to achieve algorithmically. Instead, natural language needs to be evaluated with all of its nuances and context.

The rest of the paper is organized as follows: In Section II, the paper proceeds with related work that is similar to the proposed approach and similar projects. Section III focuses on the proposed approach, including data preparation, classification, automated evaluation, and benchmarking evaluators against a human. In Section IV, the results are presented, and then insights about the results are discussed in Section V. Lastly, the paper closes with a conclusion and outlook in Section VI.

## II. Related Work

Along with the variety and improvement of LLMs came the problem of comparing them reliably. Recently, there has been a growing interest in investigating the ability of LLMs

to evaluate texts and even ones generated by other language models. The most important advantage of using LLMs as a judge is that they provide scalability, meaning that they allow for benchmarking many models without the need for a human to spend hours evaluating the performance of a certain model. This also means that comparing the performance of multiple models will take significantly less time [6]. One of the first applications, BERTScore, evaluates text generation and assigns a score that is meant to align with human evaluation [7]. Similarly, LLMs have previously been utilized to generate training data [8] or compare the performance of different LLMs [9], e.g., by generating grading categories and scores [10].

Automating a benchmarking system to evaluate LLMs that are instructed to follow certain tasks is also discussed in [11]. The authors selected text classifiers to do the evaluation task. These classifiers output a score between 0 and 1. When a score is higher than 0.5, it is considered to belong to the category the LLM is evaluating for.

In [12], an LLM is used to evaluate the output of another LLM using a scale of 5 points. Then, the authors compare the LLM evaluation with a human evaluation. The evaluated task is the ability to generate a story based on a given prompt. They evaluate two groups of models, namely, open source models like Llama:7b and Llama2:7b/13b and commercial models like GPT-4-turbo and GPT-3.5-turbo.

Unlike approaches with scoring methods, and evaluating models for general purposes, we propose a benchmarking system that adapts to regional variations. We significantly reduce the time and effort of manual evaluation and introduce a new insight into handling recycling complexities.

## III. Proposed Approach

This chapter covers our approach to benchmarking the support bot based on an MLLM. Firstly, we explain how data for the benchmark was collected and prepared. Then, the automated benchmarking system will be introduced. It consists of two main phases, the classification phase and the evaluation phase. The classification phase will be responsible for prompting the Multimodal Large Language Model (MLLM) used for classification with the recycling rules and collecting answers from the model. In the evaluation phase, these answers will be judged by another model and the system finally outputs the results. Lastly, we also compare a few LLMs that claim to have high linguistic capabilities to select the best judge model for our benchmarking system.

### A. Data Preparation

Benchmarking a recycling system requires a well-designed dataset that spans the general recycling bins and represents objects in a realistic environment. There are 6 categories represented in this dataset, first of all, we have the main four bin categories, which are yellow, blue, green, and black. Two more containers are added which are clothes containers and glass containers to dispose of cloths and glass objects respectively. We rely in this work on the recycling rules in Wolfenbüttel, which were published in an official document on their website that contains lists of objects and the corresponding bin. We have collected a total of 207 images. Figure 1 shows an example of some images from the dataset. For each object inside the list, 3 photos are collected manually using Google's image search or kleinanzeigen.de website, which is a website for people who want to sell an item they have. People can post about the product and upload an image of it.


Figure 1. Examples of the images collected for the dataset.

The images should contain objects that are in a realistic environment, with no white background, the reason for choosing a colorful background is that white backgrounds make it very easy to identify an object in an image because the item would be pretty isolated. However, we want the images to be as realistic as possible and test how well the MLLM can identify the object correctly even with colorful and noisy backgrounds.

### B. The Classification Phase

The first step of the benchmarking system is the classification phase. In this phase, the collected dataset of images will be used and the system will interact with an MLLM and ask it where to dispose of the object in each image and obtain the answers. The system initializes the process by setting the local recycling rules, which the model should follow when being asked to classify the object into one of the recycling bins to be disposed of. The system then reads the images as an input, iterates through them, and sends a request to the MLLM's API, to ask it where to dispose of the item in the image. The language model will generate an answer as text and all answers will be stored to enter the next phase. Figure 2 shows the architecture of the classification system.

The critical element is to provide the MLLM with instructions on how to behave and data to properly assist with sorting waste. This data is region-specific and in order to provide a scalable solution, is provided via a system prompt. A prompt is a text through which a human being can interact with the language model, it is written in natural language and its purpose is to give instructions or information to the model so that when it answers, it follows the user's wish that has been declared in the prompt [13]. An example of a prompt:

"Write a short story about people who figure a way to travel back in time and change certain events in their lives to

help them create a better future in no more than 1000 words."

This prompt specifies the task that it is required to perform for the system. The prompt must be detailed and clear so that the model fully understands the instructions.

There are two types of prompts which the model can interact with. Both are important and will be used in this project:

- **System Prompt:** is a prompt that influences the entire model's behavior, it could be a set of rules to follow or some information related to a context that the system must take into consideration before answering. This prompt is given to the model only once during its initialization and before any user interaction [13].
- **User Prompt:** is when a prompt is given to the system while expecting an answer, through which the user usually interacts with the model [13].

This makes the system prompt a very good solution for the regional recycling rules problem, because the model can be prompted with the recycling rules before a user can interact with it, and then, all the upcoming answers that a model will generate will be following those rules.

There are also a few prompting strategies that have been discussed in the literature, the *persona strategy* is applied in this work This strategy gives the model a personality with perspective and knowledge on how to act if a user asks it a question, just like the type of help a human being would get asking someone in real life with that role [14]. The following system prompt is used in the first phase to instruct the system with the recycling rules, giving it the role of a recycling assistant with a set of rules to follow:

"You are an assistant. Here are the local recycling rules:
1. If an item is made of glass, then it must be disposed of in the glass containers.
2. If an item is clothing--such as jeans, a shirt, t-shirt, dress, shorts, socks, hoodie, pullover, pajamas, or skirt--it must be disposed of at this address: 'Recyclinghof Klein Elbe, 38274 Elbe.'
3. If an item is made of plastic or is a food container, aluminum foil, beverage carton (such as a milk or juice carton), toothpaste tube, bottle of shampoo or soap, plant pot, cutlery, CD or DVD cover, bucket, kids' toys, clothes hanger, pan, bowl, or toothbrush, it must be disposed of in the yellow bin.
..."

The prompt contains several parts, at first we define for the model what the purpose of its existence is and what role it plays. Then we define the context that this model is being used for which is waste disposal, then we give it a set of rules on how to judge in which bin the item belongs. The rules include example objects, which are taken from the Wolfenbüttel document mentioned earlier. We can also see that for the clothes bin, an address of the disposal place has been provided, as mentioned in the documentation provided by the city. For each picture that has been collected, the model will be asked where to dispose of the item that is visible in it.

### C. The Evaluation Phase

In this phase, the system's task is to evaluate the answers from the previous phase. Figure 3 shows that the system will have two inputs, the first one is the output of the previous phase which are the answers generated by the MLLM, and the second is the source of truth file, which contains the image ID, the object inside it, and the correct bin in which it should be disposed of. It should be noted here that the information about the object in the object column is not passed to the model neither during the classification phase nor during the evaluation phase. It is only used for referencing purposes.

The evaluation system will then iterate through the source of truth file and send a request to another text-based large language model to compare the output from the previous step with the bin mentioned in the source of truth and evaluate if the answers are semantically equivalent. If so, the system with output *correct*; otherwise, it will output *incorrect*, and the output will be saved into a file.

*1) LLMs Evaluating LLMs:* According to [6] there are three types of LLM-as-a-judge. In this project, we will apply the approach *Reference-guided grading*, which means the system is provided with a reference answer, which is the correct answer to the question, and another language model's answer, and the model must compare if they match. The reason for applying this approach is the model can not rely on the general knowledge that it was trained with to judge, because the classification system makes decisions according to the regional-based rules in the prompt, and it is tailored for a certain city, which is Wolfenbüttel in our work. So, since the evaluation language model does not know about the recycling rules in Wolfenbüttel, we store the correct answers in the source of truth file, where they will serve as reference answers.

*2) Prompting The Evaluation Model:* In this phase, we want to prompt the model that plays the role of the judge in the evaluation phase. In the evaluation phase, we received the answers from the previous phase, we also have the source of truth which contains the actual answers in the *Bin* column as shown in Figure 3. So, according to the *Persona* prompt strategy [14], we want to design the system prompt for the evaluation model so that it would act like a judge and compare two texts semantically, and output the word *correct* if they semantically match or output *incorrect* if they don't. The following prompt is the system prompt of the evaluation model:

"You are an evaluation assistant. Your task is to compare two texts: the first text contains the source of truth, and the second text contains system answers. Determine if the bin mentioned in the source of truth matches the bin mentioned in the system answer. Respond

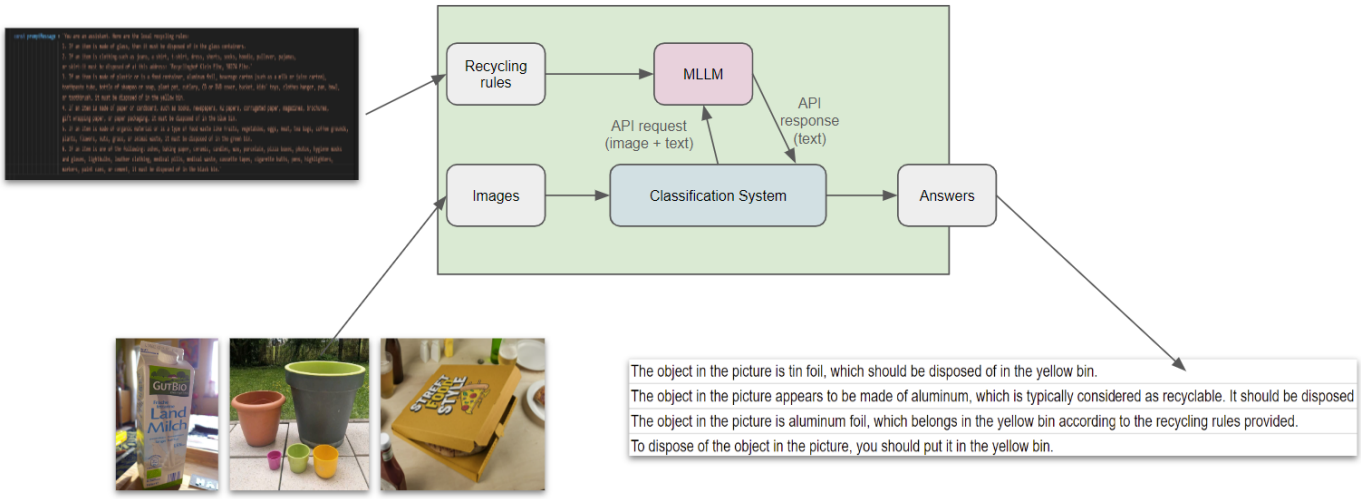



Figure 2. The classification phase of the benchmarking system.

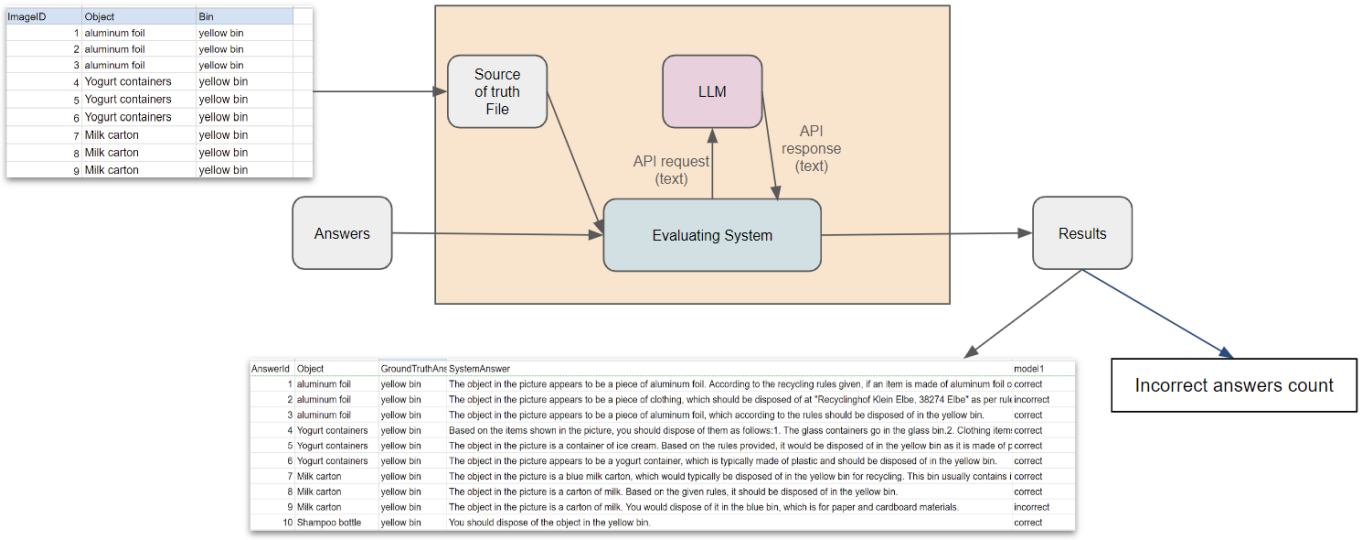



Figure 3. The evaluation phase of the benchmarking system.

`with one word: 'correct' if they match, and 'incorrect' if they don't."`

Now that we set the system prompt, we can start looping over the answers and ask the model if they match the bin mentioned in the ground truth.

### *D. Benchmarking Evaluation Models Against Human Evaluation*

After the setup of the evaluation phase, the last step is to select a large language model that will play the role of the judge. This step is very important because a benchmarking system will only be good if the evaluation model is good. However, there are many large language models that have recently gained the attention of the community and we select the following set of models because they are open-source models, they are also under constant updates, and it is worth mentioning that Qwen2 series of models [15] and Llama3 models [4] both being released around 2 months ago. We will compare the performance of each of them to select the best one that is capable of reasoning about long texts and deciding if they match the source of truth. Here are the selected models:

- Qwen:7b [3]
- Qwen:32b
- Llama3:8b [4]
- Llama3:70b
- Qwen2:7b [15]
- Qwen2:72b
- Llama3.1:8b
- Llama3.1:70b

It should be pointed out that the model name Qwen refers to version 1.5, while Qwen2 refers to version 2. The number

followed by the letter *b* in the name stands for the number of parameters used to fine-tune the model. For example, the model Llama3.1:8b has 8 billion parameters. Each model is equipped with a certain number of parameters that can be used, for example, Llama3.1 provides 8b, 70b, or 450b. We chose to work with a parameter count that is less than 80b because the higher the parameter count is the slower an LLM responds to answers. And also we were limited by the server size the models were installed on. To compare which model performs best, we let all the models evaluate the same answers obtained from the classification phase, using the same system prompt for evaluation, the final output is assembled in a CSV file, and all output models are present in a separate column. After obtaining this output, we need a human evaluation to compare these models against, so we can determine which one is the better judge. So, a new column is added where each answer from the classification phase has been notated by a human as *correct* or *incorrect*, then a confusion matrix has been calculated for each of the 8 models.

We see in Figure 4 that the first three columns (answerId, object, and GroundTruthAnswer) represent the values from the source of truth table. The *SystemAnswer* column holds the answers that were output from the classification phase, the *HumanEvaluation* is the notation added by a human to evaluate the answers in the previous column if they match the ground of truth, and the rest of the columns from model1 to model8 contain the evaluation of the models respectively.

## IV. Results

In this section, we present the results of benchmarking different LLMs as a judge and compare them against the human evaluation.

For this, we calculate the confusion matrix based on the human evaluation as well as the output from the other models. However, since LLMs produce non-deterministic responses, this means that we may receive different results when we interact with them, to solve this problem, we run the classification phase once to obtain the answers and then label them by a human. Then for that output, we run the evaluation models three times, calculate the accuracy, precision, recall, and f1 score for each one, and take the average. Table I shows that the model Qwen:32b outperforms all the others, and performs better than higher version models.

TABLE I. Results of calculating accuracy, precision, recall and f1 score after 3 runs.

| Model | accuracy | precision | recall | F1 score |
|---|---|---|---|---|
| Qwen:7b | 0.72 | 0.62 | 0.95 | 0.75 |
| Qwen:32b | 0.91 | 0.87 | 0.95 | **0.91** |
| Llama3:8b | 0.86 | 0.86 | 0.82 | 0.84 |
| Llama3:70b | 0.91 | 0.87 | 0.94 | 0.9 |
| Qwen2:7b | 0.89 | 0.85 | 0.9 | 0.88 |
| Qwen2:72b | 0.9 | 0.83 | 0.98 | 0.9 |
| Llama3.1:8b | 0.86 | 0.87 | 0.82 | 0.84 |
| Llama3.1:70b | 0.91 | 0.94 | 0.86 | 0.9 |

We also notice that it has a relatively low parameter count with only 32b, compared to Llama3:70b for example.

## V. Discussion

In general, models with a higher parameter count perform better than those with a lower parameter count within the same version, for example, Llama3.1:70b performs better than Llama3.1:8b. However, one should also know that even though a high number of parameters may ensure better results, it is slower in performance and requires more memory and computing energy, so there's always a trade-off between how fast or how accurate you want your application to be. For example, using Llama3.1:8b achieves an f1 score of 0.84, and using Llama3.1:70b would make the system a few seconds slower with only 0.06 improvement because it achieves an f1 score of 0.9.

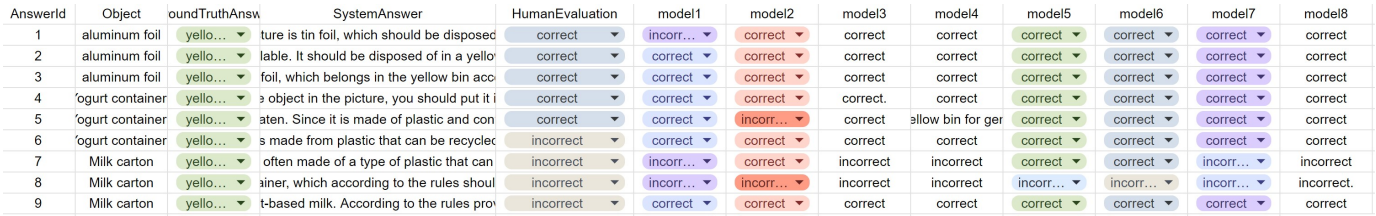
Since we now know that the model Qwen:32b generates the most accurate answers, we want to judge if the classification system is reliable or not, the model used for classification is *Llava-1.6-mistral:latest*. As shown in Figure 3, the evaluation phase will also output the number of incorrect answers, the incorrect answers count using the Qwen:32b is 108/207 while the incorrect answers annotated by a human are 116/207. We see that the numbers are very close, and the benchmarking system results show that more than half of the answers from the classification phase are wrong, so this result tells the researcher that the classification model is not reliable and needs to be either enhanced or replaced.

The cost for the benchmarking system is relatively low since it only handles text input and output Figure 3. The output in particular is very short with a token length of only a single token. The input on the other hand is much longer since it includes the entire output of the classification phase as well as the ground truth answer. On average, the input token length amounts to about 35 tokens [16]. According to current pricing on the Alibaba Cloud with 0.002$/1000 input tokens and 0.009$/1000 output tokens. This makes the evaluation of 1000 answers cost about 0.073$. This means that our benchmarking system is very affordable even for large amounts of data.

## VI. Conclusion and Future Work

In this work, we investigated the possibility of using large language models to automate the benchmarking process and evaluate other models in the context of waste disposal. We compared different models available that can play the role of a judge in benchmarking systems and we saw from the results that Qwen:32b achieved the best performance. However, there are always ways to further improve the benchmarking system results, the following are some suggestions for future work.

- **Keeping up-to-date:** The evaluation model still makes a few mistakes while judging the answers, but the good news is, these models are under constant maintenance, and even new models are always under development. Researchers must stay updated on the model landscape to ensure the best-performing system.



| AnswerId | Object | oundTruthAnsw | SystemAnswer | HumanEvaluation | model1 | model2 | model3 | model4 | model5 | model6 | model7 | model8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | aluminum foil | yello… | ture is tin foil, which should be disposed | correct | incorr… | correct | correct | correct | correct | correct | correct | correct |
| 2 | aluminum foil | yello… | lable. It should be disposed of in a yello | correct | correct | correct | correct | correct | correct | correct | correct | correct |
| 3 | aluminum foil | yello… | foil, which belongs in the yellow bin acc | correct | correct | correct | correct | correct | correct | correct | correct | correct |
| 4 | ′ogurt container | yello… | e object in the picture, you should put it i | correct | correct | correct | correct. | correct | correct | correct | correct | correct |
| 5 | ′ogurt container | yello… | aten. Since it is made of plastic and con | correct | correct | incorr… | correct | ellow bin for ger | correct | correct | correct | correct |
| 6 | ′ogurt container | yello… | s made from plastic that can be recycle | incorrect | correct | correct | correct | correct | correct | correct | correct | correct |
| 7 | Milk carton | yello… | often made of a type of plastic that can | incorrect | incorr… | correct | incorrect | incorrect | correct | correct | incorr… | incorrect |
| 8 | Milk carton | yello… | ainer, which according to the rules shoul | incorrect | incorr… | incorr… | incorrect | incorrect | incorr… | incorr… | incorr… | incorrect. |
| 9 | Milk carton | yello… | t-based milk. According to the rules pro | incorrect | correct | correct | correct | correct | correct | correct | correct | correct |

Figure 4. A snippet of the assembled output of all the models in addition to the classification system's answers and the human evaluation.

- **Prompt Engineering:** There are a set of techniques for structuring the best prompt, in this project, some of them have been applied, but it would boost the performance once this topic can be applied with more depth.
- **Categories-oriented evaluation:** An evaluation system like this gives only an overall view of the classification system. However, this view is not detailed and it does not show the researcher the areas in which the system excels or behaves poorly. For example, in which category the model used for classification *Llava-1.6-mistral:latest* performs the worst? Is it by the yellow bin or the glass container? This would help the researcher to decide where the models need to be improved.
- **Separate models for separate tasks:** In the classification phase, only one model was used to perform both tasks of identifying the object in the image and guessing the correct bin for it. However, there could be models that perform better at image recognition and others that perform better regarding reasoning about the rules in the prompt, so to enhance the classification phase, we propose using two separate models, one for the image recognition task and one for reasoning about the recycling rules.
- **Benchmarking framework:** This project is only a console project, but it can grow into a benchmarking framework, where the users can interact with an interface to upload the source of truth files, upload the dataset, select a classification model, enter the recycling rules and compare the results for different models, datasets and prompts.

  Overall this work has achieved good results, and we believe it is ready to be applied in different domains. With the aforementioned future work, we believe that the benchmarking system can be further enhanced to provide even more reliable results and better insights.

## References


[1] G. Maitlo *et al.*, "Plastic waste recycling, applications, and future prospects for a sustainable environment," *Sustainability*, vol. 14, no. 18, 2022, ISSN: 2071-1050. DOI: 10.3390/su141811637.

[2] A. Subklew, "Verbraucherumfrage zur mülltrennung," Nov. 2020, [Online]. Available: https://www.muelltrennung-wirkt.de/fileadmin/user_upload/Presse/Factsheets_und_Studien/Duale-Systeme_Factsheet_BUS_Umfrage.pdf.

[3] J. Bai *et al.*, *Qwen technical report*, 2023. arXiv: 2309.16609 [cs.CL].

[4] A. Dubey *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024. arXiv: 2407.21783 [cs.AI].

[5] W. L. W. A. (ALW), "Abfallfibel 2024," Oct. 2023, [Online]. Available: https://www.alw-wf.de/index.php/component/phocadownload/file/57-abfallfibel-2024.

[6] L. Zheng *et al.*, *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*, A. Oh *et al.*, Eds. Curran Associates, Inc., 2023, vol. 36, pp. 46 595–46 623.

[7] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with BERT," *arXiv preprint arXiv:1904.09675*, Apr. 2020. arXiv: 1904.09675 [cs.CL].

[8] B. Peng, C. Li, P. He, M. Galley, and J. Gao, "Instruction tuning with gpt-4," *arXiv preprint arXiv:2304.03277*, Apr. 2023. arXiv: 2304.03277 [cs.CL].

[9] W.-L. Chiang *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, vol. 2, no. 3, p. 6, Mar. 2023.

[10] W. Xie, J. Niu, C. J. Xue, and N. Guan, *Grade like a human: Rethinking automated assessment with large language models*, Mar. 2024. arXiv: 2405.19694 [cs.AI].

[11] Y. Chen, B. Xu, Q. Wang, Y. Liu, and Z. Mao, "Benchmarking large language models on controllable generation under diversified instructions," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 17 808–17 816, Mar. 2024. DOI: 10.1609/aaai.v38i16.29734.

[12] C.-H. Chiang and H.-y. Lee, "Can large language models be an alternative to human evaluations?" *arXiv preprint arXiv:2305.01937*, 2023. arXiv: 2305.01937 [cs.CL].

[13] M. Weber and M. Reichardt, "Evaluation is all you need. Prompting generative large language models for annotation tasks in the social sciences. A primer using open models," *arXiv preprint arXiv:2401.00284*, Dec. 2023. DOI: 10.48550/arXiv.2401.00284.

[14] J. White *et al.*, "A prompt pattern catalog to enhance prompt engineering with chatgpt," *arXiv preprint arXiv:2302.11382*, Feb. 2023. DOI: 10.48550/arXiv.2302.11382.

[15] A. Yang *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024. arXiv: 2407.10671 [cs.CL].

[16] AlibabaCloud, "Open source qwen llms: Billing," Sep. 2024, [Online]. Available: https://www.alibabacloud.com/help/en/model-studio/developer-reference/billing-for-tongyi-qianwen-7b-14b-72b.