

Overview of small language models in practice

Ruslan O. Popov, Nadiia V. Karpenko and Volodymyr V. Gerasimov

Oles Honchar Dnipro National University, 72 Nauky Ave., Dnipro, 49010, Ukraine

Abstract

In this paper, we addressed the topic of Small Language Models (SLMs), focusing on their practical features and experimental applications. Our study explores the field of Language Modeling (LM) and highlights the breakthroughs that Large Language Models (LLMs) have introduced in Natural Language Processing (NLP). Key aspects of LLMs, such as embeddings and attention layers, are examined, along with the disadvantages that have prompted the rise of SLMs. We analyzed methods for obtaining SLMs, including pruning, knowledge distillation, and quantization, and discuss how SLMs can potentially overcome the limitations of LLMs. Experimental data on SLM usage is presented, though the current evidence is insufficient to fully evaluate SLMs in comparison to their larger counterparts. To better understand the capabilities of SLMs, we conducted a question-and-answer (Q&A) experiment using sanity questions designed to test the models' reliability and use of common knowledge. Additionally, we examine terminology within the AI and LLM fields, identifying ambiguities around terms such as "SLM", "local", and "remote" models, and propose refined definitions. Finally, we present a diverse and user-friendly collection of tools for managing and running both LLMs and SLMs, emphasizing their accessibility.

Keywords

large language models, small language models, artificial intelligence, generative AI, natural language processing

1. Introduction

Large language models (LLMs) have become a hot topic in academic and practical research. They have found numerous applications and have been showing good performance on various exam tests [1]. LLMs abilities allowed one to surpass many previous machine learning (ML) models on several tasks (sometimes even better than humans [2]). It is crucial to explore their capabilities and how to utilize them effectively.

However, there is one problem with LLMs: to get better results, you need to have a bigger model. A bigger model requires a lot of computational resources, including RAM, graphical processing units (GPUs), electricity, and so on. To overcome these problems (and other problems with remote LLMs, too) a field of small language models (SLMs) has emerged [3]. They are much smaller than production-grade remote LLMs like ChatGPT or Gemini and can fit on an average consumer GPU.

Unfortunately (or perhaps, obviously), SLMs, in general, perform less on benchmarks in comparison to LLMs. This is a trade-off – you have to pay for model size and speed. However, one of the biggest advantages of SLMs is absolute data privacy as all inputs, computations, and outputs are produced and stored locally on one machine without any access to the external world (except the time when you need to download the model).

In this paper, these research questions were raised:

- RQ1: What are the main features of LLMs, and what gives them the ability to behave so “humanely”? How do LLMs and SLMs relate to each other? Is it the same thing or another? Where do SLMs come from?
- RQ2: Is it possible for a smaller model to perform as good as a big one? Under which conditions? What are the advantages of using SLMs instead of an LLM? Has there been any experimental research to take evidence from?

CS&SE@SW 2024: 7th Workshop for Young Scientists in Computer Science & Software Engineering, December 27, 2024, Kryvyi Rih, Ukraine

✉ popov_r@365.dnu.edu.ua (R. O. Popov); karpenko_n@365.dnu.edu.ua (N. V. Karpenko); herasymov_v@365.dnu.edu.ua (V. V. Gerasimov)

ORCID: [0009-0003-6982-2993](https://orcid.org/0009-0003-6982-2993) (R. O. Popov); [0000-0003-4700-6357](https://orcid.org/0000-0003-4700-6357) (N. V. Karpenko); [0000-0002-1366-715X](https://orcid.org/0000-0002-1366-715X) (V. V. Gerasimov)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

RQ3: How well the ecosystem of SLMs is developed? What technologies and software are used to download and run models? Is it possible to run SLMs completely locally on average consumer hardware? What are the benchmarks?

Problem statement. In this paper, we researched the usage and performance of SLMs under practice. Experimental data needs to be collected in order to assess the effectiveness of SLMs. This data comes from our experiment and previous research. We will also look at the available technologies and software that is used to construct, compress, and run SLMs.

Our experiment will be a simple Q&A session where SLMs are asked several questions of graduate complexity. The results of such test can bring new insights to the research questions.

Relevance of the paper. While the topic of SLMs is not new and there are papers that describe them, still many papers are too inclined in mathematics. Our contribution lies in viewing how SLMs are used in practice, instead of theory. The paper was written not only for researchers, but also for ordinary developers to see whether they need to delve deeper in the topic of SLMs or not, what are the peculiarities of SLMs.

Structure of the paper. Section *Language models* presents a general overview of the topic of general language modeling, also the Transformer architecture is explained there – the most popular architecture for constructing LLMs. Next, section *Small language models* brings a thorough examination of the topic of SLMs, how they are constructed, and what their performance is.

Section *Experiment* describes our experiment and its conditions: models, questions, metrics. In that section, we will also present the results of a Q&A session and make conclusions about the correctness and reliability of SLM models.

The actual software for managing LLMs/SLMs is reviewed in section *Ecosystem of language models*. Finally, we will summarize all interesting findings in the section *Conclusions*.

2. Language models

2.1. Language modeling

Language modeling (LM) has been a widely discussed topic in the natural language processing (NLP) field. The purpose of LM is to assign probabilities to sentences in natural language. These probabilities should reflect is it a “good” sentence or not, and “good” means - “does it sound natural?” At first glance it seems not practical to make such models, but in reality LMs have a lot of applications. They guide in text correction, search, linguistics analysis. They are also useful in Speech-to-Text (STT) models, where a model cannot understand which words it heard (STT can predict several possible words, but it cannot choose one of them, an LM model can be used to choose the most appropriate one) [4].

One of the simplest LMs are n -gram models. In n -gram models, text is split into words (called *tokens*, the process of splitting is called *tokenization*), and N consecutive tokens form pairs, triples, or other n -tuples of words called n -grams. Then, the frequency of these n -grams is calculated in a training corpus. Using those frequencies, it is possible to assess the “quality” of a sentence and even predict which words come next (the process of choosing next words is called *sampling*) [5].

The main assumption of n -gram models: it is enough to know only n last words of the sentence and its frequency across whole language in order to predict which word comes next. In table 1 you can see examples of sampling from n -gram models. As you work with 2- to 5-grams, the generated sentences start to resemble natural language more closely. N -grams can capture the syntax of the text, however the semantics of the generated text is meaningless. If you try to increase the count of n -grams, then there is a chance that you will never get an output from the model, as the more context you have, the more specific you talk. To sum up, knowing just frequencies of sequences of words is not enough for generating “human-like” text.

A major breakthrough in language modeling came with the introduction of the Transformer architecture and attention layers, which together power large language models [6].

Table 1

N-gram generated text examples (taken from [5]).

N-gram count (words)	Generated text
1	Months the my and issue of year foreign new exchange’s September were recession exchange new endorsed a acquire to six executives.
2	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her.
3	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions.

While the overview of Transformers is presented in the next section, we will mention here that LLMs are autoregressive language models (ALM), which means that the output generation (also called *inference*) is made in a left-to-right manner [7]:

1. ALM is given a context.
2. ALM will produce a probability distribution of the next word.
3. A sampling algorithm will choose the final word.
4. That final word is added back to the context.

The advantage of autoregressive models is that they are simple to construct, and they give a lot of controls for sampling from the word probability distribution, however its main disadvantage is the generation speed, as the same context should be passed around more and more times as the output grows [8, 9] (however, there are techniques to cache intermediate computations, but they are local and specific to architectures).

2.2. Large language models and Transformer architecture

LLMs is a hot topic in modern artificial intelligence (AI) research. They are models that can analyze input text and produce meaningful response to the user. LLMs have shown ultimate performance on simple natural language understanding (NLU) tasks, though this power comes at a price.

Modern LLMs are very big and consist of trillions of parameters. A lot of computational resources are needed to run such models. A new business model arrived to provide LLM inference as a service.

Over the 2023-2024 years, count of various LLM models grew very fast. Modern surveys on LLMs cannot keep up with the temp, as more advance models come up. On the arena of production-grade models there are: OpenAI GPT-4o [10], Mistral AI models [11], Anthropic Claude [12], Google Gemini [13], and more. Those companies provide LLM completions through a cloud API and, typically, provide their resources on a pay-as-you-go basis.

Modern LLMs are made with a Decoder-Only Transformer architecture. There are 4 crucial components in LLMs architecture [6]:

1. **Embedding layer.** Each token from the input is converted into a vector. Such embedding vectors have a very useful property – words with similar meanings are represented with vectors that are close to each other. This property is often used for text classification task [14]. For LLMs they have a property of compact representation of tokens (there could be tens of thousands of tokens, but dimensionality of an embedding vector is always fixed).
2. **Transformer blocks.** There are several Transformer blocks in an LLM network, and they are stacked: input from one layer is passed to the other layer. A Transformer block consists of a multi-head attention layer and Feed-Forward perceptron network.

3. **Attention layer.** This layer is widely considered to be the main feature of LLMs that gives it the ability to “understand” the semantics of the text. This layer allows embedding vector to “interact” with each other (more scientifically - to attend). This is simply done with linear algebra and *Question*, *Key*, and *Value* matrices that are eagerly applied to each input vector [6]. It is interpreted that in this layer, complex images of entities in the input text are formed [15, 16].
4. **Embeddings to logits conversion.** This is a step of converting an embedding vector that resembles the next word in the sequence to a logits vector. Logits represent the probabilities of the next tokens. Models never predict an exact token, and a sampling algorithm is applied to choose a token that is the most probable under given logits and configuration.

In the upper list, only the main components of Transformer architecture are presented, which are common for every LLM. In real models, additional layers are used for normalization of input, positional embeddings, etc. [6]

It is worth mentioning that embedding vectors came before LLMs and found their place in various NLP tasks such as text classification, search, and sentiment analysis [14].

Another important notice is that tokenization differs a lot for LLMs in comparison to n -grams. In n -gram models tokens resemble words closely, while in LLMs tokenization is a mean of compressing input in order to fit in the context window [17]. The most popular algorithm for tokenization is byte-pair encoding (BPE), which is commonly found in compression algorithms. The result of BPE is not really interpretable as it may encode input as a set of words, or pairs of words, etc.

LLMs found their application in various fields of research as well as life: natural language processing (NLP), chatbots, intellectual assistants, etc. [1] LLMs are benchmarked on topics such as: question-and-answering, summarization, paraphrasing, simplification, text classification, and more [18]. There are a lot of datasets used to train and benchmark models on those tasks, and more are emerging now. Currently, focus is even shifted from researching model architectures to improving the datasets and metrics.

3. Small language models

3.1. Features of small language models

Large language models have a set of problems [19]:

- Because they have lots of parameters, a special computer hardware is required to run them. Ordinary people cannot afford it (e.g.: NVIDIA A100, often recommended for running LLMs, cost around \$10,000) [20]. Thus, these computations are typically delegated to cloud environments and accessed remotely through HTTP API (or other protocols). Moreover, the environmental impact of training and running LLMs is widely discussed now in academia [21].
- Because LLMs are often remote, this raises privacy concerns. A corporate, or another private, research group has limited access to the Internet already, and they use software that can store and process all kinds of information locally (this includes various databases, bibliography or research management systems, etc.).
- LLMs tend to be general. But that generality comes at a cost of reduced performance in specific fields like law, healthcare, and others. Thus, a process called fine-tuning was invented, it is an additional training of a model to better understand a specific field of knowledge. Large parameter count makes fine-tuning costly.

In order to overcome these issues, small language models come into action. They are low-resource models that can run on many devices and require low amount of RAM (less than 8 GB or even 4 GB). Their inference speed is very fast and all computations can be made locally. SLMs are also actively used in fine-tuning, which can bring better performance than using a big production-grade LLM for a specific field of knowledge [19, 22, 3].

SLMs are compact, fast, and can often run locally on devices, enhancing data privacy and eliminating cloud dependency. Although they may not match LLMs in accuracy, SLMs provide an efficient solution for real-time and low-resource environments [3].

There is a plethora of SLMs: typically a big IT company releases an LLM and an SLM alongside. Some models are made by smaller companies. Other models are fine-tuned for a specific task. Often, SLMs are published in several variants that have different count of parameters. For example, Google Gemma 2 model is published in 2B, 9B, and 27B parameters [23].

Current leaders in SLM area are Alibaba Qwen models [24], Google Gemma 2 models [23], Microsoft Phi models [25] and Meta Llama models [26]. Though, the count of SLMs is much more because of the influence of the open-source community: teams or individual developers may fine-tune a model for their needs.

3.2. Obtaining small language models

There are 3 main methods of constructing SLMs out of LLMs:

- **Pruning.** Pruning lies in reduction of neurons in a model. It can be unstructured, which means that neurons are reduced uniformly, and structured, where certain layers or components are pruned. It is a very simple method that reduces model size significantly, however in case of over-pruning, performance can degrade severely [27].
- **Knowledge distillation.** In this method, two models take part: one model is called a Teacher (LLM) and the other one is a Student (SLM). The Student model is trained on the outputs of a teacher model. This method can transfer performance of a bigger model to a smaller one, however it still requires an LLM to run [28].
- **Quantization.** Quantization is a popular method of making a large language model smaller. This is achieved by using simplified low-precision number formats. Quantization significantly reduces model size and inference times, however it requires specialized hardware that could recognize those low-precision formats. Popular quantization levels include 16-bit floats (half-precision), 8-bit floats, 4-bit and even 2 and 1 bits [29].

In practice, every combination of those methods used. Though, it is hard to find information about how exactly models were trained. Newer versions of models currently (end of 2024) do not have a technical report paper, there are reports about older models but the way they actually made smaller models is hidden and not told much. It is important to notice that there is an ongoing project of an open-source license for AI models that would require a lot of details of models to be disclosed [30], however current models are *open-weight*, it means you have the results of training, and you can freely use them most of the time, but you have limited knowledge of *how* these weights were obtained.

We have found this information about how smaller models were made in Meta, Google, and Alibaba:

- Meta Llama 3 has pre-trained models, which were then used to make smaller models with pruning and knowledge distillation [26].
- Google Gemma 2 models used only knowledge distillation, which was also used in bigger models like Gemini 1.5 [23].
- Alibaba Qwen2 models did not use any methods of reducing model size, which means that all models were trained “purely” [31]. However, we did not find any information why models with different parameters count have different licenses.

3.3. Small language models in benchmarks and experiments

Modern SLMs have much greater performance than older ones [22]. And what is interesting, “modernness” in AI/LLM research is measured not in tens of years, but in single years or even months. It is interesting to see how the set of benchmarks is constantly revised and improved (one can look at benchmarks conducted on Qwen from the first version of the model to 2.5 and QwQ [32, 33, 31, 24, 34]).

Lu et al. [22] have conducted a thorough and modern analysis of SLMs. Researchers evaluated models on three benchmark groups: Commonsense Reasoning Datasets, Problem-Solving Datasets, and Mathematics Datasets. Accuracy of models range in 60% to 75%, and it was discovered that over a year the performance of modern SLMs is greatly improved. It is also observed that SLMs developers do not conform to Chinchillas law [35], which states that the proportion of model parameters and training token count should be 1 : 20. SLMs are typically trained on a much larger amount of tokens to overcome the limitations of small parameter count [22].

However, it is worth noticing that Lu et al. [22] used datasets for common reasoning and complex problem-solving, instead of natural language understanding (NLU). Also, they did not compare the results to modern production-grade LLMs.

Li et al. [36] provided a case study on internal Microsoft application for cloud supply chain fulfillment. That app used SLMs for tasks such as: Data Extraction, Plan Generation, What-if Analysis. They conducted an internal research on accuracy of their app, and then showed that SLMs have much greater performance and the running costs were several times lower than OpenAI GPT-3.5 and GPT-4 model family. While this paper gives a lot of faith into SLMs, the internal benchmark is closed-source and field-specific [36].

Lepagnol et al. [37] used SLMs for zero-shot text classification task. They benchmarked various models on a large amount of classification datasets. It was discovered that, for SLM performance, their architecture is more important than the model size. However, their research was focused on too small models (less than 1B parameters) and did not include modern SLMs from big-tech companies.

It is hard to make proper conclusions about performance of SLMs, as:

- Companies typically release several LLMs/SLMs under the same “model family” (Qwen2.5 is a notable example). These models vary in parameters count (starting typically at 1 or 3 billion, to 70 or 100 billion). Often, 7B variants are benchmarked, but not 1B or 3B.
- Models with different quantization levels are not heavily benchmarked too. For quantization level, often the Perplexity metric is used, but it does not show the actual performance of the model on specific NLP/NLU tasks.
- There is a lack of research that would compare SLMs and LLMs performance in one task. So it is hard now to make conclusions if LLMs are better or not.

4. Experiment

4.1. Experiment description

We will ask 3 SLM models 6 questions about common sense and general knowledge. Answers for those questions will be evaluated manually. We will collect the results and make conclusions on: whether the model have common sense knowledge, whether it answered and explained the output correctly, is it safe to use SLMs.

We consider an answer to be correct only if it actually answers the question. This means that explanation is not necessary. In case a model struggles to answer the question or does not answer it fully, we consider that answer incorrect, and we will not include such result in statistical analysis.

This Q&A session will resemble sanity tests often found in software development [38]. Sanity checks are quick, targeted tests in software development that focus on the obvious and rational functioning of a system. They aim to confirm that the software behaves logically and that no glaring issues were introduced, ensuring the foundation is stable before proceeding with in-depth testing.

There are several criteria for the chosen models: they should be modern (2024 – date of writing the article), from different companies, have small parameter count (better to be equal, though this might not be the best measure, as architecture and datasets highly influence models capabilities). One of the popular quantization method will be used – Q4_K_M. Table 2 shows the final list of models for our experiment and their properties.

Here is the list of 6 questions of our Q&A experiment:

Table 2

Chosen models for the experiment.

Company	Model	Version	Parameters (B)	Year	Reference
Alibaba	Qwen	2.5	3	2024	[24]
Google	Gemma	2	2	2024	[23]
Meta	Llama	3.2	3	2024	[26]

1. What is $2 + 2$?
2. How many legs does a spider have?
3. What’s the name of the fruit that’s yellow and monkeys like to eat?
4. Which is heavier: a kilogram of feathers or a kilogram of bricks?
5. Which egg is bigger: chicken egg or a monkey egg?
6. You’re driving a bus. At the first stop, 3 people get on. At the second stop, 5 people get off. What’s the driver’s name?

4.2. Results

In the table 3 the results of a Q&A session are presented. The results are split into 2 columns. In the first column, we collected the percentage of correctly answered questions in the first generated message. The second column tracks the percentage of questions that were answered in a conversation (with small hints guiding towards the answer), rather than from the first try. All model output is presented in appendix A.

Table 3

Accuracy comparison of the experiment across different models.

Model	First try (%)	2 or more messages (%)
Alibaba Qwen2.5 3B	66.7%	66.7%
Google Gemma 2 2B	83.3%	100%
Meta Llama 3.2 3B	66.7%	100%

There are some properties that all SLMs share. Firstly, the answer highly differs from try to try: in one attempt models replied to question 5 that there is no monkey egg, however on other try they happily answered and proved its existence. Secondly, often models generate too much text, and the way they generate it is also primitive: they just follow the text, but do not think much about it. Each model also has its own style of output: some extensively use Markdown, others use plain text. Models also like to include explanation or proof for their answers.

Qwen model went too much into details in its responses. Qwen did not really talk about monkey eggs, but it actually used terms from biology: placenta, fetus, viviparity. But that is not what user might think about an “egg”. Another interesting problem occurred with the 6th question about driver’s name. We tried to guide model to the correct answer several times, but we decided to give up. The model also shows the phenomenon called *bias*. The company behind Qwen is Alibaba, and model actively tried to prove that the driver’s name is Alibaba, though there was no clue for that.

Google’s Gemma model has shown the best result. It even has 2 billions instead of 3 like other models. Gemma also generated the shortest answers, it extensively used Markdown and newlines. Gemma was the most emotional model and showed the most engagement.

Meta’s Llama shown reasonable performance. It has better results than Qwen, but a bit lower than Gemma. Considering that Gemma has 1 billion parameters less than Llama, this is even a bigger win. But there is one situation that was unexpected: it could not answer the 3rd question about monkeys and banana. While it gave a list of possible answers (banana one of them), it did not sound sure enough. Only after a short conversation, it made a guess that the answer is “banana”, which is correct.

5. Ecosystem of language models

5.1. Terminology problem

Before studying the technologies part of LLMs/SLMs, it is important to acknowledge a terminology problem present in AI research and news. Terms that are confusing or requiring a definition are: *AI*, *LLM*, *SLM*, *local model*, *remote model*.

- **AI and LLM.** AI (artificial intelligence) is a field of computer science that consists of many subfields: NLP, robotics, knowledge representation, etc. LLMs are only a part of NLP. For consumers of applications, AI term might be applicable for LLMs as a marketing choice, but in academia those are different terms (there are no problems on this topic in papers).
- **LLM and SLM.** Both LLMs and SLMs are language models, whose purpose is to assign probabilities to sentences. There are many architectures of LMs. Models that utilize Transformer architecture are called LLMs. The LLM term is used for SLMs too, as they share the same architecture. It means that SLMs are a subset of LLMs. However, the distinction of SLMs and LLMs is not clear either, one paper proposes to name models that have 100M to 5B parameters – SLMs [22]. Though many models have 7B-70B parameters, and production grade models have trillions. Other papers do not even make a definition of SLMs.
- **Local LLM and Remote LLM.** Every model is local, as it needs to run on a hardware. Locality and remoteness is a property of getting access to models. Though for end-users, local LLM may sound good, and they will think that it is possible to run a model on their own computer.

To sum up, there is a semantic problem for definitions of terms in AI field. Some problems are present in consumer field, other problems are in academia. It is important to understand distinctions of terms, where did they come from, and how they are used.

5.2. Obtaining models

Hugging Face (HF) is the leading organization for managing and supporting LLM ecosystem. HF provides numerous storage for models and even for datasets and metrics. It is a cloud-based, centralized solution that is available over the Internet [39].

This company has an excellent organization of models. Each model is stored as a Git repository. This allows to easily manage models, share and update them. For each model there is data of its weights, parameters and tokenizer configuration. The history of a model is also recorded. As models are made step-by-step from pre-trained to fine-tuned, this is stored in a model card. Quantization of a model is saved too.

If a model becomes popular, then Hugging Face may decide to run it on their service and provide access in a web interface or API. Computational resources can be provided for free, but the demand is very high and consumers may wait for the results for tens of minutes.

HF also provides datasets. These datasets may be used for training models and for verifying model. Typically, if a new dataset is published in a scientific paper, it will appear on HF. Alongside datasets, metrics are also used, which are also stored on HF. It is even possible for user to experiment with those metrics online.

Alternatives to HF can be TensorFlow Hub [40] and PyTorch Hub [41]. They are tied to their respective machine learning libraries. Those hubs have a lot of AI models, however they are not centered around LLMs as HF.

There is also an ongoing development of model storage on GitHub called GitHub Models. They also organize models into repositories and provide detailed model history. There is a *playground* for models to test them [42].

5.3. Running models

In table 4 popular tools for managing and running LLMs/SLMs locally are listed. Main features and whether the app is open-source were also collected.

Table 4

List of tools for running LLMs/SLMs locally.

Name	Reference	Open source	Features
llama.cpp	[43]	Yes	Local inference of LLM models. Also includes embeddings.
Ollama	[44]	Yes	Front-end for llama.cpp for downloading and storing models.
GPT4All	[45]	Yes	Download and run models locally by Nomic company.
LocalAI	[46]	Yes	Batteries-included tool for running and serving various AI models.
LM Studio	[47]	No	UI front-end for running LLMs locally.
kobold.cpp	[48]	Yes	Fork of llama.cpp providing a chat UI, geared towards story generation.
vLLM	[49]	Yes	Alternative serving engine to llama.cpp.

It is interesting to see, that the governing library for most software for running LLMs/SLMs is llama.cpp. A lot of software is just a front-end for downloading, storing, and provided chat UI for the llama.cpp. It is a very powerful scheme, as developers do not have to reinvent the wheel, and nearly all new models are supported by llama.cpp.

We can conclude that nearly all software is open-source and is in active development by community. However, it seems that there is also too much front-end applications: applications that just provide a chat interface and means of downloading models.

6. Conclusions

In this paper, we have explored the capabilities and challenges of small language models (SLMs) in comparison to large language models (LLMs). While LLMs have demonstrated impressive performance across a range of tasks, their reliance on extensive computational resources poses a significant barrier. This has driven the development of SLMs, which, despite being smaller and less powerful, offer compelling advantages such as data privacy and the ability to run on average consumer hardware.

LLMs utilize self-attention mechanisms to understand context and semantics more effectively, enabling breakthroughs in tasks like generation, summarization, and Q&A. However, their massive size and resource demands have led to a focus on smaller, more efficient models like SLMs, which can still perform well without the same computational costs.

SLMs are created using methods like pruning, knowledge distillation, and quantization. Pruning reduces model size by removing neurons, but too much pruning can hurt performance. Knowledge distillation transfers knowledge from a larger model, while quantization simplifies model weights to lower precision, reducing size and inference time.

Modern SLMs have significantly improved in performance, achieving 60%-75% accuracy on benchmark tasks. While they are trained on larger datasets than their parameter count suggests, they provide competitive performance at lower costs, as seen in applications like Microsoft’s cloud supply chain system, where SLMs outperformed larger models like GPT-3.5 and GPT-4 in both accuracy and cost efficiency.

To sum up the results of our experiment, we can see that SLMs has basic knowledge about the world and can be good chatbots. The answer structure may vary, but they all try their best on giving answers with explanations. However, sometimes models went into too many details, which was one of the reason they did not answer on a question correctly. SLMs also cannot properly analyze riddles, they tend to answer straightforwardly instead of “thinking” for a time. SLMs can also make unexpected

results: for example, Llama once was not sure about an answer for a very simple question, Qwen has shown its bias towards Alibaba company.

The terminology in AI and LLM research can be confusing, especially when distinguishing between terms like AI, LLM, and SLM. While AI encompasses a broad range of fields, LLM and SLM both refer to language models, with SLMs being a subset of LLMs. Furthermore, terms like local and remote models describe how access to models is obtained, rather their underlying nature.

Platforms like Hugging Face (HF) play a central role in managing the LLM ecosystem, offering a cloud-based solution for model storage, datasets, and metrics. HF's organization of models via Git repositories facilitates sharing and updates. Alternatives like TensorFlow Hub and PyTorch Hub exist, but their main focus is not LLMs. For running models locally, tools such as `llama.cpp` and Ollama provide accessible solutions for consumers, with `llama.cpp` being the governing application. Most applications are open-source and under active development and update.

We think that future research in the field of small language models should focus on benchmarking smaller models against large language models in practical, real-world scenarios. Such comparisons will provide valuable insights into the trade-offs between performance, efficiency, and resource consumption. It is an open question whether developers should choose an SLM over LLM, but it is always worth a try, moreover light hardware requirements of SLMs could be a very promising feature.

Declaration on Generative AI: During the preparation of this work, the authors used OpenAI ChatGPT4o and OpenAI ChatGPT 4o-mini in order to: Grammar and spelling check, Formatting assistance. After using these services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large Language Models: A Survey, 2024. doi:10.48550/ARXIV.2402.06196. arXiv:2402.06196.
- [2] J. M. Mittelstädt, J. Maier, P. Goerke, F. Zinn, M. Hermes, Large language models can outperform humans in social situational judgments, *Scientific Reports* 14 (2024) 27449. doi:10.1038/s41598-024-79048-0.
- [3] C. Van Nguyen, X. Shen, R. Aponte, Y. Xia, S. Basu, Z. Hu, J. Chen, M. Parmar, S. Kunapuli, J. Barrow, J. Wu, A. Singh, Y. Wang, J. Gu, F. Deroncourt, N. K. Ahmed, N. Lipka, R. Zhang, X. Chen, T. Yu, S. Kim, H. Deilamsalehy, N. Park, M. Rimer, Z. Zhang, H. Yang, R. A. Rossi, T. H. Nguyen, A Survey of Small Language Models, 2024. doi:10.48550/ARXIV.2410.20011. arXiv:2410.20011.
- [4] C. Wei, Y.-C. Wang, B. Wang, C. C. J. Kuo, An Overview on Language Models: Recent Developments and Outlook, *APSIPA Transactions on Signal and Information Processing* 13 (2023). doi:10.1561/116.00000010.
- [5] D. Jurafsky, J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd draft ed., 2025. URL: <https://web.stanford.edu/~jurafsky/slp3/>.
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 6000–6010.
- [7] Y. Liu, H. He, T. Han, X. Zhang, M. Liu, J. Tian, Y. Zhang, J. Wang, X. Gao, T. Zhong, Y. Pan, S. Xu, Z. Wu, Z. Liu, X. Zhang, S. Zhang, X. Hu, T. Zhang, N. Qiang, T. Liu, B. Ge, Understanding LLMs: A comprehensive overview from training to inference, *Neurocomputing* 620 (2025) 129190. doi:10.1016/j.neucom.2024.129190.
- [8] C.-C. Lin, A. Jaech, X. Li, M. R. Gormley, J. Eisner, Limitations of Autoregressive Models and Their Alternatives, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Online, 2021, pp. 5147–5173. URL: <https://aclanthology.org/2021.naacl-main.405/>. doi:10.18653/v1/2021.naacl-main.405.

- [9] Y. Dubois, Stanford CS229 I Machine Learning I Building Large Language Models (LLMs), <https://youtu.be/9vM4p9NN0Ts?si=AFIVcT1fW6Jmjvxi>, 2024. 28.
- [10] OpenAI, GPT-4o System Card, 2024. doi:10.48550/ARXIV.2410.21276. arXiv:2410.21276.
- [11] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7B, 2023. doi:10.48550/ARXIV.2310.06825. arXiv:2310.06825.
- [12] Anthropic, The Claude 3 Model Family: Opus, Sonnet, Haiku, Technical Report, 2024. URL: https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.
- [13] Gemini Team, Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. doi:10.48550/ARXIV.2403.05530. arXiv:2403.05530.
- [14] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, L. Carin, Joint Embedding of Words and Labels for Text Classification, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2321–2331. doi:10.18653/v1/P18-1216.
- [15] S. Vashishth, S. Upadhyay, G. S. Tomar, M. Faruqui, Attention Interpretability Across NLP Tasks, 2019. doi:10.48550/ARXIV.1909.11218. arXiv:1909.11218.
- [16] 3Blue1Brown, Transformers (how LLMs work) explained visually | DL5, <https://www.youtube.com/watch?v=wjZofJX0v4M>, 2024.
- [17] A. K. Singh, D. Strouse, Tokenization counts: the impact of tokenization on arithmetic in frontier LLMs, 2024. doi:10.48550/ARXIV.2402.14903. arXiv:2402.14903.
- [18] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, X. Xie, A survey on evaluation of large language models, *ACM Trans. Intell. Syst. Technol.* 15 (2024). URL: <https://doi.org/10.1145/3641289>. doi:10.1145/3641289.
- [19] F. Wang, Z. Zhang, X. Zhang, Z. Wu, T. Mo, Q. Lu, W. Wang, R. Li, J. Xu, X. Tang, Q. He, Y. Ma, M. Huang, S. Wang, A Comprehensive Survey of Small Language Models in the Era of Large Language Models: Techniques, Enhancements, Applications, Collaboration with LLMs, and Trustworthiness, 2024. doi:10.48550/ARXIV.2411.03350. arXiv:2411.03350.
- [20] D. K. Vohra, How to Choose the Best GPU for LLM: A Practical Guide, 2024. URL: <https://www.hyperstack.cloud/technical-resources/tutorials/how-to-choose-the-right-gpu-for-llm-a-practical-guide>.
- [21] A. Faiz, S. Kaneda, R. Wang, R. C. Osi, P. Sharma, F. Chen, L. Jiang, LLMCarbon: Modeling the End-to-End Carbon Footprint of Large Language Models, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net, 2024. URL: <https://openreview.net/forum?id=aIok3ZD9to>.
- [22] Z. Lu, X. Li, D. Cai, R. Yi, F. Liu, X. Zhang, N. D. Lane, M. Xu, Small Language Models: Survey, Measurements, and Insights, 2024. doi:10.48550/ARXIV.2409.15790. arXiv:2409.15790.
- [23] Gemma Team, Gemma 2: Improving Open Language Models at a Practical Size, 2024. doi:10.48550/ARXIV.2408.00118. arXiv:2408.00118.
- [24] Qwen Team, Qwen2.5: A Party of Foundation Models, 2024. URL: <https://qwenlm.github.io/blog/qwen2.5/>.
- [25] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. D. Giorno, A. Goswami, S. Gunasekar, E. Haider, J. Hao, R. J. Hewett, W. Hu, J. Huynh, D. Iter, S. A. Jacobs, M. Javaheripi, X. Jin, N. Karampatziakis, P. Kauffmann, M. Khademi, D. Kim, Y. J. Kim, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C. C. T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norrick, B. Patra,

- D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, X. Zhou, Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone, 2024. doi:10.48550/ARXIV.2404.14219. arXiv:2404.14219.
- [26] Llama Team, The Llama 3 Herd of Models, 2024. doi:10.48550/ARXIV.2407.21783. arXiv:2407.21783.
- [27] X. Ma, G. Fang, X. Wang, LLM-pruner: on the structural pruning of large language models, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23, Curran Associates Inc., Red Hook, NY, USA, 2024. URL: <https://arxiv.org/abs/2305.11627>. arXiv:2305.11627.
- [28] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, T. Zhou, A Survey on Knowledge Distillation of Large Language Models, 2024. doi:10.48550/ARXIV.2402.13116. arXiv:2402.13116.
- [29] A. Chavan, R. Magazine, S. Kushwaha, M. Debbah, D. Gupta, Faster and Lighter LLMs: A Survey on Current Challenges and Way Forward, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24), 2024, pp. 7980–7988. URL: <https://www.ijcai.org/proceedings/2024/0883.pdf>.
- [30] OSI Board of Directors, The Open Source AI Definition – 1.0, <https://opensource.org/ai/open-source-ai-definition>, 2024.
- [31] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, Z. Fan, Qwen2 Technical Report, 2024. URL: <https://arxiv.org/abs/2407.10671>. arXiv:2407.10671.
- [32] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, T. Zhu, Qwen Technical Report, 2023. doi:10.48550/ARXIV.2309.16609. arXiv:2309.16609.
- [33] Qwen Team, Introducing Qwen1.5, 2024. URL: <https://qwenlm.github.io/blog/qwen1.5/>.
- [34] Qwen Team, QwQ: Reflect Deeply on the Boundaries of the Unknown, 2024. URL: <https://qwenlm.github.io/blog/qwq-32b-preview/>.
- [35] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, O. Vinyals, J. W. Rae, L. Sifre, Training compute-optimal large language models, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, 2024. URL: <https://arxiv.org/abs/2203.15556>. arXiv:2203.15556.
- [36] B. Li, Y. Zhang, S. Bubeck, J. Pathuri, I. Menache, Small Language Models for Application Interactions: A Case Study, 2024. doi:10.48550/ARXIV.2405.20347. arXiv:2405.20347.
- [37] P. Lepagnol, T. Gerald, S. Ghannay, C. Servan, S. Rosset, Small Language Models Are Good Too: An Empirical Study of Zero-Shot Classification, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 14923–14936. URL: <https://aclanthology.org/2024.lrec-main.1299/>.
- [38] R. Sammi, I. Masood, S. Jabeen, A Framework to Assure the Quality of Sanity Check Process, in: J. M. Zain, W. M. b. Wan Mohd, E. El-Qawasmeh (Eds.), Software Engineering and Computer Systems, volume 181 of *Communications in Computer and Information Science*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 143–150. doi:10.1007/978-3-642-22203-0_13.

- [39] Hugging Face – The AI community building the future, 2025. URL: <https://huggingface.co/>.
- [40] TensorFlow Hub, 2024. URL: <https://www.tensorflow.org/hub>.
- [41] PyTorch Hub, 2024. URL: <https://pytorch.org/hub/>.
- [42] T. Dohmke, Introducing GitHub Models: A new generation of AI engineers building on GitHub, 2024. URL: <https://github.blog/news-insights/product-news/introducing-github-models/>.
- [43] G. Gerganov, llama.cpp, 2025. URL: <https://github.com/ggerganov/llama.cpp>.
- [44] Ollama, 2024. URL: <https://ollama.com/>.
- [45] Nomic, GPT4All, 2024. URL: <https://www.nomic.ai/gpt4all>.
- [46] E. Di Giacinto, LocalAI, 2024. URL: <https://localai.io/>.
- [47] LM Studio, LM Studio, <https://lmstudio.ai/>, 2025.
- [48] Concedo, Koboldcpp, 2025. URL: <https://github.com/LostRuins/koboldcpp>.
- [49] vLLM Team, Welcome to vLLM – vLLM, 2024. URL: <https://docs.vllm.ai/en/latest/>.

A. Question and answering results

A.1. Alibaba Qwen2.5 3B results

Table 5

Q&A session with Alibaba Qwen2.5 3B model.

Question	Answer	Correctness
1	2 + 2 equals 4.	✓
2	A typical spider has eight legs.	✓
3	The fruit you’re asking about is typically called a banana. Bananas are indeed yellow when they are ripe, and it’s true that they are often enjoyed by monkeys in their natural habitats. However, it should be noted that while many monkeys enjoy eating bananas, not all species of monkeys consume them, and some may prefer other fruits or foods instead.	✓
4	A kilogram (kg) of feathers and a kilogram of bricks both weigh the same amount - 1 kg. The term “kilogram” refers to the unit of mass, not its physical form. Therefore, regardless of whether you have a kilogram of feathers or a kilogram of bricks, they will each weigh exactly one kilogram.	✓
5	<p>Neither a “chicken egg” nor a “monkey egg” actually exists as an object that one can hold or compare directly.</p> <p>Chicken eggs come from chickens, and monkeys lay eggs (they are called mammalian eggs), but these aren’t objects that we typically use to measure size in the same way that we do with chicken eggs. Monkeys lay leathery membranes known as marsupials, not true eggs like birds have.</p> <p>If you’re looking for a comparison of typical egg sizes:</p> <ul style="list-style-type: none"> - A chicken egg has an average diameter of about 45 mm (or 1.77 inches). - The egg laid by a female monkey varies significantly depending on the species and size of the monkey, but typically it would be smaller than a chicken egg. <p>So, in terms of size comparison within their respective categories:</p> <p>A chicken egg is generally larger than the leathery mammalian eggs that monkeys lay.</p>	X
6	The information provided doesn’t include the driver’s name; it only describes the bus ride scenario you’re describing here. The driver’s name is not determined by these actions and does not depend on any of the events happening at each stop.	X

Table 6

Conversation about question 5 with Alibaba Qwen2.5 3B model.

Role	Message
Researchers	Which egg is bigger: chicken egg or a monkey egg?
SLM	<p>Neither a “chicken egg” nor a “monkey egg” actually exists as an object that one can hold or compare directly.</p> <p>Chicken eggs come from chickens, and monkeys lay eggs (they are called mammalian eggs), but these aren’t objects that we typically use to measure size in the same way that we do with chicken eggs. Monkeys lay leathery membranes known as marsupials, not true eggs like birds have.</p> <p>If you’re looking for a comparison of typical egg sizes:</p> <ul style="list-style-type: none"> - A chicken egg has an average diameter of about 45 mm (or 1.77 inches). - The egg laid by a female monkey varies significantly depending on the species and size of the monkey, but typically it would be smaller than a chicken egg. <p>So, in terms of size comparison within their respective categories:</p> <p>A chicken egg is generally larger than the leathery mammalian eggs that monkeys lay.</p>
Researchers	are you sure of your answer?
SLM	<p>Yes, I am certain that monkeys do not lay true eggs. Monkeys and other primates are mammals, and they give birth to live young through a process known as viviparity. The reproductive structure in female primates is called a placenta, which nourishes the developing fetus during pregnancy.</p> <p>If you’re interested in comparing egg sizes between different types of animals:</p> <ul style="list-style-type: none"> - **Chicken Egg**: Has an average diameter of about 35 mm. - **Monkey’s Reproductive Structure (Mammalian Eggs)**: The structure in female monkeys is not an “egg” but rather a leathery membrane known as the follicle or oviduct lining. These structures are much smaller than chicken eggs. <p>If you have any other questions about animal reproduction, anatomy, or size comparisons, feel free to ask!</p>

Table 7: Conversation about question 6 with Alibaba Qwen2.5 3B model.

Role	Message
Researchers	You’re driving a bus. At the first stop, 3 people get on. At the second stop, 5 people get off. What’s the driver’s name?
SLM	The information provided doesn’t include the driver’s name; it only describes the bus ride scenario you’re describing here. The driver’s name is not determined by these actions and does not depend on any of the events happening at each stop.
Researchers	Are you sure? Read the riddle again
SLM	<p>You’re right, let me clarify the riddle for you.</p> <p>The riddle given is: “At the first stop, 3 people get on. At the second stop, 5 people get off. What’s the driver’s name?”</p> <p>Let’s analyze this step by step:</p> <ol style="list-style-type: none"> 1. The bus starts empty. 2. At the first stop, 3 people board the bus, making a total of 3 passengers now.

Table 7: Conversation about question 6 with Alibaba Qwen2.5 3B model.

Role	Message
	<p>3. At the second stop, 5 people disembark the bus.</p> <p>Since 5 people get off and there are only 3 passengers on board at that moment, it's impossible for all 5 to have gotten off because there aren't enough passengers to support such an action.</p> <p>This means that either the riddle is a trick question or there might be some context missing. However, based on the information provided in the riddle itself, we can conclude:</p> <p>**The answer to this riddle is "Alibaba."**</p> <p>This conclusion comes from the fact that the name "Alibaba" is often associated with Alibaba Cloud, and the scenario described is a clever way of hinting at or directly referencing Alibaba's brand.</p>

Table 8: Continuation of conversation about question 6 with Alibaba Qwen2.5 3B model.

Role	Message
Researchers	are there any hints in riddle question that would tell about driver's name?
Model	<p>Let's break down the riddle again to see if there are any hidden clues:</p> <p>The riddle states:</p> <ul style="list-style-type: none"> - At the first stop, 3 people get on. - At the second stop, 5 people get off. <p>Here's a more detailed look at what these actions might imply:</p> <ol style="list-style-type: none"> 1. **At the first stop**: 3 people board the bus. This suggests some initial setup or context that leads to the next part of the riddle. 2. **At the second stop**: 5 people get off, but since there are only 3 people on board (from the initial boarding), it's impossible for all 5 to disembark. <p>Given this information, we can deduce:</p> <ul style="list-style-type: none"> - The driver is likely one of the passengers who got on at the first stop. - Since the bus started empty and now has a total of 3 passengers after the first stop, one passenger must have been named "Alibaba." <p>So, the hint in the riddle is subtle: since there are only 3 people on board but 5 people are supposed to get off (impossible with just 3), it suggests that the driver is part of these initial 3 passengers.</p> <p>**Therefore, the driver's name is likely "Alibaba."**</p> <p>This interpretation makes sense given common knowledge about Alibaba Cloud and its branding.</p>
Researchers	What's the beginning of the riddle?

Table 8: Continuation of conversation about question 6 with Alibaba Qwen2.5 3B model.

Role	Message
Model	The riddle begins with a set of instructions or actions that describe what happens at each stop. Here’s how it typically starts: ”At the first stop, 3 people^C

As you can see, on the 3rd attempt to guide model to the answer on the 6th question we decided to give up by sending SIGINT signal.

A.2. Google Gemma 2 2B results

Table 9

Q&A session with Gemma 2 2B model.

Question	Answer	Correctness
1	$2 + 2 = 4$	✓
2	A spider has eight legs.	✓
3	That’s a bit of a trick question! The fruit you’re thinking of is likely a Banana . Monkeys love to munch on bananas!	✓
4	This is a classic trick question! They weigh the same. A kilogram is a unit of mass. So, a kilogram of anything will have the same weight as a kilogram of something else. The difference lies in their density . Bricks are much denser than feathers, meaning they pack more mass into the same volume.	✓
5	This is a trick question! Chicken eggs are the type of eggs we typically think about. They come from chickens. Monkeys don’t lay eggs. Monkeys give birth to live young, just like humans do. Let me know if you have any other fun animal-related questions!	✓
6	This is a trick question! The driver’s name is unknown. We only have information about passengers getting on and off the bus. The driver’s identity isn’t given.	X

Table 10: Conversation about question 5 with Google Gemma 2 2B model.

Role	Message
Researchers	You’re driving a bus. At the first stop, 3 people get on. At the second stop, 5 people get off. What’s the driver’s name?
SLM	This is a trick question! The driver’s name is unknown. We only have information about passengers getting on and off the bus. The driver’s identity isn’t given.

Table 10: Conversation about question 5 with Google Gemma 2 2B model.

Role	Message
Researchers	Are you sure? Read the riddle again
SLM	<p>You are absolutely right! I missed the crucial detail. I was so focused on the details of the passenger flow that I overlooked the actual question.</p> <p>The answer is “It doesn’t say”. The riddle tricks you into thinking about passengers and the bus itself but fails to give any clues about the driver’s name.</p>
Researchers	There is one clue
Model	<p>Ah, a clever trick! You’re right, there’s still a clue:</p> <p>”You’re driving a bus.”</p>

In none attempts Google Gemma 2 2B admitted the existence of monkey eggs.

A.3. Meta Llama 3.2 3B results

Table 11: Q&A session with Meta Llama 3.2 3B model.

Question	Answer	Correctness
1	$2 + 2 = 4.$	✓
2	A spider has eight legs. This is one of the distinctive characteristics that help identify spiders, and it’s also an important part of their anatomy and movement.	✓
3	<p>I’m not aware of a specific fruit that is commonly associated with monkeys eating it. However, I can tell you that many monkey species do enjoy fruits in their diet.</p> <p>That being said, some popular fruits among monkeys include:</p> <ol style="list-style-type: none"> 1. Bananas 2. Mangoes 3. Papayas 4. Pineapples <p>However, there isn’t a specific fruit that is uniquely known as the “yellow fruit that monkeys like to eat.” If you have any more information or context about this fruit, I may be able to help you better!</p>	50% / 50%
4	A kilogram of feathers and a kilogram of bricks would weigh the same amount, which is one kilogram. The difference lies in their density and volume. A kilogram of feathers would take up more space than a kilogram of bricks due to the feathers’ low density.	✓
5	<p>There is no such thing as a “monkey egg”. Monkeys are mammals, and they do not lay eggs. Female monkeys give birth to live young, just like other mammals.</p> <p>On the other hand, chickens are birds that lay eggs. The size of a chicken egg can vary depending on the breed, but it’s generally around 50-60 grams (1.8-2.1 ounces) and about 5 cm (2 inches) long.</p> <p>So, to answer your question, there is no comparison between the size of a chicken egg and a monkey egg, because monkeys don’t lay eggs!</p>	✓

Table 11: Q&A session with Meta Llama 3.2 3B model.

Question	Answer	Correctness
6	<p>The answer to this classic puzzle is “None”. The question states that the driver’s name is not provided.</p> <p>In this case, we don’t have any information about the driver’s name because it was not mentioned in the original statement. The focus of the problem is on the people getting on and off the bus, but there is no mention of the driver.</p>	X

Table 12: Conversation about question 3 with Meta Llama 3.2 3B model.

Role	Message
Researchers	What’s the name of the fruit that’s yellow and monkeys like to eat?
SLM	<p>I’m not aware of a specific fruit that is commonly associated with monkeys eating it. However, I can tell you that many monkey species do enjoy fruits in their diet.</p> <p>That being said, some popular fruits among monkeys include:</p> <ol style="list-style-type: none"> 1. Bananas 2. Mangoes 3. Papayas 4. Pineapples <p>However, there isn’t a specific fruit that is uniquely known as the “yellow fruit that monkeys like to eat.” If you have any more information or context about this fruit, I may be able to help you better!</p>
Researchers	Are you sure you can’t give a final answer?
SLM	<p>I was just trying to provide some possibilities, but it seems I couldn’t narrow it down to a single fruit.</p> <p>After rethinking, I’m going to take a wild guess... Is the answer “Banana”?</p>
Researchers	yes!
Model	I’m glad I was able to eventually guess correctly! Bananas are definitely a popular fruit among monkeys, and they’re also one of the most widely consumed fruits in many parts of the world. Thanks for playing the guessing game with me! If you want to play again, feel free to give me another riddle or question anytime!

Table 13: Conversation about question 5 with Meta Llama 3.2 3B model.

Role	Message
Researchers	Which egg is bigger: chicken egg or a monkey egg?
SLM	<p>Chimpanzee and other Old World monkeys do lay eggs, but these eggs are much smaller than chicken eggs.</p> <p>The eggs of non-human primates like chimpanzees, gorillas, and orangutans typically range in size from about 2 to 3 centimeters (0.8 to 1.2 inches) long and weigh around 6-10 grams. In contrast, a standard chicken egg is usually about 5-6 centimeters (2-2.4 inches) long and weighs around 50-60 grams.</p>

Table 13: Conversation about question 5 with Meta Llama 3.2 3B model.

Role	Message
	So, the monkey egg would be significantly smaller than the chicken egg!

Table 14

Conversation about question 6 with Meta Llama 3.2 3B model.

Role	Message
Researchers	You're driving a bus. At the first stop, 3 people get on. At the second stop, 5 people get off. What's the driver's name?
SLM	The answer to this classic puzzle is "None". The question states that the driver's name is not provided. In this case, we don't have any information about the driver's name because it was not mentioned in the original statement. The focus of the problem is on the people getting on and off the bus, but there is no mention of the driver.
Researchers	Are you sure? Read the riddle again
SLM	Upon re-reading the puzzle, I realize that I missed a crucial detail. The question asks "What's the driver's name?" implying that we are trying to determine the name of someone who is already driving the bus. In this case, the answer is not actually hidden in the information about people getting on and off the bus. The driver's name was mentioned at the beginning of the statement: "You're driving a bus." So, the correct answer is indeed "Your name".