# Exploring Zero-Shot Learning Capabilities of LLMs

Gaurav Gulati
gauravgulati@gmail.com
Dronacharya College of Engineering
India

Garima Rana
gsr15597@gmail.com
Dronacharya College of Engineering
India

Prof. Parveen Kumari
Dronacharya College of Engineering
India

Dr. Ritu Pahwa
Dronacharya College of Engineering
India

## ABSTRACT

Transformative tools in natural language processing have recently emerged in the form of Large Language Models (LLMs), which have resulted in substantial advancements in zero-shot learning (ZSL). Large language models can apply their broad knowledge base to accomplish diverse tasks that they have not been trained on, all without requiring specialized data for each labeled task. ZSL research is a crucial area of machine learning because of its potential to tackle challenges in fields where annotated datasets are limited. The performance of zero-shot learning is significantly affected by the architecture of the models, the variety of pretraining datasets used, and the natural language prompt design.

In this study, systematic experimentation was used to investigate these factors using established benchmark datasets. The proposed methods assess LLMs' capabilities in natural language inference, sentiment classification, and summarization by leveraging specially designed prompts to optimize performance. Our research focused on the impact of model size and developed methods to reduce the occurrence of spurious results typically found in zero-shot scenarios.

Our research reveals that larger models and diverse training data significantly enhance zero-shot learning outcomes, with thoughtfully crafted prompts playing a key role in unlocking the potential of large language models. In addition, we identify key trade-offs that could result in unforeseen outcomes and propose remedies to address these challenges.

This study provides substantial information about the capabilities and limitations of big language models in zero-shot learning. We provide a framework for effectively and responsibly leveraging LLMs by highlighting their advantages and disadvantages, thus allowing their implementation in addressing real-world issues where labeled data are unavailable.

## KEYWORDS

Large Language Models (LLMs), Zero-Shot Learning, Few-Shot Learning, Natural Language Processing (NLP), Prompt Engineering

## 1 INTRODUCTION

The trouble of propagating models from little data continues to plague the areas of machine learning and artificial intelligence. One novel strategy is Zero-Shot Learning (ZSL), that enables models to complete tasks without the need for explicit training examples. Large Language Models' (LLMs') rise has sped up development in this area by offering previously unheard-of chances to finish challenging tasks with less oversight. The dynamic capabilities of big language models in zero-shot learning are examined in this research, which focuses on the substantial knowledge gap in artificial intelligence by examining the models' performance across a variety of unexpected tasks.

Understanding the possibilities and constraints of Zero-Shot capabilities is essential as we develop AI systems that are more adaptable and scalable. Through this study, a clearer knowledge of how prompt engineering and LLM design affect an LLM's endurance for tackling formerly unlabeled jobs will be acquired. The research's conclusions have important ramifications with wide-ranging real-world implications in machine learning, natural language processing, and AI ethics.

Evidently large language models (LLMs) exhibited impressive accomplishments in sentiment analysis, summarization, and natural language inference, with some models achieving these tasks with little to no specialised fine-tuning. Addressing task-specific nuances and ensuring consistent performance across many regions are only two of the ongoing problems. In order to address these issues, this study meticulously examined various model topologies, pretraining datasets, and prompt design techniques for enhanced zero-shot learning performance.

To address these discrepancies, we do extensive experiments on multiple standardised test datasets, evaluate the performance of different Large Language Models on multiple tasks, and investigate the impact of different pretraining data and prompting strategies.

The linkage between model design, dataset variability, and prompt techniques is examined in this paper, with a focus on how these factors affect the performance of Zero-Shot Learning.

Not only this discussion advance our knowledge of LLMs' zero-shot learning capabilities, but it also has the potential to impact the construction of AI applications. The intended goal of the present study is to enhance the efficacy of models in practical environments when task-specific data is likely nonexistent or insufficient.

**In Zero-Shot Learning (ZSL), machine learning models are trained without labeled data to perform tasks that have never been attempted before.** The foundation of ZSL is the utilisation of previously acquired knowledge, which is then expanded to handle new tasks using contextual or semantic information provided during the inference stage, usually in the form of instructions in natural language.

**Few-Shot Learning (FSL) addresses scenarios where a novel job has a restricted number of annotated examples available.** These sparse examples serve as the foundation for FSL's model, which extrapolates the task's fundamental structure and expands

its application to new situations. The input prompt usually contains examples, that allow the model to recognise patterns with comparatively little guidance.

Both ZSL and FSL leverage the power of pre-trained Large Language Models, which allows them to function well in situations when resources are scarce and labeled data is difficult to obtain. While FSL offers a fundamental framework that gives guidance, ultimately improving comprehension and task performance, ZSL demonstrates the model's capacity to generalise beyond particular examples. These theoretical foundations are responsible for the capacity for generalisation of artificially intelligent systems in a wide range of fields, and they are especially useful when gathering labeled data is difficult or costly.

## 2 EXPERIMENTAL SETUP

The **Kaggle**[8] platform was employed as our principal computing environment for this project. With its comprehensive cloud-based setup, Kaggle removes the need for local hardware restrictions while serving the resources indispensable for processing and developing models. Our model evaluation technique was significantly streamlined by the platform's GPU availability and seamless communication with several libraries.

We established and verified the models via Kaggle Notebooks, significantly expedited the workflow and supported remote code execution. Throughout the experimentation phase, optimal performance was guaranteed by the lack of hardware restrictions and the capacity to scale up tools on Kaggle.

We incorporated the **Hugging Face Transformers** library, offering an extensive assortment of pretrained models, such as **Qwen2.5-1.5B-Instruct, Llama-3.2-3B-Instruct, and Gemma-2-2B-it**, for model construction and assessment. These models, which emphasized inferences regardless of prior task-specific training, were optimized for zero-shot learning tasks. For the sake of efficient computation and a seamless relationship with **Hugging Face's APIs**, we relied on **PyTorch** to set up and administer the models.

By utilizing Kaggle's resources, the experiment was carried out effectively and without regard to hardware constraints, offering a strong basis for investigating the potential of huge language models in natural language processing.

## 3 LIBRARIES

In order to evaluate the zero-shot learning capabilities of large language models, we employed vital libraries as PyTorch and Hugging Face transformers. For natural language processing, the Hugging Face Transformers library offers pre-trained models and tools, while the PyTorch library offers an adaptable and effective framework for putting these language models into practice and executing.

### 3.1 Hugging Face[3]

For an expansive variety of natural language processing (NLP) applications, the Hugging Face Transformers library showcases an exceptionally strong open-source framework. Numerous pre-trained models and tools have been made accessible through this library, assisting tasks like text generation, question answering, and zero-shot classification. With the help of its many capabilities, we could quickly implement cutting-edge language models without needing a lot of computational or training time.

Hugging Face makes it easier to integrate and use sophisticated models by offering pre-built pipelines and tokenizers that enable inference operations to be executed directly. Through its smooth integration with frameworks such as PyTorch, computational resources are efficiently controlled, allowing for the study of large language models in a variety of natural language processing scenarios. Because of its adaptability and simplicity of usage, Hugging Face was a vital resource for our zero-shot learning research.

### 3.2 PyTorch[2]

A further significant component of this study was PyTorch, an acclaimed open-source machine learning system. PyTorch, a framework esteemed for its flexibility and efficiency in deep learning applications, supplies robust features for dynamic computational graphs, automatic differentiation, and tensor computation. The aforementioned features made it possible for us to precisely and efficiently build and evaluate large language models.

High-performance inference tasks were supported in our study by using PyTorch to load and execute the chosen pre-trained models. Its interoperability with Hugging Face Transformers made it possible for the libraries to integrate seamlessly, guaranteeing that zero-shot learning procedures performed smoothly. PyTorch's dynamic computation graph feature proved especially useful for managing intricate models and maximizing computational effectiveness when conducting experiments.

A simplified process for putting large language models into practice and assessing them was made possible by the merging of Hugging Face Transformers and PyTorch. In aggregate, these libraries provide the computational efficiency, scalability, and flexibility needed to thoroughly investigate models' zero-shot learning potential. This combination made them essential parts of our research framework because it enabled us to successfully handle the difficulties presented by various NLP tasks.

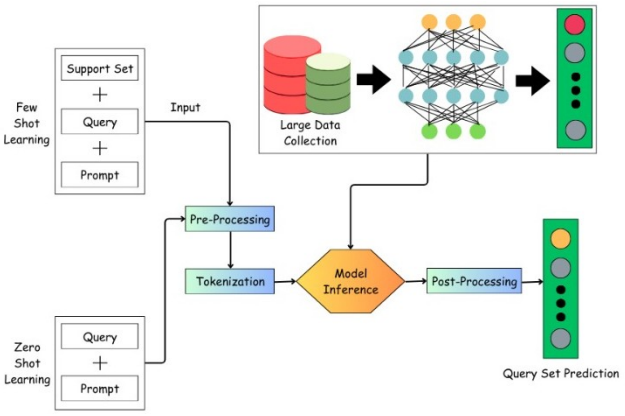


**Figure 1: Zero-shot Inferencing Pipeline**

# 4 DATA COLLECTION

Assessing the large language models' (LLMs') zero-shot learning capabilities required careful consideration of the data collecting procedure. This section describes the data collection, processing, and organization approach, which includes using the **Google Boolean Question (BoolQ) Dataset**[1] in a structured format consisting of a question and a Boolean response for efficient model evaluation.

## 4.1 Data Capture Setup

The **BoolQ** dataset, which consists of yes/no questions combined with contextual excerpts, served as the main dataset for this investigation. In order to make sure it was accurate and in accordance with the input criteria of the designated models, this dataset was pre-processed. This configuration offered a solid framework to rank LLMs' classification and reasoning abilities in a zero-shot scenario, accentuating their knack to perceive and adjust to contextual information.

## 4.2 Data Format and Sample Collection

The BoolQ dataset depicts each sample as a question and context pair having a binary label corresponding to "FALSE" or "TRUE" as the response. The dataset has been structured as follows: **question, context, and label.** For illustration: **"Is the sky blue?" , "The sky appears blue when viewed from Earth during daylight hours." , "TRUE"** In accordance to this format, the inquiry appears first, then the context with pertinent details, and finally the binary label with the response. A wide range of questions and instances are incorporated in each example so that we had an adequate base of data to judge our algorithms' zero-shot learning abilities.

## 4.3 Challenges and Considerations

A number of issues that we ran into during the evaluation of our zero-shot learning models needed careful thought. Assuring consistent performance across many datasets was one of the main challenges; in particular, our models did not attain the anticipated accuracy on the GSM8K[5] dataset. The intricacy of applying larger models to specific tasks without fine-tuning was brought to light by this constraint, which occasionally produces less-than-ideal results for specialized or smaller datasets.

In light of these boundaries, we opted for smaller models like "qwen2.5_1.5b", "llama3.2_3b", and "gemma-2-2b-it" over larger ones that would have yielded more precise results but expected a greater amount of computing power. Notably on the Kaggle platform, where computational assets are constrained, the smaller models rendered a better trade-off between performance and resource usage. The selection of these models was contingent on their propensity for successfully carrying out zero-shot learning tasks having reasonable amount of resources.

Verifying the models' resilience against various input queries presented a further challenge. Despite having pre-trained models, our models occasionally experienced trouble with more complex reasoning tasks since real-time inference without fine-tuning in zero-shot settings was required. This concentrated on the trade-off between efficiency and accuracy when working with large, pre-trained models in a zero-shot setting.

All things taken into consideration, the models offered a firm foundation for determining zero-shot learning capabilities. Nevertheless we had to choose smaller models considering that we were working with specialized datasets and had confined platform resources. By making this choice, we were able to efficiently research zero-shot learning while taking advantage of the resources disposal to us.

# 5 RESULTS

The BoolQ dataset was used to assess the zero-shot learning models' performance using the following important metrics: accuracy, precision, recall, and F1-score. These metrics offer a thorough assessment of the models' accuracy in classifying the data while maintaining a balance between true positive and false positive rates. For a thorough comparison, the categorization reports for each model are described in Tables 1, 2, and 3 illustrating both their advantages and disadvantages.

## 5.1 Classification Performance[4]

**Table 1: Classification Report for Qwen**

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| FALSE | 0.74 | 0.70 | 0.72 | 3553 |
| TRUE | 0.82 | 0.85 | 0.84 | 5874 |
| **Accuracy** | 0.79 (9427 samples) | | | |
| **Macro Avg** | 0.52 | 0.52 | 0.52 | 9427 |
| **Weighted Avg** | 0.79 | 0.79 | 0.79 | 9427 |

**Table 2: Classification Report for Gemma**

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| FALSE | 0.81 | 0.71 | 0.76 | 3553 |
| TRUE | 0.84 | 0.90 | 0.87 | 5874 |
| **Accuracy** | 0.83 (9427 samples) | | | |
| **Macro Avg** | 0.82 | 0.81 | 0.81 | 9427 |
| **Weighted Avg** | 0.83 | 0.83 | 0.83 | 9427 |

**Table 3: Classification Report for Llama**

| Class | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| FALSE | 0.51 | 0.98 | 0.67 | 3553 |
| TRUE | 0.97 | 0.44 | 0.60 | 5874 |
| **Accuracy** | 0.64 (9427 samples) | | | |
| **Macro Avg** | 0.50 | 0.47 | 0.43 | 9427 |
| **Weighted Avg** | 0.80 | 0.64 | 0.63 | 9427 |

Different models showed different levels of performance. The Qwen model's weighted F1-score was 0.79, and its accuracy was 79%. With a weighted F1-score of 0.83 and an accuracy of 83%, the Gemma model fared better than Qwen. Despite its excellent precision for the TRUE class, Llama presented a weighted F1-score of 0.63 and a lower overall accuracy of 64%.

These findings demonstrate the trade-offs between consistency and accuracy in various zero-shot learning algorithms. Gemma had the best overall performance, while Qwen yielded competitive scores, and Llama showed promise in managing particular situations, despite recall issues for several classes.

## 5.2 Model Architecture Differentiation

The architectures as well as applications of the three models employed for this project fluctuate extensively:

- **Qwen2.5-1.5B-Instruct:[9]**
  - ***Compact Transformer Architecture:*** This kind of transformer is ideal for instruction-following tasks and possesses a simplified architecture with 1.5 billion parameters.
  - ***Multilingual Training and RLHF:*** To strengthen its alignment with a range of human preferences, the model incorporates multilingual training capabilities and makes use of Reinforcement Learning from Human Feedback (RLHF).
  - ***Optimized for Efficacy:*** Qwen2.5-1.5B-Instruct prioritizes low latency and computational economy, making it especially suitable for deployment in contexts with limited resources, assuring quick and economical performance.
- **Llama-3.2-3B-Instruct:[6]**
  - ***Dense Transformer Architecture:*** With its robust 3 billion parameter architecture, this model provides impressive generalization capabilities for a number of tasks.
  - ***Extensive Pretraining and Fine-Tuning Support:*** It is adaptable for specific applications due to to its extensive pretraining across a variety of domains, as well as its support for parameter-efficient fine-tuning techniques like LoRA and rich natural language processing.
  - ***Versatility with Higher Resource Demand:*** Designed for a variety of NLP tasks, Llama-3.2-3B-Instruct balances capacity and resource considerations by delivering good performance at a higher computational cost than smaller models.
- **Gemma-2-2B-it:[7]**
  - ***Generalized Multi-Task Model Architecture:*** This model has a flexible architecture with two billion parameters that is tailored for zero-shot and multi-task learning situations.
  - ***Effective Inference and Multi-Task Pretraining:*** Gemma-2-2B-it ensures uniform and efficient performance on a range of activities by utilizing multi-task pretraining in conjunction with excellent inference approaches.
  - ***Strength in Classification and Reasoning:*** The model is very flexible for dynamic, on-the-fly jobs and performs exceptionally well in reasoning and classification tasks, with a focus on real-time zero-shot applications.

The trade-offs among the three models' efficiency, generalization, and task-specific performance are highlighted in this comparison. On the Kaggle platform, smaller models like Qwen were selected due to their computational efficiency, whereas Gemma offered well-balanced multitasking capabilities. Llama's greater computational requirements limited its generalization potential. The models utilized in this project were all pre-trained for zero-shot inference tasks and include a total of 1.5 billion to 3 billion parameters. These models use revolutionary techniques for natural language processing that have been optimized for generalization across a range of NLP tasks.

**Table 4: Comparative Summary of Model Architectures**

| Feature | Qwen | Llama | Gemma |
|---|---|---|---|
| **Parameters** | 1.5B | 3B | 2B |
| **Type** | Compact | Dense | Multi-Task |
| **Focus** | Efficiency | Generalization | Reasoning |
| **Strength** | Low Latency | Versatility | Multi-Task |
| **Limitations** | Niche Tasks | Resource-Heavy | Complexity |
| **Best Fit** | Edge/Cloud | NLP Diversity | Zero-Shot Tasks |

## 5.3 Resource Utilisation

On the Kaggle platform, our models exhibited efficient achievement with reduced latency during inference in terms of resource usage. Even though the models had a lot of parameters (1.5 billion to 3 billion), the zero-shot learning tasks were handled well, guaranteeing timely and precise answers. The models' capacity to manage intricate natural language tasks with little computing overhead is demonstrated here, which qualifies them for real-time inference in scalable cloud-based environments such as Kaggle.

# 6 CONCLUSION

This study highlights the intriguing potential of utilizing large language models for real-time zero-shot learning problems. We showed effective performance on the Kaggle platform via integrating the Hugging Face Transformers library with leveraging models like Qwen2.5-1.5B-Instruct, Llama-3.2-3B-Instruct, and Gemma-2-2B-it. Without requiring task-specific fine-tuning, the models demonstrated their versatility in generalizing across a broad range of natural language processing tasks by doing exceptionally well on inference tasks.

The models produced precise and fast responses for zero-shot reasoning problems, yielding very excellent results. This demonstrates that even with challenging NLP problems, large pre-trained models can do high-performance inference in real-time.

There are a number of intriguing prospects to advance the models' capabilities in the future. To broaden the models' range of applications, we may investigate other zero-shot learning challenges, increase model efficiency using strategies like quantization, or modify the models for particular domains. Furthermore, the models' general performance and usefulness could be expanded by incorporating more complex multi-modal capabilities or optimizing them for specific use situations. This project paves the way for further research into large language models and zero-shot learning, spurring innovation in a number of fields.

# 7 ETHICAL CLEARANCE

We independently selected and processed the data for this research in order to assess the specified models' zero-shot learning capabilities. To ensure a thorough setup for testing the models' capacity to

execute inference without fine-tuning, this method involved compiling a range of tasks, which were subsequently organized and labeled for use in our model evaluation.

By overseeing the labeling and data preparation phases, we made sure that the process was accurate and consistent. Additionally, we made sure that the evaluation was carried out in a method that preserves privacy because the models used for inference do not rely on sensitive or personal data. Moreover, employing pre-trained models from sources like Hugging Face guarantees that no sensitive user data is incorporated, as the models operate on general language tasks without the need for personally identifiable information.

## 8 ACKNOWLEDGMENT

We genuinely appreciate the essential suggestions as well as critiques from our mentor, Dr. Ritu Pahwa, and instructor, Prof. Parveen Kumari, which greatly improved the caliber of our work. We would want to give credit to the open-source community, particularly the creators of PyTorch, Hugging Face, and the pre-trained models that were utilized in this project. In order for us to deploy a successful zero-shot learning system for natural language processing jobs, their dedication to making machine learning technologies available and making resources easily accessible was crucial. Additionally, We also thank the authors of the research publications and sources that helped create and comprehend these potent models.

## REFERENCES

[1] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova.
[2] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. arXiv:1912.01703 [cs.LG] https://arxiv.org/abs/1912.01703
[3] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Pierre Jégou, Pierric Cistac, Timothée Rault, Renaud Louf, Morgan Funtowicz, Zéphir Ma, et al. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6
[4] Zeljko Vujovic. 599–606. https://doi.org/10.14569/IJACSA.2021.0120670
[5] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman.
[6] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. arXiv:2302.13971 [cs.CL] https://arxiv.org/abs/2302.13971
[7] Gemma Team. https://doi.org/10.34740/KAGGLE/M/3301
[8] Kaggle Inc. *Kaggle: Your Home for Data Science.* https://www.kaggle.com Accessed: 2024-12-15.
[9] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. arXiv:2409.12186 [cs.CL] https://arxiv.org/abs/2409.12186