Lightweight Sentiment Analysis: Comparing Small LLMs, Large Models, and Traditional ML Approaches
Small NLP: CS 7643

Jaden Zwicker, Houshmand Abbaszadeh, Third Author
Georgia Institute of Technology
Atlanta, GA 30332
jzwicker3@gatech.edu, habbaszadeh6@gatech.edu, thirdauthoremail@gatech.edu

Abstract

Recent advancements in large language models (LLMs) have set unprecedented benchmarks in sentiment analysis, but their exceptional performance often comes at the cost of significant computational resources. This project investigates the viability of a lightweight, fine-tuned deep learning model capable of running on modest hardware, as a cost-effective alternative to models like BERT and GPT. Using IMDb's movie review dataset [3], we evaluate the performance of a 135M-parameter small LLM, "SmolLM," leveraging few-shot learning and fine-tuning to improve its sentiment analysis capabilities.

The project benchmarks SmolLM against state-of-the-art LLMs, non-transformer deep learning networks such as CNNs and LSTMs, and traditional machine learning approaches like Logistic Regression, Naive Bayes, and Random Forests. Performance metrics are derived from internal experimentation as well as from existing literature on this well-studied dataset.

Our goal is to demonstrate how fine-tuned small models can offer a competitive alternative to traditional methods and larger LLMs for relatively simple NLP tasks. In addition, we explore the diminishing returns of computationally intensive models, highlighting the potential to reduce deployment and operational costs without sacrificing performance. The outcomes of this work aim to guide practitioners in balancing accuracy with resource efficiency in sentiment analysis tasks.

1. Introduction/Background/Motivation

MENTION THE PARAM SIZE OF THE MODELS AND TIE IT BACK TO THE PROBLEM WE ARE SOLVING AND WHAT DIFF IT WILL MAKE

This project aims to compare the performance and computational costs of small and large language models for sentiment analysis on the IMDb movie reviews dataset [3]. The objective is to demonstrate that simpler models, when properly tuned, can deliver competitive performance without the need for expensive computational resources. This project will also explore how the small and large language models compare to traditional machine learning models like naive bayes, logistic regression, and random forest learners. By showcasing the effectiveness of lightweight models, we seek to address the widespread over-reliance on large, resource-intensive models and provide a sufficient and cost effective alternative.

Furthermore the source of the small models, which we sought to fine tune, mentions that they were specifically designed for this type of use case [1]. In their analysis they mention that iPhone 15 Pros have 8GB of DRAM which is sufficient to run all three of their SmolLM2 Models [2]. With this they provide a memory footprint, which is in Figure 1, of the models in their default configurations proving just how little computational resources are needed to run these models which we hope to get to a point of comparable performance to larger industry grade LLMs.

In today's world, sentiment analysis for simple tasks like review classification is often performed using heavy-weight models such as BERT and GPT. The limits of the current practice include longer training and inference times, higher energy consumption, and restricted accessibility for smaller organizations or individuals. These models also require expensive hardware and consume significant computing resources, making them costly to implement. This approach is extremely resource intensive for straightforward tasks, whereas lightweight models can achieve competitive results at a fraction of the cost and training time.

1.1. Why this project matters

If this project succeeds, and the SmolLM is able to achieve competitive results, then this will provide a means for small businesses and even individuals to implement sim-

Model	bf16	int8	int4
SmolLM-135M	269.03	162.87	109.78
SmolLM-360M	723.65	409.07	251.79
SmolLM-1.7B	3422.76	1812.14	1006.84

Figure 1. Memory footprint of SmolLM models [1].

Model	#Param	#Layers	#Head	#KV-Head	Emb Dim	Hidden Dim	LR	BS
SmolLM-135M	135M	30	9	3	576	1536	3e-3	1M
SmolLM-360M	362M	32	15	5	960	2560	3e-3	1M
SmolLM-1.7B	1.71B	24	32	32	2048	8192	5e-4	2M

Figure 2. Architecture details of SmolLM models [1].

ple sentiment analysis without the need for advanced hardware or significant computing resources. Sentiment analysis would become more accessible and usable to a wider range of clients, improving their business insights and application capabilities

1.2. Dataset Selection

The dataset chosen to test the various models sentiment analysis capabilities was the IMDb's movie Review dataset [3]. This dataset contains strings of movie reviews, which was pulled from IMDb's website in 2011 making it a sample of the total amount of reviews they store. Each movie review sample is accompanied by the dataset's binary sentiment classification of Positive or Negative. The dataset contains 50,000 total samples with 25,000 being for training and 25,000 for testing with an approximately 50/50 split between Positive and Negative classifications. These specifics of the dataset make it very balanced leading to it being well tested in literature and the perfect benchmark for basic sentiment analysis. Given the datasets format as strings already and it being from the same creators as the models we were testing no pre-processing was needed outside of tokenizing each input prompt before feeding it into the language models.

2. Approach

Mention the models were run on A100 with 40gb for consistency among tests

2.1. SmolLM Choice

Talk about why we chose the models in more detail for fine tuning and beating LLMs, ie small and can run on phone and open source. Cite this [1]. Also, mention why 135M over the other models

Important: Mention any code repositories (with citations) or other sources that you used, and specifically what changes you made to them for your project.

2.2. Generating SmolLM Benchmarks

To first investigate or compare the variety of models mentioned some initial tests and benchmarking had to be done. While traditional ML techniques for sentiment analysis and well documented LLMs have existing literature providing metrics for them running on the IMDb dataset [3] our choice for tuning a SmolLM model would require us to determine the baseline performance. Hence one of the first goals of this project was to test all three sizes of SmolLM models in both zero-shot and few-shot scenarios so later comparisons could better be made. This benchmarking was done on the 25,000 test samples from the dataset with each sample getting its own individual prompt to the model (no batching was used).

An initial problem which was quickly realized from the zero-shot testing was that the SmolLM models did not always return a clear Positive or Negative classification of the movie prompt. Since no previous examples of answering the question were given (few-shot did this) the smaller sized models such as SmolLM-135M would simple repeat the question back or answer with random commonly used words such as 'I' or The repeatedly. To accommodate for this issue our initial approach had to be adjusted. Rather then parse the full reply that the models generated we took the top 50 most probable next words (logits) that the model generated and parsed through them to see if Positive of Negative appeared and if it did the first one was taken as the models answer. This methodology keeps the premise of sentiment analysis while taking into account that LLMs can give wordy responses making it hard to determine what sentiment they actually gave the prompt. Also, if the model was unable in those 50 most likely words to give an answer of either Positive of Negative this was treated as a default Negative answer however, we tracked these scenarios to bring up in later data points and analysis.

Another initial testing issue that arouse, all of which allowed us to learn about the models better before fine tuning, was the dependence on the prompting method for determining the Models response. For example, if the prompt was phrased as follows: ***"Only Answer if this Movie Review is Positive or Negative:"*** then the smaller models would be more likely to select the first mentioned classification. So if the prompt asked Positive vs Negative then the model would be biased towards the Positive class. We could not determine a good way to elevate this for the zero-shot testing which is why we set the default class to Negative to assist in biases the choice the other direction. For the more advanced larger models or for the few-shot prompts this was less of an issue but during testing some bias could have been shown to the Positive class.

Finally, for the few-shot prompting of the models 3 example reviews were used before appending the given sample to the end of them. The 3 example reviews were gener-

ated by us and were purposely kept short for the sake of the models interpretation of them, they can be seen in Table 1.

2.3. Gathering Data on LLMs and ML techniques

2.4. Fine-tuning SmolLM-135M

(10 points) What did you do exactly? How did you solve the problem? Why did you think it would be successful? Is anything new in your approach?

(5 points) What problems did you anticipate? What problems did you encounter? Did the very first thing you tried work?

3. Experiments and Results

(10 points) How did you measure success? What experiments were used? What were the results, both quantitative and qualitative? Did you succeed? Did you fail? Why? Justify your reasons with arguments supported by evidence and data.

Important: This section should be rigorous and thorough. Present detailed information about decision you made, why you made them, and any evidence/experimentation to back them up. This is especially true if you leveraged existing architectures, pre-trained models, and code (i.e. do not just show results of fine-tuning a pre-trained model without any analysis, claims/evidence, and conclusions, as that tends to not make a strong project).

3.1. Testing Default SmolLM Models

Results are given in Tables 2 and 3.

SmolLM2-135M Zero-Shot adjusted accuracy when counting unknown predictions as false answers is 0.38

3.2. Comparison Between LLMs, Traditional ML, and SmolLMs

Experiment 4

3.3. Results of Fine-tuning SmolLM-135M

recompare and give data and why stuff was how it was

4. Other Sections

You are welcome to introduce additional sections or subsections, if required, to address the following questions in detail.

(5 points) Appropriate use of figures / tables / visualizations. Are the ideas presented with appropriate illustration? Are the results presented clearly; are the important differences illustrated?

(5 points) Overall clarity. Is the manuscript self-contained? Can a peer who has also taken Deep Learning understand all of the points addressed above? Is sufficient detail provided?

(5 points) Finally, points will be distributed based on your understanding of how your project relates to Deep Learning. Here are some questions to think about:

What was the structure of your problem? How did the structure of your model reflect the structure of your problem?

What parts of your model had learned parameters (e.g., convolution layers) and what parts did not (e.g., post-processing classifier probabilities into decisions)?

What representations of input and output did the neural network expect? How was the data pre/post-processed? What was the loss function?

Did the model overfit? How well did the approach generalize?

What hyperparameters did the model have? How were they chosen? How did they affect performance? What optimizer was used?

What Deep Learning framework did you use?

What existing code or models did you start with and what did those starting points provide?

Briefly discuss potential future work that the research community could focus on to make improvements in the direction of your project's topic.

5. Work Division

A summary of each authors contributions are provided in Table 5.

Few-shot Prompt 1	Movie Review: I loved this movie ! So good plot ! Only Answer if this Movie Review is Positive or Negative: Positive
Few-shot Prompt 2	Movie Review: I hated this, could be a lot better Only Answer if this Movie Review is Positive or Negative: Negative
Few-shot Prompt 3	Movie Review: This move was so good I would recommend to all my friends! Only Answer if this Movie Review is Positive or Negative: Positive

Table 1. Few-shot prompts used to test default SmolLM models.

Model	Accuracy	Recall	Specificity	Precision	F-Score	% Positive	% Negative	# Unknown
SmolLM2-135M Zero-Shot	0.50	0.01	1.00	0.78	0.02	00.78%	99.22%	2993
SmolLM2-360M Zero-Shot	0.56	1.00	0.11	0.53	0.69	94.26%	05.74%	2
SmolLM2-1.7B Zero-Shot	0.72	0.99	0.45	0.65	0.78	77.01%	22.99%	14
SmolLM2-135M Few-Shot	0.59	0.80	0.37	0.56	0.66	71.86%	28.14%	0
SmolLM2-360M Few-Shot	0.66	0.99	0.33	0.60	0.75	82.74%	17.26%	0
SmolLM2-1.7B Few-Shot	0.81	0.99	0.62	0.73	0.84	68.48%	31.52%	0

Table 2. Default SmolLM2 Model's Performance Metrics on IMDb Test Dataset

References

[1] HuggingFace. Smollm - blazingly fast and remarkably powerful. https://huggingface.co/blog/smollm, 2024. 1, 2

[2] HuggingFace. Smollm2. https://huggingface.co/collections/HuggingFaceTB/smollm2-6723884218bcda64b34d7db9, 2024. State-of-the-art compact LLMs for on-device applications: 1.7B, 360M, 135M. 1

[3] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. 1, 2

6. A. Project Code Repository

The Github repository containing the experiments mentioned throughout the report alongside the default and tuned models can be found here.

Model	Total Inference Time (s)	Average Inference Time (s)
SmolLM2-135M Zero-Shot	464.59	0.02
SmolLM2-360M Zero-Shot	499.90	0.02
SmolLM2-1.7B Zero-Shot	342.76	0.01
SmolLM2-135M Few-Shot	342.76	0.02
SmolLM2-360M Few-Shot	490.53	0.02
SmolLM2-1.7B Few-Shot	333.55	0.01

Table 3. Default SmolLM2 Model's Inference Times on Test Dataset

Model	Accuracy	Recall	Specificity	Precision	F-Score	% Positive	% Negative	# Unknown
SmolLM2-135M Zero-Shot	0.50	0.01	1.00	0.78	0.02	00.78%	99.22%	2993
SmolLM2-360M Zero-Shot	0.56	1.00	0.11	0.53	0.69	94.26%	05.74%	2
SmolLM2-1.7B Zero-Shot	0.72	0.99	0.45	0.65	0.78	77.01%	22.99%	14
SmolLM2-135M Few-Shot	0.59	0.80	0.37	0.56	0.66	71.86%	28.14%	0
SmolLM2-360M Few-Shot	0.66	0.99	0.33	0.60	0.75	82.74%	17.26%	0
SmolLM2-1.7B Few-Shot	0.81	0.99	0.62	0.73	0.84	68.48%	31.52%	0

Table 4. Large Model's Performance on IMDb Dataset

Student Name	Contributed Aspects	Details
Jaden Zwicker	Default Model Experimentation & Analysis	Performed various experiments on the 3 SmolLM2 models in their default configuration. Analyzed the results of these experiments to compare and contrast them with the larger LLMs and to set a baseline of performance before fine tuning.
Houshmand Abbaszadeh	Analysis & Comparison of Various ML Techniques	Trained the LSTM of the encoder and analyzed the results. Analyzed effect of number of nodes in hidden state. Implemented Convolutional LSTM.
Team Member 3	Implementation and Analysis	Trained the LSTM of the encoder and analyzed the results. Analyzed effect of number of nodes in hidden state. Implemented Convolutional LSTM.

Table 5. Contributions of team members.