Small Language Models are Good Too: An Empirical Study of Zero-Shot Classification

Pierre Lepagnol[1,2], **Thomas Gerald**[1], **Sahar Ghannay**[1], **Christophe Servan**[1,3], **Sophie Rosset**[1]
[1]Université Paris-Saclay, CNRS, LISN, [2]SCIAM, [3]QWANT
{firstname.lastname}@lisn.upsaclay.fr

Abstract

This study is part of the debate on the efficiency of large versus small language models for text classification by prompting. We assess the performance of small language models in zero-shot text classification, challenging the prevailing dominance of large models. Across 15 datasets, our investigation benchmarks language models from 77M to 40B parameters using different architectures and scoring functions. Our findings reveal that small models can effectively classify texts, getting on par with or surpassing their larger counterparts. We developed and shared a comprehensive open-source repository that encapsulates our methodologies. This research underscores the notion that bigger isn't always better, suggesting that resource-efficient small models may offer viable solutions for specific data classification challenges.

Keywords: Zero-shot, Prompting, language modeling, LLMs, data labeling

1. Introduction

Large Language Models (LLMs) have been massively favored over smaller models to solve tasks through prompting (Brown et al., 2020; Hoffmann et al., 2022; OpenAI, 2023; Chowdhery et al., 2022) in a zero-shot setting. However, while their utility is extensive, they come with challenges - they are resource-intensive, costly to employ, and their performances are not always warranted for every task (Nityasya et al., 2021). Bigger models (Kaplan et al., 2020; Hoffmann et al., 2022) were built, always sophisticated datasets were necessary (Zhang et al., 2023) to achieve claimed performances. Their perceived superior performance has typically made them the go-to choice for various tasks, even basic classification problems.

An application of LLMs is the generation of pseudo-labels through zero-shot prompting, a method often employed to construct labeled datasets (Smith et al., 2022). As the field advances, we must ask: are large language models essential for effective data classification?

This study examines how well small models can match big models in creating labels using different datasets. We want to see how small models perform in this zero-shot text classification and determine what makes them do well with specific data. We are comparing how small and big models work with zero-shot prompting on various data sets to understand if we can get good results with less resources.

We believe this research is the beginning of understanding the true capabilities of LLMs when prompted for zero-shot classification tasks.

Our main contributions are:

1. We benchmark a large scale of language models (up to 70b parameters) fine-tuned on instructions-following datasets, with different architectures(encoder-decoder or decoder only) and sizes on many datasets in a zero-shot setting.

2. We provide relatively strong evidence of the effectiveness of small models in zero-shot classification, and we show that the performances of small models are comparable to those of large models on many datasets in classification problems.

3. We present a fully open-source repository encapsulating our proposed methodologies, thereby contributing to the integrity and robustness of research in this field. The code is available online in this repository.

The paper is organized as follows: Section 2 provides a literature review on related zero-shot approaches. Then, in Section 3, we describe the methodology we follow for this study. Section 4 presents the consequences given the different analyses. Finally, we conclude in Section 5 and discuss future work and research directions.

2. Related Work

The domain of zero-shot classification has previously been explored. These studies offer valuable insights and set the stage for our investigations.

2.1. Zero-Shot Text Classification & Prompting

General zero-shot text classification aims to categorize texts into classes not part of the training dataset. It has caught the attention of many researchers because it removed the need for extra fine-tuning steps and labeled datasets. To effectively transfer knowledge from seen classes to unseen ones, there's a need for precise and distinguishing class descriptions, as noted by Xia et al. (2018) and Liu et al. (2019a). Yet, these approaches depend on supervised data from recognized labels, which renders them unsuitable when there's a complete absence of labeled data for any given category.

Fei et al. (2022) enhances zero-shot classification by segmenting input texts and leveraging class-specific prompts. While Meng et al. (2020) proposed a strategy that employs label names combined with self-training tailored for zero-shot classification. Many methods necessitate an unlabeled dataset or a knowledge base to extract pertinent topic words and facilitate self-training. More recently, Zhao et al. (2023b) proposed to use k-Nearest-Neighbor on embeddings similarity to augment their verbalizers. Lu et al. (2023) proposed Perplexity Selection to select the best prompts in a zero-shot setting.

Discussion

While previous work focused on new methods to make language models better zero-shot learners, we want insight into model features and how well they perform.

3. Experimental Setup

Although authors of LLMs have compared their different model sizes(Kaplan et al., 2020; Hoffmann et al., 2022), this study widens this analysis by directly comparing different architectures on an extensive set of datasets. We prompt various language models using 4 different scoring functions (see Section 3.4.2) to classify sentences and report accuracy and F1 scores for each triple model-datasets-scoring function.

3.1. Tasks & Datasets

We examine a diverse set of 15 datasets, curated to represent a broad spectrum of classification challenges. We draw from datasets like *AG-News*, with its 4 distinct classes, and *BBCNews*, offering 5 unique categories for topic classification. Sentiment classification is represented through binary choices like in *ethos* (Mollas et al., 2022) and more granular datasets like *sst-5* (Socher et al., 2013). Standard Spam classification tasks such as *youtube* comments (Alberto et al., 2015) or *sms* (Almeida and Hidalgo, 2012) are included. Relation classification tasks are also included using datasets like *semeval* (Hendrickx et al., 2010).

The balance ratios across our chosen datasets varied extensively, from the perfectly balanced *imdb* to those displaying significant imbalances like *chemprot* (Krallinger et al., 2017).

The complete list will be given in the final version.

3.2. Metrics

We distinguish datasets on whether they are balanced using the balance ratio *i.e.* the ratio between the majority class and the minority class. The accuracy (acc) is used to evaluate binary tasks and balanced datasets, while the macro f1 (f1) score is used for the other tasks.

3.3. Models

Our study assesses a total of 72 unique models. We select both encoder-decoder models (like `T5` (Raffel et al., 2020), `mT0` (Muennighoff et al., 2023), and Bart (Lewis et al., 2020)) and causal-decoder-only models (such as `Llama` (Touvron et al., 2023) and `Falcon` (Penedo et al., 2023)). We opt for various sizes for the same models, ranging from 77 million to hundreds of 40 billion parameters. We called small language models, models within the size range 77M to 3B parameters. These models are comparatively smaller, ranging from 13 to 156 times less in parameter count than our largest model, `Falcon 40B`[1]. Moreover, at the time our study was conducted, `TinyStories` (Eldan and Li, 2023) models, which are on an even smaller scale, starting at 1M parameters.

These models were chosen based on their prevalence in literature, reported efficacy on similar tasks, and the fact that instruction-tuned versions were available for some of them.

Instruction-tuning refers to the strategy for fine-tuning a language model on instruction datasets (Longpre et al., 2023).

The complete list will is given in appendix A.

3.4. Prompts & Scoring Functions

This section sets our research's specific prompts and scoring functions. We follow (Brown et al., 2020) to craft simple prompts while ensuring domain relevance. Additionally, we explore various scoring functions, assessing their impact on our models' performance.

3.4.1. Prompts

Our experiments' prompts are hand-crafted and designed to be simple and straightforward.

[1] We do not test `Falcon 180B`, as it was not released during our experiments

Prompts are either translated from the code-based labeling functions provided by the WRENCH benchmark (Zhang et al., 2021) or created from scratch. They are tailored for each task, *e.g.* prompts for the *healthcare* dataset are framed differently from those for the financial dataset to ensure domain relevance and to maximize model comprehension.

For example, for the dataset *sms*, the prompt is

Prompt

```
Is the following message spam?
Answer by yes or no.\n"{TEXT}"
```

Verbalizer

```
{1:"yes", 0:"no"}
```

For *bbcnews*, the prompt is

Prompt

```
""{TEXT}" is about "
```

Verbalizer

```
{"0": "tech", "1": "business",
"2": "sport", "3": "entertainment",
"4": "politics"}.
```

3.4.2. Scoring Functions

In prompt-based classification, using a verbalizer mapping tokens to class labels is crucial for accurate classification. As suggested by (Holtzman et al., 2022), many valid sequences can represent the same concept, called *surface form competition*. For example, `"+"`, `"positive"`, `"More positive than the opposite"` could be used to represent the same concept of positivity for the sentiment analysis task. As this competition exists, how verbalizers are designed could either mitigate or exacerbate the effects of surface form competition, thereby influencing the overall effectiveness of the prompt-based classification approach. Zhao et al. (2023b) uses k-Nearest-Neighbor for verbalizer construction and augments their verbalizers based on embeddings similarity.

We use several scoring functions to evaluate the impact of scoring functions on the performances of our models. We describe in plain english these scoring function in appendix C.

3.5. Comparison

We compare our results with Majority Voting (i.e predicting the class of the majority class in the dataset) and state-of-the-art (SOTA) Zero-Shot Learning methods. Table 2 presents the SOTA scores for each dataset[3].

Probability	$\arg\max_i \mathbb{P}(y_i \mid x')$
DCPMI	$\arg\max_i \frac{\mathbb{P}(y_i \mid x')}{\mathbb{P}(y_i \mid x_{\text{domain\_conditional}})}$
PMI	$\arg\max_i \frac{\mathbb{P}(y_i \mid x')}{\mathbb{P}(y_i \mid x_{\text{domain\_unconditional}})}$
Similarity	$\arg\max_{c_i \in C} \cos(e(t_0), e(y_i))$[2]

Table 1: Scoring functions from (Holtzman et al., 2022)

3.6. Tools for Statistical Analysis

For our analysis, we make use of three main statistical tools, detailed below:

The Biweight Midcorrelation Coefficient is a robust alternative to Pearson's correlation coefficient to quantify the strength of association between two samples. It is designed to be less sensitive to outliers than other correlation coefficients like Pearson's correlation.

Analysis of Covariance - ANCOVA combines the techniques of ANOVA and regression to evaluate whether the means of a dependent variable are equal across levels of a categorical independent variable while statistically controlling for the effects of other continuous variables (covariates).

Kruskal-Wallis Test is a non-parametric method to test whether samples originate from the same distribution. We used it as a non-parametric method, which does not assume a normal distribution of the residuals, unlike the analogous standard one-way analysis of variance.

4. Results

We compare the performance of the LLM models on several datasets, studying the correlation with the number of parameters, the impact of the architecture, and the type of training strategy (instruction or not). Then, for the two types of architectures (encoder-decoder & decoder-only), we study the impact of the instruction-tuning and the different scoring functions to understand the discriminating factors on performance.

[3] We removed scores from the `mT0` model for some datasets (*agnews*, *imdb*, *yelp*,*trec*) because these models were trained on those datasets.

dataset	SOTA Scores	Majority Class - Scores	Best Score	Model Used	Number of parameters
agnews	0.625	0.266	**0.734**	MBZUAI/LaMini-GPT-124M	163.0 Millions
bbcnews	NaN	0.236	0.869	bigscience/mt0-large	1.2 Billions
cdr	NaN	0.676	0.717	bigscience/bloomz-3b	3.6 Billions
chemprot	0.172	0.049	**0.192**	bigscience/bloomz-3b	3.6 Billions
ethos	0.667	0.566	0.597	bigscience/bloomz-1b1	1.5 Billions
financial_phrasebank	0.528	0.254	**0.744**	MBZUAI/LaMini-GPT-774M	838.4 Millions
imdb	0.718	0.500	**0.933**	MBZUAI/LaMini-Flan-T5-783M	783.2 Millions
semeval	0.435	0.054	0.270	bigscience/mt0-xxl	12.9 Billions
sms	0.340	0.464	**0.699**	mosaicml/mpt-7b	6.6 Billions
spouse	0.630	0.479	0.521	gpt2	163.0 Millions
sst-2	0.710	0.501	**0.956**	bigscience/bloomz-3b	3.6 Billions
sst-5	0.598	0.286	0.485	tiiuae/falcon-40b-instruct	41.8 Billions
trec	NaN	0.072	0.324	mosaicml/mpt-7b-instruct	6.6 Billions
yelp	0.888	0.522	**0.977**	MBZUAI/LaMini-Flan-T5-783M	783.2 Millions
youtube	0.468	0.528	**0.716**	tiiuae/falcon-40b	41.8 Billions

Table 2: Table illustrating the performance metrics across various datasets:
Columns present (1) the dataset name, (2) the reported state-of-the-art (SOTA) scores, (3) scores obtained when predicting the majority class, (4) the highest achieved scores (highlighted in red), (5) the model architectures associated with these top scores, and (6) the number of parameters for each respective model. Note the presence of NaN entries, signifying datasets where SOTA benchmarks have not been established or found.

4.1. Data-based Analysis

For the dataset-based analysis, we propose to study: 1) the relationship between the task performances and the model sizes (the number of parameters), 2) the task performances and the architecture, and 3) whether the model was fine-tuned on instruction datasets.

4.1.1. Model size doesn’t really matter

Figure 1 presents the relationship between the number of parameters and the performance in terms of Acc/F1 scores across various datasets. The correlations observed range from positive and negative to zero.

To further understand these correlations, we calculate the Biweight Midcorrelation Coefficient and associated p-values for each dataset. These findings are detailed in Table 3.

From our analysis, 10 of 15 datasets show p-values exceeding 0.05, suggesting no significant link between Acc/F1 scores and model size. However, three datasets exhibit p-values below 0.05, indicating a notable correlation. Of these, the direction of correlation is positive for the *cdr* dataset but negative for both *ethos* and *imdb* datasets. Two datasets, namely *agnews* and *chemprot*, present p-values near the 0.05 threshold, making their correlation inconclusive.

In conclusion, while many datasets do not show a direct relationship between larger model sizes and improved performance, datasets like *cdr*, *ethos*, and *imdb* do. Overall, the variance in the correlation coefficient across datasets suggests that model size isn’t the sole determinant of performance.

dataset	correlation coef	pvalue
agnews	-0.1418	0.0536
bbcnews	0.0489	0.4877
cdr	0.2541	**0.0002**
chemprot	0.1318	0.0531
ethos	-0.1519	**0.0256**
financial_phrasebank	0.0419	0.5406
imdb	-0.2862	**0.0001**
semeval	-0.0506	0.4595
sms	-0.1209	0.0763
spouse	-0.0254	0.7106
sst-2	0.0755	0.2693
sst-5	0.0061	0.9293
trec	-0.1085	0.1403
yelp	-0.0620	0.4008
youtube	-0.0014	0.9836

Table 3: The Biweight Midcorrelation Coefficients and P-values Indicating the Relationship Between Acc/F1 and Model Size (Log-Number of Parameters) Across Datasets

4.1.2. Impact of Architectural Choices on Performance

Figure 2 illustrates the performance variations between encoder-decoder and decoder-only architectures.

Using ANCOVA, we measure the impact of the architecture choice on Acc/F1 scores, while controlling the effect of the model size variable. The results are presented in Table 4. On one hand, 7 out of 15 datasets, namely *agnews*, *bbcnews*, *chemprot*, *semeval*, *sms*, *spouse*, and *youtube*, show p-values bellow 0.05, suggesting there the architecture has a significant impact.

On the other hand, datasets such as *cdr*, *ethos*, and *financial_phrasebank* remain unaffected by

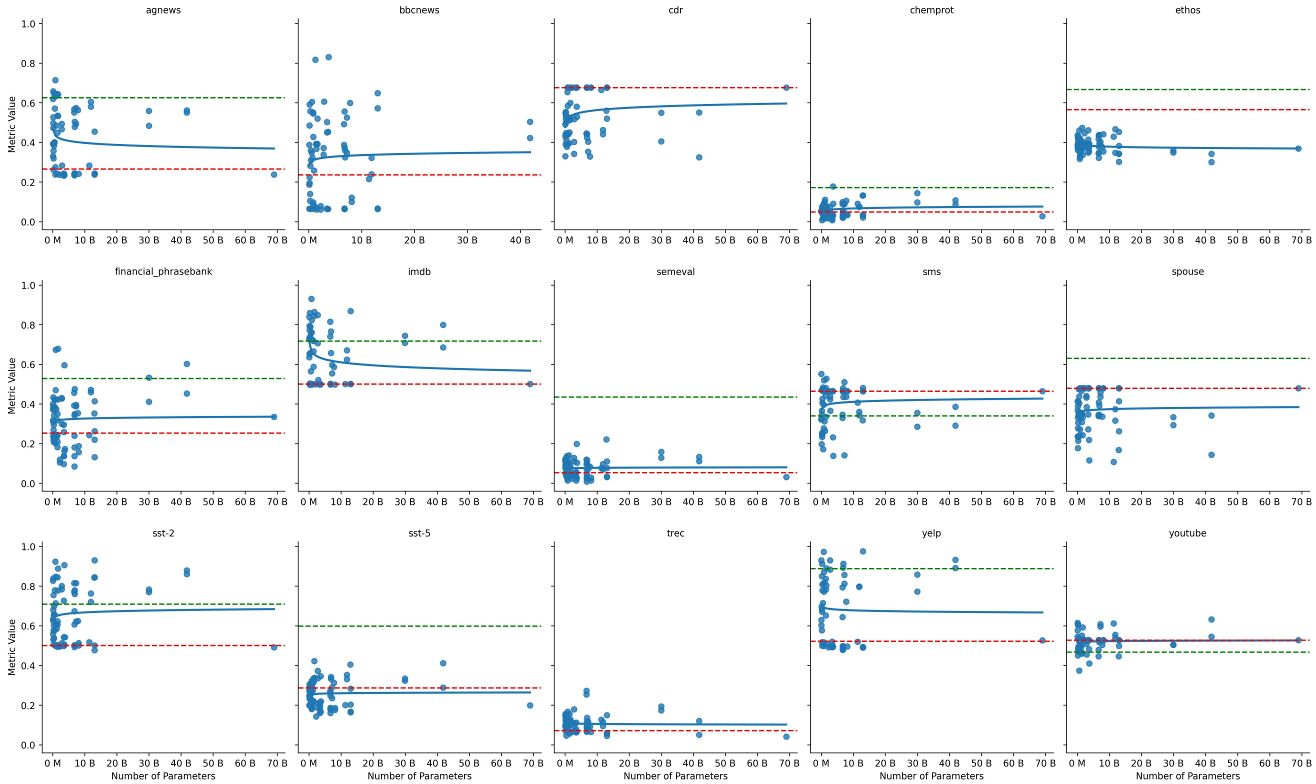

Figure 1: Performance Comparison of Different Model Sizes Across Datasets.
4

Dataset	Statistic	Pvalue	Equal Variances
agnews	4.0676	**0.0452**	True
bbcnews	7.0640	**0.0085**	False
cdr	0.2519	0.6163	True
chemprot	4.4883	**0.0353**	True
ethos	0.3945	0.5306	False
financial_phrasebank	1.4592	0.2284	False
imdb	3.6687	0.0570	True
semeval	8.2301	**0.0045**	True
sms	11.9951	**0.0006**	False
spouse	4.7794	**0.0299**	True
sst-2	0.2501	0.6175	True
sst-5	0.7852	0.3766	True
trec	0.3382	0.5616	False
yelp	0.7103	0.4004	True
youtube	18.0011	**0.0000**	False

Table 4: ANCOVA Indicating the Impact of Architectures on Acc/F1 Across Datasets

the architectural choice. The *imdb* dataset demonstrates a borderline significance.

In conclusion, while the model size might not be a dominant factor, the architectural choice significantly impacts performance across specific datasets.

4.1.3. Impact of Instruction fine-tuning on performance

In the same way as architecture, we quantified the impact of instruction-tuning on performances while controlling the number of parameters.

Figure 3 visually compares the impact of instruction-tuning and performance metrics (Acc/F1) across various datasets.

The y-axis of each graph displays the performance metric (Acc/F1). The x-axis has two values: `False` and `True`, indicating whether instruction fine-tuning is applied to the model.

We use ANCOVA to test whether the means of our ACC/F1 scores are equal across modalities of instruction tuning while statistically controlling the effect of the number of parameters.

dataset	statistic	pvalue	Equal Variances
agnews	10.5411	**0.0014**	True
bbcnews	1.9492	0.1642	True
cdr	0.1635	0.6864	True
chemprot	2.3152	0.1296	True
ethos	5.8015	**0.0169**	True
financial_phrasebank	0.0001	0.9917	False
imdb	13.6945	**0.0003**	True
semeval	1.4016	0.2378	False
sms	2.6667	0.1039	True
spouse	0.3379	0.5617	True
sst-2	3.0055	0.0844	False
sst-5	1.8271	0.1779	True
trec	8.3534	**0.0043**	False
yelp	12.5571	**0.0005**	True
youtube	5.8369	**0.0165**	True

Table 5: ANCOVA Indicating the Impact of The instruction Fine-Tunning on Acc/F1 Across Datasets

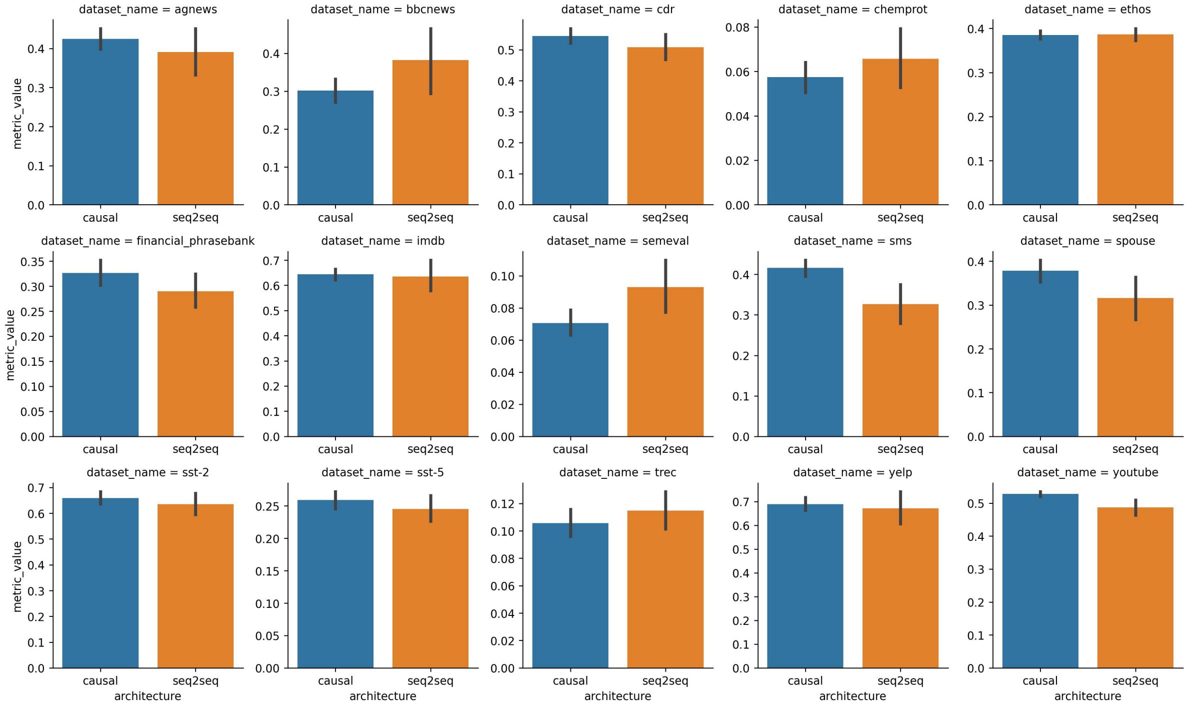

Figure 2: Performance Variation Across Different Architectures.

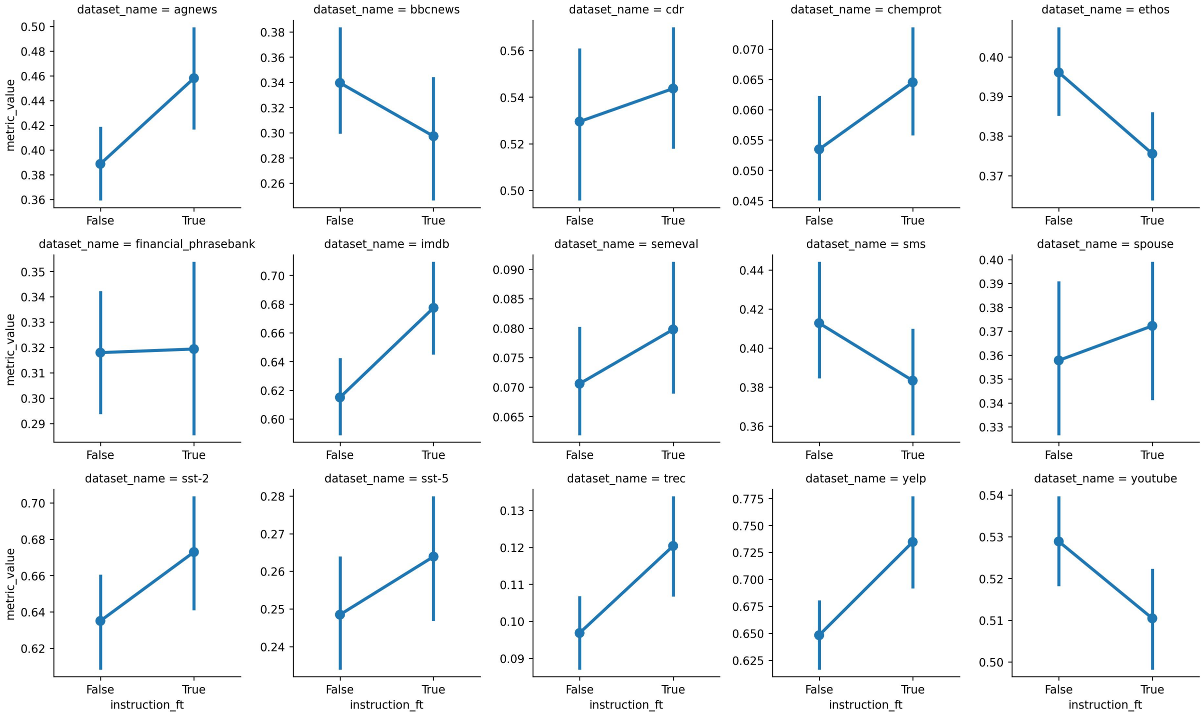

Figure 3: Performance Comparison between Instruction-Tuned models or not Across Datasets.

For many datasets, instruction fine-tuning improves performances when compared to not fine-tuning (*e.g.*, *agnews*, *ethos*, *imdb*, *trec*, *yelp*, and *youtube*). This is evident from the graphical representation and the significant p-values from the ANCOVA. Datasets like *bbcnews*, *youtube*, and *sms* show a decrease in performance when instruction fine-tuning is applied, but ANCOVA tells us that it is not significant. While for *ethos*, it is significant.

For other datasets, while there might be visual differences in performance with and without instruction fine-tuning, these differences aren't statistically significant based on the p-values.

Therefore, while instruction fine-tuning has the potential to enhance model performance on many datasets, its impact may vary depending on the specific dataset.

4.2. Architecture-based Analysis

In our analysis, we shift our attention to which features among the model size, instruction-tuning, and scoring functions have an impact on performance.

4.2.1. relationship between model size and performances per architecture

Table 6 presents The Biweight Midcorrelation Coefficients between the model sizes (log-number of parameters) and performance metrics (Acc/F1) for either encoder-decoder and decoder-only.

dataset	correlation coef	pvalue
causal	-0.0435	**0.0299**
seq2seq	0.0065	0.8728

Table 6: The Biweight Midcorrelation Coefficients and P-values Indicating the Relationship Between Acc/F1 and Model Size (Log-Number of Parameters) Across Architectures

Table 6 shows a slight but significant correlation for decoder models but largely insignificant for encoder-decoder ones.

This suggests that decoder-only could be more sensitive to the number of parameters; too many parameters could harm performance.

4.2.2. Impact of Instruction Fine-tuning and Performances per architecture

Figure 4 visually compares the impact of instruction-tuning and performance metrics (Acc/F1) for the two architectures.

The y-axis is the performance metric (Acc/F1). The x-axis has two values: `False` and `True`, indicating whether instruction fine-tuning is applied to the model.

An ANCOVA is made to quantify the impact of instruction-tuning on each architecture (encoder-decoder/decoder-only) while statistically controlling for the effect of the model size feature. Table 7 reports statistics and p-values.

dataset	statistic	pvalue	Equal Variances
causal	0.1825	0.6693	True
seq2seq	6.9406	**0.0086**	False

Table 7: ANCOVA Indicating the Impact of instruction_ft on Acc/F1 Across Architectures.

For the causal architecture, there is no significant impact of instruction-tuning on Acc/F1 scores. The p-value for the decoder-only architecture is 0.6693, much greater than 0.05. For the seq2seq architecture, there is a significant impact of instruction tuning on Acc/F1 scores. The p-value for the encoder-decoder architecture is highlighted in red as 0.0086, less than 0.05. Additionally, the variances between the groups for seq2seq are not equal.

The difference in results between the two architectures suggests that the impact of instruction-tuning might be architecture-dependent. Both the graphical analysis and the ANCOVA show an effect of instruction-tuning on encoder-decoder architecture.

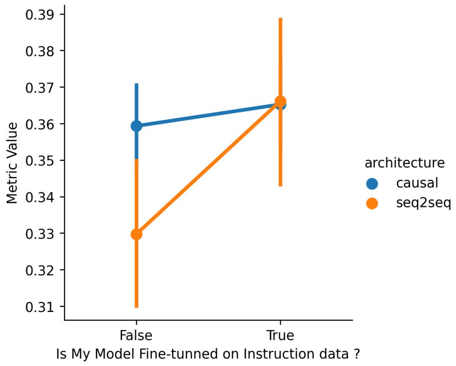

Figure 4: Performance Comparison between Instruction-Tuned models or not, Across Model Architecture

4.2.3. Impact of Scoring Functions and Performances per architecture

Table 8 reports the ANCOVA results of the impact of different scoring functions on performances for the two architectures.

Architecture	statistic	pvalue	Equal Variances
causal	0.6711	0.5113	False
seq2seq	0.5003	0.6066	True

Table 8: ANCOVA Indicating the Impact of Scoring Functions on Acc/F1 Across Architectures.

For both encoder-decoder and decoder-only models, values are above the standard 0.05 by a large margin. This suggests no significant impact on the choice of scoring functions.

To sum it up, no matter which model architecture we look at, the choice of scoring function doesn't seem to affect more than another.

5. Conclusion & Perspectives

This paper aimed to understand better whether we need large models to tackle classification problems through prompting.

The performance of LLM models varies based on multiple factors, including model size, architectural choices, and fine-tuning strategies. While larger model sizes do not consistently lead to improved performance across all datasets, the architectural choice significantly influences outcomes on specific datasets. The impact of instruction fine-tuning is also evident, but its efficacy is dependent on the architecture. Notably, the choice of scoring function doesn't seem to make a marked difference in performance.

A comprehensive study of other emerging architectures, such as RWKV architecture (Peng et al., 2023) or Retentive Network (Sun et al., 2023), could bring nuances and detail to this analysis. The varied impact of instruction fine-tuning across datasets suggests the need for more advanced fine-tuning techniques like incorporating information retrieval during fine-tuning to ensure even better classification performances during zero-shot prompting.

6. Limitations

We limit this evaluation to simple prompting methods and hand-crafted, unoptimized prompts. We also provide a single prompt for each dataset.

We focused on causal-decoder-only and encoder-decoder models without comparing them with encoder-only or non-causal decoders as recently released models focused on those architectures.

We did not mention external factors such as pretraining time, data quality, or potential biases in the datasets. These external factors might impact the results or the generalizability of the conclusions.

The choice and assumptions of the statistical tools could influence the results. There might be newer or specialized models not included in this study, which could exhibit different behaviors.

7. Acknowledgements

This work is supported by the ANRT (Association nationale de la recherche et de la technologie) with a CIFRE fellowship granted to SCIAM[5]. (CIFRE N°2022/1608)

This work was performed using HPC resources from GENCI–IDRIS (Grant 2023-AD011014242).

Ethics Statement

It is worth noting that the behavior of our downstream models is subject to biases inherited from the dataset it was trained, as no alignment nor specific filtering was done. We envision the same research progress in reducing anti-social behaviors in LLMs can also be applied to improve smaller language models.

8. Bibliographical References

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. Technical report. ArXiv:2304.01373 [cs] type: article.

BigScience Workshop. 2022. BLOOM (revision 4ab0472).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David

[5] https://www.sciam.fr/

Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. Technical report. ArXiv:2204.02311 [cs] type: article.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. Technical report. ArXiv:2210.11416 [cs] type: article.

Nolan Dey, Gurpreet Gosal, Zhiming, Chen, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, and Joel Hestness. 2023. Cerebras-GPT: Open Compute-Optimal Language Models Trained on the Cerebras Wafer-Scale Cluster. Technical report. ArXiv:2304.03208 [cs] type: article.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english?

Yu Fei, Zhao Meng, Ping Nie, Roger Wattenhofer, and Mrinmaya Sachan. 2022. Beyond prompting: Making Pre-trained Language Models Better Zero-shot Learners by Clustering Representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8560–8579, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training Compute-Optimal Large Language Models. Technical report. ArXiv:2203.15556 [cs] type: article.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2022. Surface Form Competition: Why the Highest Probability Answer Isn't Always Right. Technical report. ArXiv:2104.08315 [cs] type: article.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes. Technical report. ArXiv:2305.02301 [cs] type: article.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. Technical report. ArXiv:2001.08361 [cs, stat] type: article.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y.S. Lam. 2019a. Reconstructing Capsule Networks for Zero-shot Intent Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4799–4809, Hong Kong, China. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. Technical report. ArXiv:2107.13586 [cs] type: article.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned LLaMA Outperforms GPT-4 on Arithmetic Tasks. Technical report. ArXiv:2305.14201 [cs] type: article.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. Technical report. ArXiv:1711.05101 [cs, math] version: 3 type: article.

Jinghui Lu, Dongsheng Zhu, Weidong Han, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023. What makes pre-trained language models better zero-shot learners? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2288–2303, Toronto, Canada. Association for Computational Linguistics.

Josh Magnus Ludan, Yixuan Meng, Tai Nguyen, Saurabh Shah, Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2023. Explanation-based Finetuning Makes Models More Robust to Spurious Cues. Technical report. ArXiv:2305.04990 [cs] version: 2 type: article.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text Classification Using Label Names Only: A Language Model Self-Training Approach. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9006–9017, Online. Association for Computational Linguistics.

Alejandro Mosquera. 2022. Tackling Data Drift with Adversarial Validation: An Application for German Text Complexity Estimation. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*, pages 39–44, Potsdam, Germany. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual Generalization through Multitask Finetuning. Technical report. ArXiv:2211.01786 [cs] type: article.

Made Nindyatama Nityasya, Haryo Akbarianto Wibowo, Radityo Eko Prasojo, and Alham Fikri Aji. 2021. Costs to Consider in Adopting NLP for Your Business. Technical report. ArXiv:2012.08958 [cs] type: article.

OpenAI. 2023. GPT-4 Technical Report. Technical report. ArXiv:2303.08774 [cs] type: article.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. Technical report. ArXiv:2203.02155 [cs] type: article.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. Technical report. ArXiv:2306.01116 [cs] type: article.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Xiangru Tang, Bolun Wang, Johan S. Wind, Stansilaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. Rwkv: Reinventing rnns for the transformer era.

Ildikó Pilán and Elena Volodina. 2018. Investigating the importance of linguistic complexity features across different datasets related to language learning. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 49–58, Santa Fe, New-Mexico. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a General-Purpose Natural Language Processing Task Solver? Technical report. ArXiv:2302.06476 [cs] type: article.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Technical report. ArXiv:1910.10683 [cs, stat] type: article.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. Technical report. ArXiv:2110.08207 [cs] type: article.

Timo Schick and Hinrich Schütze. 2021. Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. Technical report. ArXiv:2001.07676 [cs] type: article.

Ryan Smith, Jason A. Fries, Braden Hancock, and Stephen H. Bach. 2022. Language Models in the Loop: Incorporating Prompting into Weak Supervision. Technical report. ArXiv:2205.02318 [cs] type: article.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying Language Learning Paradigms. Technical report. ArXiv:2205.05131 [cs] type: article.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization? Technical report. ArXiv:2204.05832 [cs, stat] type: article.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2023. PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization. Technical report. ArXiv:2306.05087 [cs] type: article.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models are Zero-Shot Learners.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. 2023. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *CoRR*, abs/2304.14402.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip S. Yu. 2018. Zero-shot User Intent Detection via Capsule Neural Networks. Technical report. ArXiv:1809.00385 [cs] type: article.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction Tuning for Large Language Models: A Survey. Technical report. ArXiv:2308.10792 [cs] type: article.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023a. A Survey of Large Language Models. Technical report. ArXiv:2303.18223 [cs] type: article.

Xuandong Zhao, Siqi Ouyang, Zhiguo Yu, Ming Wu, and Lei Li. 2023b. Pre-trained language models can be fully zero-shot learners. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15590–15606, Toronto, Canada. Association for Computational Linguistics.

9. Language Resource References

Tulio C. Alberto, Johannes V. Lochter, and Tiago A. Almeida. 2015. TubeSpam: Comment Spam Filtering on YouTube. *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 138–143.

Tiago Almeida and Jos Hidalgo. 2012. SMS Spam Collection. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5CC84.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Martin Krallinger, O. Rabal, S. Akhondi, M. Pérez, J. Santamaría, Gael Pérez Rodríguez, G. Tsatsaronis, Ander Intxaurrondo, José Antonio Baso López, U. Nandal, E. V. Buel, A. Chandrasekhar, Marleen Rodenburg, A. Lægreid, Marius A. Doornenbal, J. Oyarzábal, A. Lourenço, and A. Valencia. 2017. Overview of the BioCreative VI chemical-protein interaction Track.

Ioannis Mollas, Zoe Chrysopoulou, Stamatis Karlos, and Grigorios Tsoumakas. 2022. ETHOS: an Online Hate Speech Detection Dataset. *Complex & Intelligent Systems*, 8(6):4663–4678. ArXiv:2006.08328 [cs, stat].

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander J. Ratner. 2021. WRENCH: A Comprehensive Benchmark for Weak Supervision. *ArXiv*.

A. Models

Table 9 presents decoder-only models used for classification. Table 10 presents encoder-decoder models used for classification. Column `Model` contains the name of each model on their HuggingFace repository, column `Number of Parameters` and `Instruction-Tuned` are quite explicit.

Note that *M* stands for million and *B* for billion.

B. Datasets

C. Prompts & Scoring functions

The first is the probability of the label given the prompt, it is the most straightforward method, giving the probability of the continuation. The second and third methods are the ratio between this probability and the probability of the label given a ”tasks specific premise” (called DCPMI) and an ”unconditional/not task specific premise”. These methods are a reweighting of each label options according to its a priori likelihood in/out of the context of the task. The fourth is cosine similarity, wich gives a measure of similarity between the embedding of the predicted token and the label. Th intuition behind this method is that a performant model should output a token similar to the label.

As we noticed difference in classification performances under different scoring functions but none could lead to a clear winner, could'nt juge really how well models performed. So we decided to take the mean of these scores to have a more robust evaluation of the model's performance.

Model	Number of Parameters	Instruction-Tuned
bigscience/bloom (**?**)	560M, 1B1, 1B7, 3B, 7B1	No
bigscience/bloomz (Muennighoff et al., 2023)	560M, 1B1, 1B7, 3B, 7B1	Yes
tiiuae/falcon	7B, 40B	Yes/No
tiiuae/falcon-rw	7B, 40B	No
MBZUAI/LaMini-Cerebras (Wu et al., 2023)	111M, 256M, 590M, 1.3B	Yes
MBZUAI/LaMini-GPT (Wu et al., 2023)	124M, 774M, 1.5B	Yes
mosaicml/mpt	7B 30b	Yes/No
databricks/dolly-v2	3b, 7B, 12b	Yes
EleutherAI/pythia (Biderman et al., 2023)	70M, 160M, 410M, 1B, 1.4B, 2.8, 6.9B, 12B	No
openlm-research/open_llama	3B 7B 13B	No
openlm-research/open_llama_v2	3B 7B	No
pankajmathur/orca_dolly	3B	Yes
pankajmathur/orca_alpaca	3B	Yes
pankajmathur/orca_mini	7B, 3B, 13B	Yes
pankajmathur/orca_mini_v2	7B, 13B	Yes
pankajmathur/orca_mini_v3	7B, 13B	Yes

Table 9: Decoder Only Models

Model	Number of Parameters	Instruction-Tuned
MBZUAI/LaMini-Flan-T5 (Wu et al., 2023)	77M, 248M, 783M	Yes
T5 vanilla (Raffel et al., 2020)	77M, 248M, 770M, 3B, 11B	No
bigscience/mt0 (Muennighoff et al., 2023)	300M, 582, 1.2B, 3.8B, 13B	Yes
Bart (Lewis et al., 2020)	255M, 561M	No

Table 10: Encoder-Decoder Only Models

Datasets	Tasks	#Classes	#Test Examples	Balance ratios
AGNews	Topic Classification	4	12000	0.897
BBCNews	Topic Classification	5	2000	0.742
CDR bio	Relation Classification	2	4673	0.478
Chemprot	Chemical Relation Classification	10	1607	0.004
ETHOS	Sentiment Classification	2	998	0.766
financial_phrasebank	Topic Classification	3	2264	0.218
IMDB	Sentiment Classification	2	2500	1.000
SemEval	Relation Classification	9	600	0.042
SMS	Spam Classification	2	500	0.155
Spouse	Relation Classification	2	2701	0.088
SST2	Sentiment Classification	2	1821	0.997
SST5	Sentiment Classification	5	2210	0.441
TREC	Question Classification	6	500	0.065
Yelp	Sentiment Classification	2	3800	0.915
Youtube	Spam Classification	2	250	0.894

Table 11: Datasets Descriptions

Table 12: Prompt used

dataset	prompt	pmi_premise	dcpmi_premise
sms	Is the following message spam? Answer by yes or no.\n"TEXT"	:	The message is a spam ?
youtube	Is the following comment spam? Answer by yes or no.\n"TEXT"	:	The comment is a spam ?
spouse	Context: "TEXT"\n\nAre ENTITY2 and ENTITY1 married? Answer by yes or no.	:	Are the two entity are married?
cdr	Context: "TEXT"\n\nDoes ENTITY1 induce ENTITY2 ? Answer by yes or no.	:	Does the drug induce the disease?
chemprot	Context: "TEXT"\n\nWhat is the relation between ENTITY1 and ENTITY2 ?	:	What is the relation between the two entities?
semeval	Context: "TEXT"\n\nWhat is the relation between ENTITY1 and ENTITY2 ?	:	What is the relation between the two entities?
sst-2	"TEXT" has a tone that is	:	The quote has a tone that is
sst-5	"TEXT" has a tone that is	:	The quote has a tone that is
yelp	"TEXT" has a tone that is	:	The quote has a tone that is
imdb	"TEXT" has a tone that is	:	The quote has a tone that is
ethos	"TEXT" has a tone that is	:	The quote has a tone that is
financial_phrasebank	"TEXT" has a tone that is	:	The quote has a tone that is
trec	"TEXT" is about	:	The topic is
agnews	"TEXT" is about	:	The topic is
bbcnews	"TEXT" is about	:	The topic is