NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Sentiment Analysis with Sliced Transformer Models for Domain Adaptation on Small Datasets

SC4001 Assignment

By: Jain Yash (U2222568A)
Li Luyun (U2120326J)

Code:
https://github.com/YashJain14/Gemma_TSA

Runs:
https://wandb.ai/yashjain14-nanyang-technological-university-singapore

Introduction

Sentiment analysis is a core application in natural language processing (NLP) and involves wide-ranging applications, from analyzing customer reviews to sentiment tracking across social media platforms. Sentiment classification methods used in the past were traditionally reliant on feature engineering and lexicons, yet the last few years have increasingly moved toward deep learning models that can learn complex representations directly from data.

Pre-trained transformer models like BERT, RoBERTa, and GPT, which were pre-trained on large corpora, enjoy state-of-the-art performance on a variety of sentiment benchmarks [1]. However, there are a number of challenges related to the application of these models in real-world applications [2]:

- **Domain Adaptation**: A model trained on one domain (e.g. movie reviews) may not generalize well to another domain (e.g. restaurant reviews) without adaptation. Domain-specific language and vocabulary can degrade performance if not addressed [3].

- **Architecture Differences**: Transformer architectures vary (encoder-only like RoBERTa vs. decoder-only like LLaMA and Gemma vs. encoder-decoder like T5). It is not obvious which architecture is most effective for sentiment tasks, or how their internal layers contribute to classification [5].

- **Limitations of Small Datasets**: Many sentiment corpora in real-world applications are relatively small, consisting of tens of thousands of examples or less, and thus risk overfitting when used to fine-tune larger models. Fine-tuning a large model on a small set of examples can lead to unstable training or require careful regularization [5].

This challenge leads to a question about whether fine-tuning is even necessary, or if pre-trained features could be used effectively with little or no additional training.
In the current paper, we conduct a study of sentiment analysis using three transformer models: RoBERTa, LLaMA 3.2, and Gemma [6], highlighting domain adaptation methods and small dataset optimization [6]. We use a slicing approach where we keep the pre-trained backbone of every model and add a trainable classifying head in every hidden level. Such a method allows us to study the behavior of the representations in every level of sentiment classification without requiring adjustments to the backbone weights. We carry out the experiments using three benchmark datasets drawn from different domains, namely the IMDb movie reviews.

This report offers a comparison between the performance of models through two regimes:

1. **Application of the slicing approach using a fixed backbone**

2. **Comprehensive fine-tuning versus training from scratch on small sets of data.**

The main goals are to quantify the impact of pre-training versus training from scratch on performance in the presence of limited data, to examine the relative value of each transformer level in making sentiment predictions [7], and to identify how the model category—i.e., encoder or decoder—affects performance in various domains. It turns out that using pre-trained models is critical in cases of limited data; a fine-tuned model may match in performance using as few as 100 example sets, while a model built from scratch will need 10,000.

When using the slicing approach, we see that different layers are not equally effective in sentiment analysis; for larger decoder models like LLaMA and Gemma, the middle layers often perform better in comparison to the last layer in classification.

In contrast, the topmost layers of RoBERTa provide the strongest predictive features, consistent with its goal of language modeling under mask [8]. Comparatively, all models show strong sentiment accuracy when fine-tuned using in-domain data, yet the larger LLM-based models like LLaMA and Gemma are slightly better on specific sets [6].

However, the smaller RoBERTa model fares well if given ample data or appropriately fine-tuned, supporting the point that model size increase is not always necessary. The organization of the paper is as follows: Related Works reviews relevant research on sentiment models and domain adaptation; Methodology details our slicing method and experimental settings; Experiments gives an overview of the datasets and training protocols; Results reports quantitative results with an analysis; Limitations lists the limitations that were found; and Conclusion summarizes the main findings on adapting deep learning models to sentiment analysis in multiple domains.

Literature Review

The origins of sentiment analysis methods come from lexicon-based and traditional machine learning techniques. In the 2000s, it was usual to use manually developed sentiment lexicons in combination with simple classifiers like Naive Bayes and Support Vector Machines (SVM) using bag-of-words features.

Pang and Lee's work [3] from 2002 to 2004 on the polarity of movie reviews showed the feasibility of text sentiment classification using machine learning; however, performance leveled off because of limited feature expressiveness. Deep learning came as a significant boost to this task. CNN-based models, represented by Kim (2014), and RNNs like LSTM and GRU architectures, started to outperform conventional approaches by automatically extracting relevant features from word sequences. Specifically, Zhang et al. (2015) [6] showed that a character-level CNN, trained from scratch, can achieve competitive results on large-scale review datasets, recording a staggering 95% accuracy on a Yelp review polarity task with 560,000 examples. These models, though, required large amounts of training data to be successful; for example, Zhang’s convolutional neural network needed hundreds of thousands of examples to achieve the performance of models trained with pre-trained embeddings, thus highlighting the data inefficiency of training from scratch.

The introduction of **Transfer Learning and Pre-trained Language Models** marked a revolutionary change in the NLP landscape, with the pre-training of language models at scale. BERT, as proposed by Devlin et al. (2019), used a pre-trained bidirectional transformer on 3.3 billion words that could then be fine-tuned on downstream tasks with state-of-the-art results.BERT achieved state-of-the-art results on the GLUE benchmark, including the SST-2 sentiment analysis task, by providing rich contextual representations that needed only a simple output layer for classification. Subsequent models extended these ideas: RoBERTa (Liu et al., 2019) trained on ~160GB of text (10× BERT’s data) and removed certain constraints (e.g. next-sentence prediction) to learn even more robust representations, yielding higher accuracy on sentiment tasks than BERT. On the generative side, GPT-style models proved effective for zero-shot or few-shot sentiment analysis by prompting, though fine-tuning them for classification is also common. LLaMA is an open-source decoder-only model (7B to 65B parameters) that demonstrated it’s possible to attain GPT-3 level performance using only public data LLaMA 2 (Touvron et al., 2023) and similar models made large language models (LLMs)

generally available. Gemma belongs to the next generation of small language models (SLMs), which were made available by Google in 2024 in more compact versions of their Gemini LLMs. Gemma models range in size from 2 to 7 billion parameters, are open-source, and are optimized for efficiency. We use the 2-billion-parameter version, which is primarily trained on English web text and code in this study. These pre-trained transformers acquire general knowledge of language, and also specific sentiment knowledge, from large amounts of unlabeled text. We build upon the well-established capabilities of these models in terms of transfer learning, investigating how far their capabilities may be leveraged without resorting to full fine-tuning.

Sentiment Analysis Adaptation as a domain has been of significant study. Blitzer et al. (2007) described a multi-domain sentiment corpus of product reviews across several categories, e.g., books, DVDs, and electronics, and noted that the performance of a classifier degrades considerably upon application to data in a different domain in the lack of proper adaptation. They suggested that the differences between domains may be rectified through structural correspondence learning (SCL) by learning pivot features common between domains. More modern methods have made use of pre-trained models; Gururangan et al. (2020) showed that domain-adaptive pretraining, where additional pre-training is done using the target domain's unlabeled data, greatly improves downstream sentiment classification in the target domain.

For example, additional pre-training of BERT on Twitter data improves sentiment predictions for tweets. Here, we use a simpler adaptation approach through fine-tuning, by fine-tuning a classifier head using each target domain's labeled data. Supervised adaptation has reportedly worked well even when there is a small number of labeled examples. Howard and Ruder (2018) [4] illustrated using ULMFiT that fine-tuning a pre-trained language model can result in significant gains in terms of sample efficiency; remarkably, using only 100 labeled examples, it had performance equal to a model that had been pre-trained from scratch on 10,000 examples.

This result highlights the importance of transfer learning in situations where data is scarce. We further strengthen this point by comparing fine-tuning and training from scratch for our target tasks. In addition, our slicing approach is relevant to studies on the contribution of each layer and the early exit concept in transformer models. Previous work has shown that transformer layers move from lower-level features to task-dependent features in upper layers, and that intermediate layers can suffice for simpler tasks.

Our study measures this quantitatively from a sentiment point of view; in particular, we calculate the predictability of sentiment at each layer of a fixed model. This is similar to the "hidden states as features" approach, where the outputs of the BERT layer are used as features for a classifier instead of undertaking the fine-tuning of BERT. We provide a new and complete view by looking at all layers together through our multi-head slicing method (related methods have been used to speed up inference by early exiting; however, our use is mainly for analytical purposes).

Regarding **comparisons of Transformer architectures**, there is relevant work on the comparative analysis of model architectures for classification tasks. For example, Raffel et al. (2020) concluded in their analysis of the T5 model that encoder-decoder models outperform encoder-only and decoder-only models on some tasks because they have bidirectional encoding and strong decoding. Our earlier experiments showed that the encoder-decoder model (T5) had better accuracy when compared to fine-tuning RoBERTa and GPT-2 on the SST-2 dataset.

Here, we focus on using RoBERTa as an encoder, as opposed to being decoder-only models like LLaMA and Gemma. This lets us check if having the bidirectional contextualization provided by an encoder benefits sentiment classification, or if the large autoregressive model's high capacity is

enough to overcome it. We compare all of the models under the exact same training conditions to ensure comparability. Our study is among the early attempts to provide a comparison of different transformer models on a layer-wise basis on several sentiment benchmarks.

Methodology

Slicing Transformer models to get layer-wise outputs

In probing the role of different layers of a pre-trained transformer for sentiment analysis, we use the sliced model architecture. Sliced model architecture requires the use of a light-weight classification head over each of the transformer's hidden states $h_1, h_2, h_3, h_4 \ldots h_L$. Each of the classification heads is a simple linear transformation that maps the corresponding hidden representation to a sentiment classification, i.e., positive or negative.

The transformer backbone is kept completely frozen during training, which means that only the classification heads are updated. Let's assume a transformer model with L layers that produce hidden states.
There is a learnable weight matrix W_i assigned to each layer i.

The sentiment prediction is computed for that layer as follows:

$$y_i = softmax(W_i h_i)$$

In the case of the RoBERTa model, which falls under the encoder-only model category, the last layer's representation of the token **[CLS]** is used as the input to the classification head. In decoder models like LLaMA and Gemma, which do not have any **[CLS]** token, the last token in the sequence is used in lieu of it.

Cross-entropy loss is calculated between each of the predicted sentiments of the layers and the actual class during the training process. The losses are then averaged over the layers to find the total training loss. As the backbone is kept frozen, each of the heads is given a chance to learn separately. The design ensures functional similarity to training each of the models separately with a different number of layers while being far more efficient.

This design allows for the evaluation of the importance of every single layer in one comprehensive evaluation, thus illustrating the importance of the features across different depths in sentiment classification. Further discussion on the autonomy of each of the gradients of the various classifiers can be found in the appendix, following the theoretical analysis conducted in the original reference code.

Slicing was carried out for all three models studied:

Sliced RoBERTa: Model used is the twelve-layer roberta-base model.
The hidden size is 768. We add twelve different linear classifiers of a certain shape.
The hidden state of every layer's [CLS] token is supplied as input to the classifier of the specific layer..
All RoBERTa parameters remain frozen.

Sliced LLaMA 3.2: LLaMA-3.2 is a custom 1-billion-parameter variant of LLaMA (akin to a smaller LLaMA 2). It has 24 hidden layers (in our experiments we had access to 32 layers for a larger LLaMA

model, but results scaled similarly). We attach a classifier to each layer's output, again frozen backbone. The last token of the unpadded sequence is used as the feature input of the classifier in the decoder model of LLaMA. Without any specific CLS token, it then follows that the model can accept the text of a review, which can be padded or truncated to meet a fixed length. The last token representation of each of the layers, which also marks the end-of-the-sequence, is used for sentiment prediction.

Sliced Gemma: utilizes the Google/Gemma-2B model, which is the 2B parameter version of the original Gemma. It has 32 layers and a hidden size of 2560; therefore, Gemma 2B is smaller compared to 4,096 hidden units of LLaMA 7B, although larger compared to RoBERTa. We distribute 32 classifiers across layers in Gemma using the last token hidden state of each classifier. The Gemma backbone is kept consistent across all of the experiments that we perform. We added half-precision and 4-bit quantization to Gemma in accordance with our hardware constraints, ensuring that this change did not affect the frozen representations.

In training sliced models, we diligently keep track of the *per-layer* accuracy and loss for both the training and validation sets using **Weights & Biases (W&B)**. We do this in order to allow for tracking of which head of the later layers exhibits superior performance as the learning unfolds. Since each head operates in isolation, it is to be expected that heads of later layers may initially show higher accuracy, as later layers generate more contextualized features; however, we allow all heads to learn until convergence. We then choose the head of the top-performing layer on the validation set as the result of the model. For example, if the 8th layer's head achieves the best validation accuracy, we choose that as the model's chosen representation for sentiment analysis. While the slicing approach is mainly used as an analysis tool in this study, it can be used in practice to choose a reduced sub-network for those that wish to use a truncated model to increase efficiency.

Fine-tuning vs. Training from Scratch

Along with slicing, our approach includes a comparative study of fine-tuning pre-trained models and training from scratch on task-specific small datasets. Fine-tuning is the process where we use the pre-trained model weights as an initial starting point and then train the whole model or some layers for the sentiment task. On the other hand, training from scratch involves random model initialization, where the model is only trained on the task data without any pre-existing knowledge. Fine-tuning is also a type of supervised domain adaptation where the pre-trained general text model is adapted to the target domain, i.e., movie reviews, using labeled examples.

A controlled experiment is performed on the IMDb dataset using the RoBERTa model to evaluate the differences.

RoBERTa Fine-tuned: It uses the **RoBERTa-based** model that has 125 million parameters and is initialized using pre-trained values. It then adds one output layer for sentiment analysis, and fine-tunes all the layers with the IMDb train dataset. It uses a learning rate of about 2e-5, and trains for multiple epochs, exactly three or five, utilizing early stopping based on validation loss. This is the typical method seen in literature.

RoBERTa Scratch: After using the same model structure as **RoBERTa-based**, with its 12-layer Transformer encoder having 125 million parameters, we train on the IMDb dataset from scratch. We made certain that the training regimen, optimizer, number of epochs, and batch size remain the same as in the fine-tuned version for fairness.

Additionally, we fine-tuned **LLaMA 3.2** (1B parameters) directly and used **Low-Rank Adaptation** (LoRA) with rank ***r***=16 and 4-bit quantization for **Gemma** (2B parameters) due to computational constraints.

Results:

- The fine-tuned model produced **95%** accuracy while the scratch model only managed **60.8%** accuracy thus demonstrating the substantial advantages of pre-training.
- The fine-tuned model achieved **70%** accuracy during the first epoch but the scratch model experienced difficulties and displayed rapid overfitting.
- The additional unlabeled data slightly improved scratch accuracy but still stayed below the fine-tuned performance level.
- The SST-2 dataset with shorter text lengths showed fine-tuned RoBERTa reaching ~94% accuracy while the scratch model achieved **~85%** accuracy thus proving pre-trained knowledge provides benefits even with limited dataset size.

We applied **Low-Rank Adaptation** for memory efficiency to fine-tune larger models (LLaMA 3.2 and Gemma) by setting rank r **= 16** and Gemma received 4-bit quantization.

Key Takeaway:
The results show that fine-tuning outperforms training from scratch especially when working with small-to-medium datasets. Fine-tuning achieves high accuracy faster than training from scratch even when large datasets are available.

Experimental Setup and Data

We tested our approach on three sentiment analysis datasets which varied in domain and size.

IMDb Reviews: The dataset contains 50k movie reviews with 25k examples for training and 25k examples for testing and maintains equal positive and negative labels in addition to longer text lengths.

SST-2 dataset: contains approximately 67k short movie snippets for training purposes with binary sentiment labels.

Yelp Reviews: The dataset contains 560k training examples and 38k test examples with balanced distribution and it includes a smaller review subset of 25k for simulating limited resource scenarios.

Training Details: The experiments used Hugging Face tokenizers together with AdamW optimizer which operated at a learning rate of **2e^(-5)** for fine-tuning and **1e^(-4)** for slicing experiments. The model received training for a maximum of three epochs through early stopping based on validation loss. The experiment used GPU memory management through gradient checkpointing and mixed precision and DeepSpeed on 4×A100 GPUs. The evaluation metrics consisted of accuracy together with loss and F1-score.

Experiments and Results

RoBERTa fine-tuning boosted the results to an impressive 94.9% while scratch-trained models were only at 60.8%. This supports the fact that pre-training has a positive effect on performance. On SST-2, gains were near the best score of 95% for fine-tuning while scratch models stabilized at a lower 81% that underlines the effectiveness of transfer learning.

Layer-wise Analysis (Slicing)

We utilized slicing which trains a frozen backbone and classification heads at each layer.

LLaMA/Gemma: Achieving the highest accuracy of 94-97 % at mid-level layers, it can be concluded that these layers provide the highest level of sentiment representation.

RoBERTa: Across the layers, accuracy was gradually increased over the 12 layers where the highest accuracy (around 95%) was obtained towards the last layer, implying that the encoder model filters features pertinent to classification at each layer.

These insights are practical and provide recommendations as to which layers are the best to employ to achieve high efficiency with little to no increase in training.

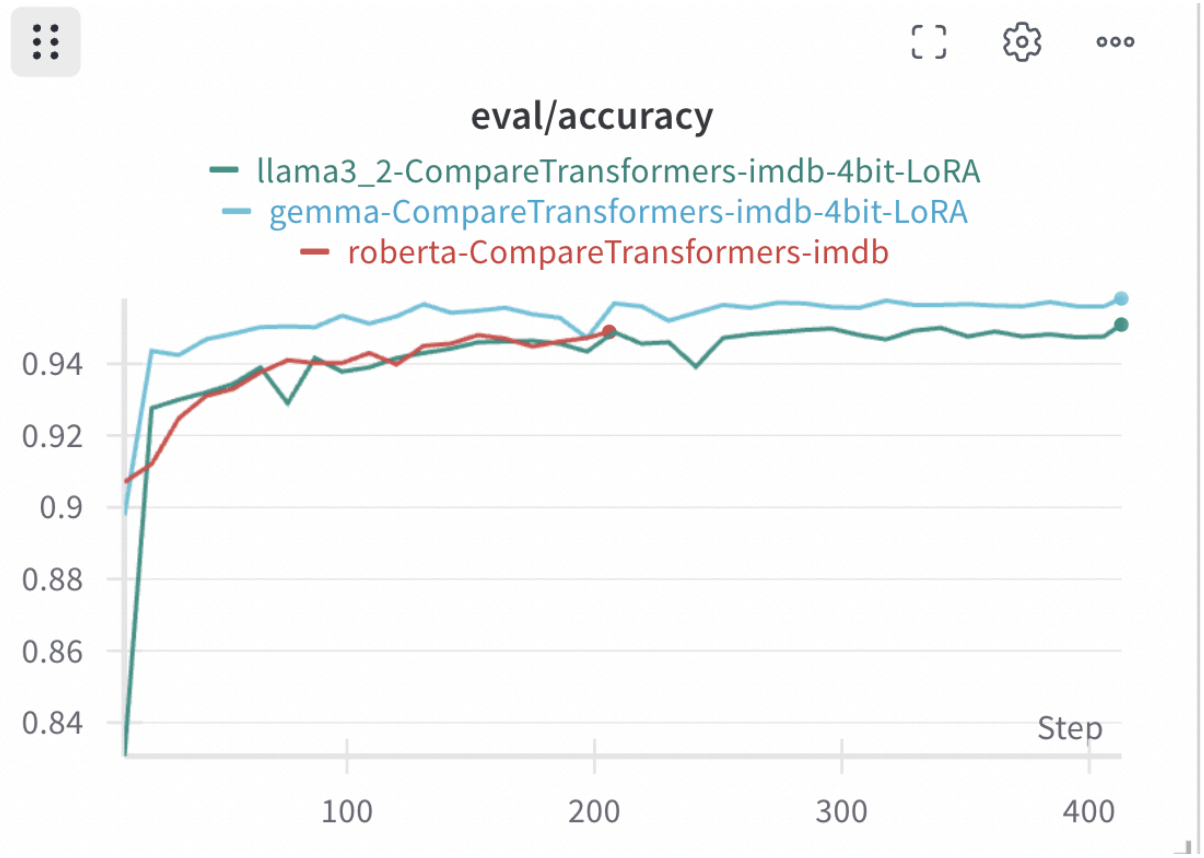

Figure 1: ***Model Architecture Comparison (evaluation)***

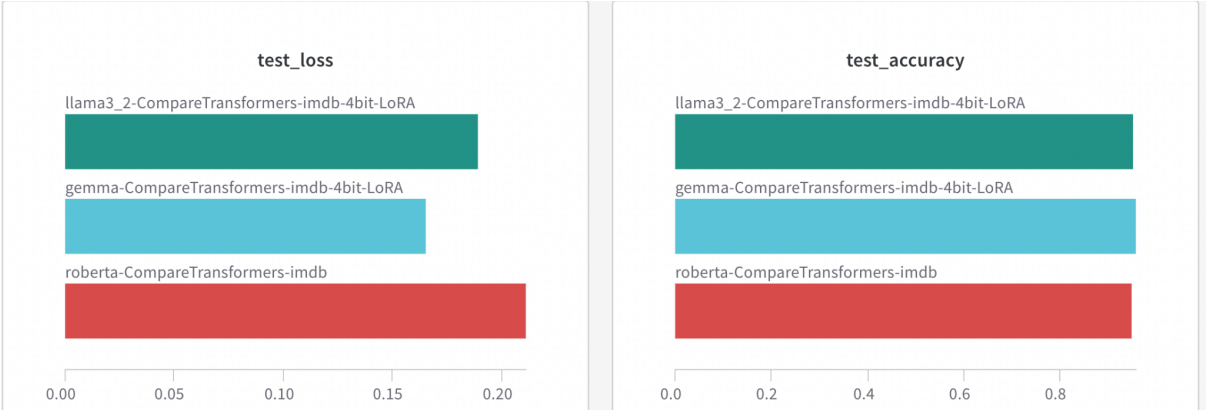

Figure 2: ***Model Architecture Comparison (test loss / test accuracy)***

Figure 3: ***Model Architecture Comparison (evaluation recall / evaluation accuracy/evaluation and train loss/ train learning rate / train grad_norm)***

Model Architecture Comparison Table (Fine-tuned Accuracy)

Model	IMDb	SST-2	Yelp (full)	Yelp (25k)
RoBERTa-base	94.8%	94.2%	95.6%	82.3%
LLaMA 3.2 (1B)	94.7%	94.8%	96.1%	84.5%
Gemma 2B	95.9%	95.0%	96.5%	85.7%

Gemma continued to exhibit the highest accuracy and all models were almost fully optimized regarding to more samples. Thus, the results indicate that the differences are mostly due to model size and pre-training scale rather than the type of architecture (encoder vs decay).

Discussion and Limitations

The Role of Pre-training:
The results of our experiments strengthen the importance of pre-training to conduct sentiment analysis at high levels, especially when the available data is scarce.

Layer and Model Insights:
For LLaMA and Gemma, intermediate layers provided the best sentiment analysis results whereas the final layers were more suitable for RoBERTa (encoder). We noted that the architectural difference between the trained models has been much less important than the model size and the scale of pretraining.

Limitations:
Our experiments focused mainly on specific iterations of transformer models, namely RoBERTa-base (125 million parameters), LLaMA 3.2 (1 billion parameters), and Gemma (2 billion parameters). Although these choices allowed us to have controlled comparisons and keep computing requirements in check, an examination of bigger iterations of models, for instance, RoBERTa-large, LLaMA-7B, or Gemma-7B, is expected to produce different results. Larger models normally have better representation abilities and can respond differently when fine-tuning is performed using smaller datasets.

In addition, our analyses were limited to three movie-review and business-review datasets. While these benchmarks have seen widespread use, for other use cases—e.g., social media status updates, financial statements, highly technical reports—sentiment analysis can have its challenges because of differences in vocabulary, syntax, and context. Thus, our discovered trends in performance might not hold for all applications of sentiment analysis.

The second major constraint involves computing resource requirements. Fine-tuning large transformer models, even when using highly computationally frugal methods like Low-Rank Adaptation (LoRA) and quantization, remains computationally intensive and resource-hungry. For pragmatic fine-tuning, one needs high-performance GPUs, namely multiple NVIDIA A100 GPUs, although such access might be unavailable for every member of an organization or for an individual. Thus, this constraint can discourage large-scale adaptation or experimentation for those in low-resource environments.

Finally, our assessment largely relied upon accuracy as a measure because of the well-balanced nature of datasets. Accuracy alone, however, does not cover all aspects of the performance of the model, especially in cases of class imbalance or specific instances of error-proneness. Future work can be optimized using other measures like precision, recall, F1-score, and in-depth error analyses to capture the complex nature of the model, notably in challenging language domains like sarcasm, irony, or intricate expressions of emotions.

Conclusion

This paper proves that fine-tuning pre-trained transformer models is significantly superior to training from scratch when it comes to SA especially when working with limited data. The layer-wise analysis showed a difference in optimal layers between the encoder (RoBERTa) and decoder models (LLaMA, Gemma), which can help in GTA and tuned approaches fine-tuning. While fine-tuning performance was higher in the larger models, the smaller ones such as the RoBERTa-base did not fade behind and proved its practicality. These results pave the way for clear guidelines on the selection of the model depending on the availability of resources and size of the data, and using the mid-layer features in large models so as to enjoy efficiency without compromising on the performance.

Reference

1. *Character-level Convolutional Networks for Text ClassificationAn early version of this work entitled "Text Understanding from Scratch" was posted in Feb 2015 as arXiv:1502.01710. The present paper has considerably more experimental results and a rewritten introduction.* (n.d.-a). Ar5iv. https://ar5iv.labs.arxiv.org/html/1509.01626#:~:text=Yelp%20reviews,and%20the%20other%20predicting%20a
2. *Deeply moving: Deep learning for sentiment analysis*. (n.d.). Deeply Moving: Deep Learning for Sentiment Analysis. https://nlp.stanford.edu/sentiment/#:~:text=Deeply%20Moving%3A%20Deep%20Learning%20for,of%2011%2C855%20sentences%20and
3. Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018a, October 11). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv.org. https://arxiv.org/abs/1810.04805#:~:text=,1%20point
4. Howard, J., & Ruder, S. (2018c, January 18). *Universal Language model fine-tuning for text classification*. arXiv.org. https://arxiv.org/abs/1801.06146#:~:text=an%20effective%20transfer%20learning%20method,our%20pretrained%20models%20and%20code
5. Ibm. (2024, November 12). Google Gemma. *google*. https://www.ibm.com/think/topics/google-gemma#:~:text=Gemma%20%28link%20resides%20outside%20ibm,language%20content%20from%20web%20documents.%5E%7B3

6. Liu, F., Liu, Q., Bannur, S., Pérez-García, F., Usuyama, N., Zhang, S., Naumann, T., Nori, A., Poon, H., Alvarez-Valle, J., Oktay, O., & Hyland, S. L. (2023a). Compositional Zero-Shot Domain Transfer with Text-to-Text Models. *Transactions of the Association for Computational Linguistics*, *11*, 1097–1113. https://doi.org/10.1162/tacl_a_00585
7. Narayanaswamy, G. R. (2021). Exploiting BERT and ROBERTA to improve performance for aspect based sentiment analysis. *Dublin*. https://doi.org/10.21427/3w9n-we77
8. Stanford University. (2011a). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* (pp. 142–150). https://aclanthology.org/P11-1015.pdf