Data Mining, Text Mining and Big Data Analytics Project Work

Gian Mario Marongiu Giacomo Gaiani
Salvatore Guarrera

July 2024

Abstract

This project investigates the application and development of a text embedding strategy using a composite approach that integrates large and small language models for text classification. The primary objective was to examine whether this hybrid model could enhance the efficiency and effectiveness of embeddings within the MTEB framework. We detail our methodology and compare our results to existing benchmarks. Our findings suggest that, for text classification tasks, using a smaller model like *RoBERTa* alone still yields superior results compared to our hybrid model approach.

1 Introduction

Text classification tasks are central to many applications in Natural Language Processing, relying heavily on the quality of text embeddings. The concept of word embeddings is one of the most powerful in deep learning for NLP applications. A set of word vectors for a vocabulary is able to capture the meaning of words and the relationships between them and their context. These vector representations embed words in a feature space, hence the name word embeddings [5]. Our work involved studying the source code from the MTEB framework and developing our tests by gaining a deep understanding of the logic underlying the MTEB methods and functions.

The MTEB suite is composed of several tasks, in this study we focus on the classification task, for which many models have been proposed and their performance are published in the official leaderboard on *Hugging Face* [2]. Our tests were based on an embedding pipeline that utilizes both an untrained large language model, *Gemma* [7], and a trained smaller model, *RoBERTa* [6]. These models are linked by a projection layer, which is also trained. The goal was to assess potential improvements in performance metrics such as accuracy and F1 score with respect to the single components.

The idea for this approach was inspired by a recent study that explores the inverse concept: utilizing a hybrid approach to combine a LLM encoder-only

and SLM decoder-only for efficient autoregressive decoding in natural language generation tasks [3]. This approach shows substantial speedups of up to 4× with only minor performance penalties for tasks like translation and summarization. By applying a similar concept to text classification, we aim to leverage the strengths of both large and small models, using a trained projection layer to bridge them effectively.

1.1 Common Embedding Methods

As widely described in [5], various methods have been developed to generate these embeddings, each with its unique approach and characteristics:

- Count-based embeddings (such as Bag of Words [4]) are derived from word occurrence counts within a corpus, capturing frequency but not the order of words. The size of these vectors scales with the vocabulary, potentially causing scalability issues in extensive datasets.
- Compositional methods generate embeddings by algebraically composing word vectors, without necessitating parameter training. The effectiveness of compositional embeddings depends on the rules used for combining the vectors, impacting their ability to reflect complex linguistic properties.
- Leveraging large volumes of unannotated text, unsupervised embeddings rely on the distributional hypothesis, which posits that words with similar contexts share meanings. The design of the objective function and the learning framework significantly influences their characteristics.
- Contrary to unsupervised methods, supervised embeddings use labeled data, allowing for task-specific tuning influenced by the neural network type and label quality. These embeddings are effective for targeted applications but depend heavily on the relevance and quality of the training labels.
- Transformers utilize attention mechanisms to produce context-aware embeddings, considering the full context of word appearances. While embeddings from pre-trained transformers are general-purpose, fine-tuning can yield models specialized for particular tasks.
- Integrating textual data with other data types like visual or acoustic information, multimodal embeddings provide enriched representations. The success of these embeddings hinges on the methods used to merge different data modalities, affecting their application effectiveness.

Our approach seeks to capitalize on the transformative potentials of transformer-based and compositional methods, combining the strengths of both to explore new avenues in embedding technology. By harnessing the deep contextual awareness of transformer models, specifically through a large language model like

Gemma, and integrating the flexible, dynamic strategies characteristic of compositional methods, our method aims to generate embeddings that are semantically detailed and versatile for various NLP tasks. This hybrid approach strives to balance context-awareness with processing efficiency and adaptability, positioning our method at a potential convergence point between unsupervised and supervised learning paradigms.

2 Model, Training and Evaluation

Our model architecture (Figure 1) integrates two core components: the Gemma language model and the RoBERTa model. Gemma generates deep contextual embeddings, which are then adapted through a projection layer to fit the input requirements of RoBERTa.

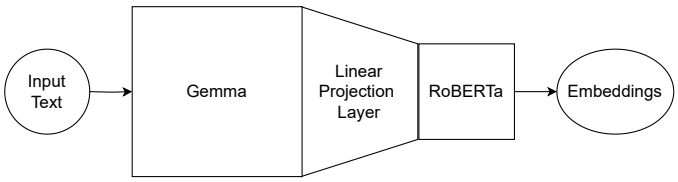

Figure 1: The composite model architecture combining Gemma and RoBERTa. In this configuration *Gemma* is the only layer that is not trained due to hardware limitations.

Gemma is designed for generative tasks, so we expect the final layers to perform poorly in classification scenarios, as they are optimized for generative applications. Hence, we have explored various configurations, extracting representations from the last 18 layers individually or in combinations, merged through different pooling techniques.

To train the model effectively, we attach an additional classification head, crucial for proper learning of the projection and *RoBERTa* layers. The training regimen varies depending on the method, with a specifically tailored low learning rate for downstream tasks involving *RoBERTa*. In contrast, the evaluation phase is conducted using a task pipeline from the MTEB framework, which does not necessitate the classification head; thus, it is removed for evaluation. This pipeline directly leverages the embeddings generated by our model to train a regressor on a training set. This same regressor is subsequently used to make predictions on the test set embeddings, also produced by our model. The better the regressor performs, the higher the quality of the embeddings is considered to be.

3 Experimental Setup and Results

We conducted extensive testing to evaluate the effectiveness of our model under various configurations and parameters. The experiments were designed to be reproducible, with detailed settings of hyperparameters provided in our supplementary materials [1].

In addition to the model described in Chapter 2, we defined two baseline models for comparison:

- *Gemma_only* Model consists solely of the *Gemma* language model component. This model is not trained; however, similar to the full model, we can select from which of the last 18 layers to extract the output embedding.
- *RoBERTa_only* Model is exclusively composed of the *RoBERTa* language model component. This model can be trained, and its performance has been evaluated both with and without training.

Given the time and hardware limitations, the tests were mainly performed on two of the datasets that compose the MTEB classification library: *AmazonCounterfactualClassification* and *EmotionClassification*. Each dataset has been evaluated in four different ways: by using the *RoBERTa_only* model, both with and without fine tuning; by using the *Gemma_only* and extracting the embedding from each transformer layer; by using the Hybrid Model, extracting the Gemma output features from each transformer layer. We performed an additional test, using this output features of the four most promising layers of the *Gemma* model and adding a pooling layer between the LLM and the projection layer. The test has been performed both for max and mean pooling.

The figures 2 and 3 show respectively the values of the Accuracy and F1 Score for the test performed on the *AmazonCounterfactualClassification*. The fine tuned *RoBERTa* model achieved the best score (0.92358 in Accuracy; 0.88786 in F1), while the use of an Hybrid model increase the results in almost all layer with respect the the base *Gemma* model. The scores of a single layers do not seem to respect a specific pattern. In this specific dataset the best score of the hybrid model was achieved using four Gemma transformers layer (0.87493 in accuracy; 0.82954 in F1).

The figures 4 and 6 show respectively the values of the Accuracy and F1 Score for the test performed on the *EmotionClassification*. Again, the best results were achieve by the trained *RoBERTa* model (0.84130 in accuracy; 0.79769 in F1). In this second dataset the hybrid model showed fluctuating results while the *Gemma* model seemed more stable. In this specific dataset the best score of the hybrid model was achieved using seven Gemma transformers layer (0.39920 in accuracy; 0.32163 in F1).

Tables 3 and 3 show a test performed on both datasets, adding a pooling layer before the projection layer, sampling form the four most promising layer of *Gemma*. The tests show a deterioration in results with respect of the single layers, for both Max and Mean Pooling.

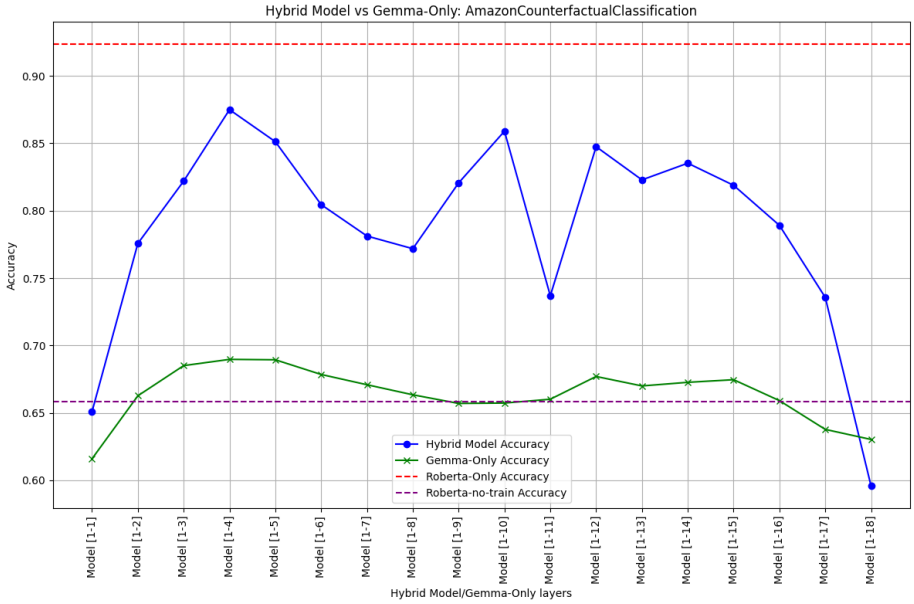

Figure 2: Accuracy comparison between Hybrid Model and Gemma Only in AmazonCounterfactualClassification task

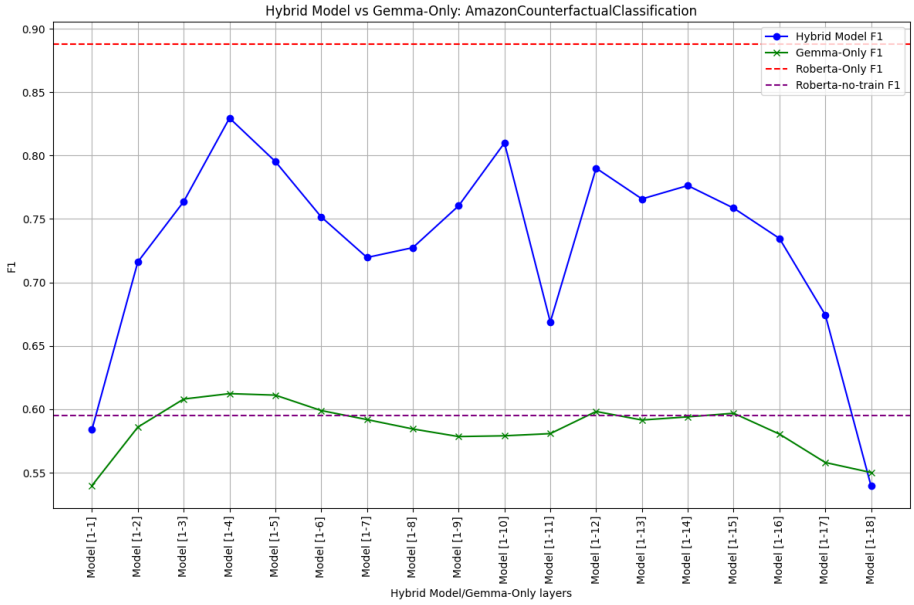

Figure 3: F1 comparison between Hybrid Model and Gemma Only in AmazonCounterfactualClassification task

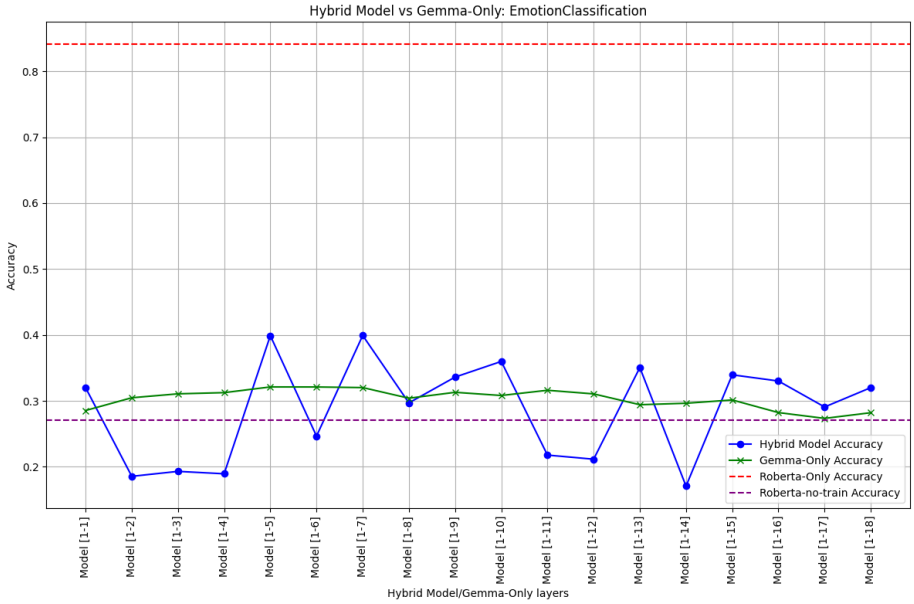

Figure 4: Accuracy comparison between Hybrid Model and Gemma Only in EmotionClassification task

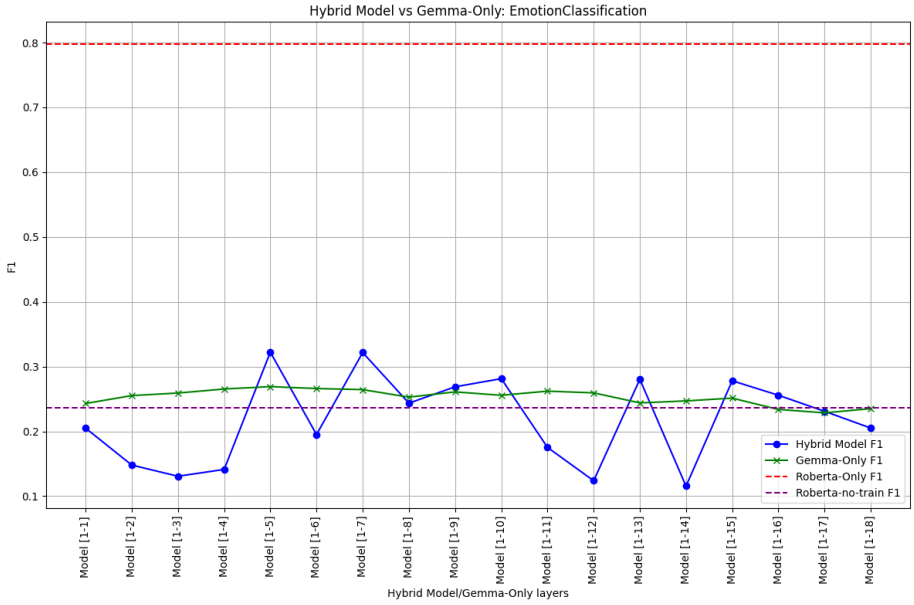

Figure 5: F1 comparison between Hybrid Model and Gemma Only in Emotion-Classification task

	Accuracy	F1	Layers
Max Pooling	0.66925	0.60444	4, 5, 10, 14
Mean Pooling	0.60552	0.53763	4, 5, 10, 14

Table 1: Results for the AmazonCounterfactualClassification Dataset

	Accuracy	F1	Layers
Max Pooling	0.20540	0.13375	5, 7, 10, 13
Mean Pooling	0.36020	0.28673	5, 7, 10, 13

Table 2: Results for the EmotionClassification Dataset

In the *AmazonCounterfactualClassification* task, the hybrid model surpasses the *Gemma-only* model in both accuracy and F1 score. For *EmotionClassification*, the hybrid model shows a smaller advantage, excelling only with certain layer choices. However, fine-tuned *Roberta* consistently outperforms the hybrid model across all configurations by a significant margin, establishing it as the superior model.

Considering the leaderboard [2], the hybrid model can be compared with the approaches used there. For the *AmazonCounterfactualClassification* task, our best accuracy is 0.87493, achieved using four *Gemma* transformer layers. This places our hybrid model in 9th position, as the first model we outperform is the instructor-base model.

For the *EmotionClassification* task, our highest accuracy is 0.39920, which was achieved with seven *Gemma* transformer layers. This places our hybrid model at the lower end of the leaderboard.

Rank	Model	Model Size (Million Parameters)	Memory Usage (GB, fp32)	Average	AmazonCounterfactualClassification (en)	AmazonPolarityClassification	AmazonR (en)
4	NV-Embed-v1	7851	29.25	87.35	95.12	97.14	55.47
2	neural-embedding-v1			87.91	93.1	97.54	61.17
3	stella_en_1.5B_v5	1543	5.75	87.63	92.87	97.16	59.36
1	SFR-Embedding-2_R	7111	26.49	89.05	92.72	97.31	61.04
5	stella_en_400M_v5	435	1.62	86.67	92.36	97.19	59.53
6	gte-Qwen2-7B-instruct	7613	28.36	86.58	91.31	97.5	62.56
13	voyage-lite-02-instruct	1220	4.54	79.25	88.31	96.32	56.25
67	instructor-large	335	1.25	73.86	88.13	91.53	47.86
88	instructor-base	110	0.4100000000000003	72.36	86.21	88.36	44.64
80	instructor-xl	1241	4.62	73.12	85.09	86.54	42.96
219	LLaMA-embedding				84.82	76.88	36.72

Figure 6: MTEB Leaderboard ordered by AmazonCounterfactualClassification task, in red the first model we outperform

4 Conclusion

The base idea presented in [3], uses the LLM-to-SLM architecture to successfully accelerate the autoregressive decoding for generative tasks. Translating this idea for classification scenarios, lead to a series of complications. Using the internal representation of a frozen LLM decoder-only, does not seem to have the positive outcome we had imagined. As further experiments, a possible attempt would be testing different LLMs and settings, trying alternative tasks, or train Gemma instead of freezing it. That said, the hardware and time limitation make part of those experiments not feasible. Given the lack of patters in the presented results, it is hard to draw a line towards future improvements.

References

[1] Embedder-gemma-roberta github repository. `https://github.com/GianM0027/Embedder-Gemma-RoBERTa`. Accessed: July 2024.

[2] Massive text embedding benchmark (mteb) leaderboard. `https://huggingface.co/spaces/mteb/leaderboard`. Accessed: July 2024.

[3] Benjamin Bergner, Andrii Skliar, Amelie Royer, Tijmen Blankevoort, Yuki Asano, and Babak Ehteshami Bejnordi. Think big, generate quick: Llm-to-slm for fast autoregressive decoding, 2024.

[4] Zellig S. Harris. *Distributional Structure*, pages 3–22. Springer Netherlands, Dordrecht, 1981.

[5] Francesca Incitti, Federico Urli, and Lauro Snidaro. Beyond word embeddings: A survey. *Information Fusion*, 89:418–436, 2023.

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

[7] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024.