PAI at SemEval-2025 Task 11: A Large Language Model Ensemble Strategy for Text-Based Emotion Detection

Zhihao Ruan*, Runyang You*, Kaifeng Yang, Junxin Lin,
{archfool.ruan, junyang.yu.2000, yangkaifeng1985, ljx03123}@gmail.com
Wenwen Dai, Mengyuan Zhou, Meizhi Jin, Xinyue Mei
{jk123124dww, zhoumengyuanstce, jinmeizhi0924, vrmei2146}@gmail.com
Ping An Life Insurance Company of China

Abstract

This paper describes our system used in the SemEval-2025 Task 11: Bridging the Gap in Text-Based Emotion Detection. To address the highly subjective nature of emotion detection tasks, we propose a model ensemble strategy designed to capture the varying subjective perceptions of different users towards textual content. The base models of this ensemble strategy consist of several large language models, which are then combined using methods such as neural networks, decision trees, linear regression, and weighted voting. In Track A, out of 28 languages, our system achieved first place in 19 languages. In Track B, out of 11 languages, our system ranked first in 10 languages. Furthermore, our system attained the highest average performance across all languages in both Track A and Track B.

1 Introduction

The objective of Task 11(Muhammad et al., 2025b) is to determine, within different linguistic contexts, what emotion most people would perceive the speaker to be feeling based on a sentence or short text snippet uttered by the speaker(Muhammad et al., 2025a)(Belay et al., 2025). Track A of Task 11 includes 28 languages, while Track B consists of 11 languages. The task requires detecting the presence of the following emotions and assessing their intensity: joy, sadness, fear, anger, surprise, and disgust.

Given the highly subjective nature of emotion detection in textual content, different annotators may provide varying answers regarding whether a certain emotion is present in the text (or the degree to which it is present). Similarly, for large language models (either untrained or only lightly fine-tuned), different models may output differing judgments for the same textual content. Therefore, bridging the gap between these two sources of subjectivity—annotator variability and model inconsistency—becomes the central focus and optimization goal of our system.

Considering the powerful capabilities of large language models and the potential for catastrophic forgetting resulting from improper training, we select both the original and lightly fine-tuned versions of several models as base models. We then employ ensemble strategies such as neural networks, decision trees, linear regression, and weighted voting to combine the outputs of multiple base models, ultimately providing the final prediction. The rationale behind using a ensemble strategy is that the independent predictions made by multiple base models resemble the behavior of multiple annotators independently labeling data, each with their own judgment tendencies. We hypothesize that, when there is sufficient divergence between the prediction results of different base models, appropriate ensemble strategies can better capture the annotators' labeling outcomes.

2 System Overview

2.1 Prompt Optimization

We present an iterative data-driven prompt optimization framework. The pipeline evaluates and evolves a prompt set through up to $T_{\max}$ iterations, dynamically expanding candidates via `ContextAugment` – labled data examples into prompts to improve their alignment with training data; and `StructVar` – prompt the LLM to Generate syntactically diverse prompt variations (e.g., rephrasing, synonym substitution) to explore broader prompt spaces and avoid overfitting to specific formulations. Then pruning low-performing options (threshold τ). The process terminates early if no improvement exceeds threshold η or reaches $T_{\max}$ iterations. The final output selects the top-k prompts with highest F1 scores, balancing perfor-

*Equal contributions

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1136–1142

mance and generalization.

Algorithm 1 details the full optimization process, including early termination checks and pruning strategies to maintain efficiency.

Algorithm 1 Iterative Prompt Optimization with Early Termination and Multiple Outputs

Input: Initial prompt set $\mathcal{P}_0 \subseteq \mathcal{P}_{\text{baseline}}$,
Training dataset D, validation set D_{val},
Eval Metric: $M = \text{F1 score}$,
Hparams: $\Theta = \{\eta, \tau, T_{\max}, k\}$
Output: k $\mathcal{P}_{\text{final}} = \{p_1^*, p_2^*, ..., p_k^*\}$; Best score s^*

$s^* = -\infty$
for $t = 1$ ***to*** T_{max} **do**
  for $p \in \mathcal{P}_t$ **do**
    Generate responses $\{R_{p,d}\}_{d \in D_{\text{val}}}$ using p
    $S_p = \frac{1}{|D_{\text{val}}|} \sum_{d \in D_{\text{val}}} \text{F1}(R_{p,d})$
  end
  if $S_{p^*} > s^*$ **then**
    $s^* = S_{p^*}$
  end
  if $t \geq T_{max}$ **then**
    Break loop
  end
  $\mathcal{P}_{t+1} = \emptyset$ **for** $p \in \mathcal{P}_t$ **do**
    $p' = \text{ContextAugment}(p, D)$
    $p'' = \text{StructVar}(p)$
    $\mathcal{P}_{t+1} \leftarrow \mathcal{P}_{t+1} \cup \{p', p''\}$
  end
  $\mathcal{P}_{t+1} = \text{Prune}(\mathcal{P}_{t+1}, \tau)$
end
$\mathcal{P}_{\text{final}} = \arg\max_{\mathcal{P}_t} \{S_p \mid p \in \mathcal{P}_t\}$
return $\mathcal{P}_{final}$ *and* s^*

In addition to the prompts mentioned earlier, during inference, we also randomly select 2-3 training samples from the training dataset to serve as few-shot examples.

2.2 Training LLM as Embedding Model

This approach trains smaller LLMs to generate robust embeddings that capture both the semantic and emotional nuances of text, enabling accurate emotion classification. The core principle, inspired by (Liu et al., 2024; Li and Zhou, 2024), lies in extracting representations that reflect the underlying emotional state of a sentence.

Adapter We employed AdaLoRA (Zhang et al., 2023) for parameter-efficient fine-tuning of our pre-trained language model. This method leads to significant computational savings while maintaining or improving performance, particularly when resources are limited.

Emotion Representation We formalize the task as follows: Given an input sentence x, we aim to derive an embedding vector $\mathbf{v}_x \in R^d$ that preserves both its semantic content and emotional salience. Using a prompt template "Detect the emotion of this sentence: {x}", the sentence is processed through a language model Φ with L transformer layers.

To distill sentence-level emotional semantics, we apply a meaning pooling operator Ψ that aggregates token-level representations across the entire sequence. The final-layer hidden states $\mathbf{H}^L$ are used to compute the sentence embedding:

$$\mathbf{v}_x = \Psi(\mathbf{H}^L) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{h}_i^L$$

In the final layer of the model, we added a fully connected layer to transform the embedding vector outputs into outputs suitable for the classification task.

2.3 Ensemble Strategy

We will implement a two-round ensemble strategy, adopting a stacking-like approach for both rounds of fusion. In this section, we will describe the ensemble strategy for the first round.

The first-round ensemble strategy involves using several ensemble schemes to generate individual prediction results, which will then serve as inputs for the subsequent second-round ensemble. In the first round, our system employs four ensemble schemes: neural network, XGBoost, LightGBM, and linear regression, with the five prediction outputs from the large language models mentioned earlier serving as inputs.

The neural network strategy employs a three-layer neural network, with each layer consisting of a fully connected layer of dimension 16, followed by a ReLU activation function. The final output layer uses mean squared error (MSE) as the loss function.

The XGBoost strategy: In Track A, a binary classification approach is used, with negative log-likelihood (NLL) as the evaluation metric. In Track B, a regression approach is adopted, with root mean squared error (RMSE) as the evaluation metric.

The LightGBM strategy: In Track A, a binary classification approach is adopted, with accuracy

as the evaluation metric. In Track B, a regression approach is employed, using root mean squared error (RMSE) as the evaluation metric. The boosting method used for all models is Gradient Boosting Decision Trees (GBDT).

The linear regression strategy: A second-order polynomial regression fitting approach is used.

2.4 Data Analysis and Voting Strategy

In this section, we describe the ensemble strategy for the second round. The ensemble strategy in this round employs a weighted voting approach, where the voting weights of each model are determined through statistical data. The implementation of this voting ensemble strategy differs between Track A and Track B.

This round's weighted voting strategy consists of three steps. The first step is to select the models eligible for voting, the second step is to calculate the voting weights for each model, and the third step is to derive the final prediction results based on the voting outcomes.

Step 1: After training the models on the training dataset, we evaluate them using the development dataset to obtain an evaluation score (F1 score for Track A and Pearson correlation coefficient(PCCr) for Track B). For each language in both tracks, the model with the highest score is selected as the baseline. Models whose scores are lower than the baseline model by 0.2 points are excluded from the subsequent voting and ensemble steps.

Step 2: The voting weight of a model, denoted as $weight$, is derived by multiplying several sub-weights. The first sub-weight, $weight_1$, is the evaluation score of the model on the development dataset, representing the accuracy of the model's predictions.

$$weight_1 = \begin{cases} \text{f1 score} & \text{if Track A} \\ \text{PCCr} & \text{if Track B} \end{cases} \quad (1)$$

The second weight, $weight_2$, is the Jensen-Shannon Divergence (JS divergence), which characterizes the similarity between the distributions of the training dataset and the development dataset. The intermediate variable for calculating the JS divergence is the KL divergence (Kullback-Leibler Divergence). This weight is used to assess and correct the confidence of $weight_1$. Let P and Q represent the distributions of the training dataset and the development dataset, respectively, and let M denote their average distribution.

$$M = \frac{1}{2}(P + Q) \quad (2)$$

$$KL(P\|M) = \sum_x P(x) log \frac{P(x)}{M(x)} \quad (3)$$

$$weight_2 = \frac{1}{2}KL(P\|M) + \frac{1}{2}KL(Q\|M) \quad (4)$$

The third sub-weight, $weight_3$, is used only in Track A. It is calculated based on the ratio between the number of labels in the development dataset and the number of corresponding labels predicted by the model. This weight corrects for potential subjective bias in the model's label predictions.

$$weight_3 = \begin{cases} \sqrt{\frac{count(\text{gold label = 0})}{count(\text{predict label = 0})}} & \text{if label=0} \\ \sqrt{\frac{count(\text{gold label = 1})}{count(\text{predict label = 1})}} & \text{if label=1} \end{cases} \quad (5)$$

Final weight:

$$weight_{TrackA} = weight_1 * weight_2 * weight_3 \quad (6)$$

$$weight_{TrackB} = weight_1 * weight_2 \quad (7)$$

Step 3: In Track A, labels 0 and 1 are first mapped to -1 and 1, respectively. Then, the label predictions from all models are weighted and summed. Finally, a threshold of 0 is applied, where predictions greater than or equal to 0 are classified as 1, and those less than 0 are classified as 0. In Track B, the label predictions from all models are weighted and summed to obtain a score. Based on this score, all cases are sorted in descending order. Next, we combine the labeled data from both the training and development datasets and calculate the percentage of cases labeled with scores from 3 to 0 in the total dataset. Finally, using this percentage, we assign the sorted scores proportionally to the labels from 3 to 0.

3 Experimental Setup

Models Training-free: ChatGPT-4o [1]; Deepseek-V3[2]. Training LLMs as Embedding Models:

[1] https://chatgpt.com/
[2] https://chat.deepseek.com/

Gemma-9b-it[3]; qwen-2.5-32b-instruct[4] Mistral-Small-24B[5]

Hyperparameters

- Training-free: $T_{\max} = 10$, pruning threshold $\tau = 0.5$, top-k prompts $k = 5$.
- Fine-tuning: Learning rate = 1×10^{-5}, attention dimension = 128, batch size = 32. Models trained for 10 epochs with early stopping, evaluated using 5-fold cross-validation.

Ensembling The learning rate for the three-layer neural network model used for the ensemble is set to 3e-2, with the AdamW optimizer and a weight decay of 1e-3. The model is trained for 15 epochs.

For the XGBoost model, the maximum depth is set to 6, and the learning rate is set to 0.1.

For the LightGBM model, the maximum depth is set to 8, the learning rate is set to 0.3, and the number of leaves is set to 31.

4 Results

Due to limited time and GPU resources, we initially conducted experiments and exploration only on the ENG and PTBR languages (Table 1)(Table 2).

In devlopment dataset, compared to the performance metrics of single-path large language models (either untrained or lightly fine-tuned), the fusion strategy consistently provides an additional improvement of 0.01 to 0.02 on top of the optimal single-path model's metrics.

After the release of the test dataset, we plan to apply the same strategy to the 28 languages in Track A and the 11 languages in Track B.

In the final test dataset, we achieved first place in 19 out of 28 languages in Track A(Table 3), and first place in 10 out of 11 languages in Track B (Table 4).

5 Conclusion

Similar to the emotion detection task discussed in this paper, strongly subjective tasks are prevalent in industry. At both ends of such tasks, on one side, users or annotators have their own subjective judgment criteria, and on the other, language models, due to the nature of their training data, also

[3] https://huggingface.co/google/gemma-2-9b-it

[4] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct

[5] https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501

Track A Dev Dataset

method	eng	ptbr
Gemma-9b-it	0.792	0.667
qwen-2.5-32b-instruct	0.733	0.632
Mistral-Small-24B	0.815	0.672
Deepseek-v3	0.749	0.643
ChatGPT-4o	0.808	0.669
3-layer-nn	0.766	0.647
xgboost	0.826	0.681
lightgbm	0.818	0.677
linear regression	0.809	0.658
vote	0.832	0.688

Table 1: In Track A, the evaluation results on the dev dataset for the F1 score of the strategies trained on the train dataset for the English and Portuguese (Brazil) languages.

Track B Dev Dataset

method	eng	ptbr
Gemma-9b-it	0.812	0.665
qwen-2.5-32b-instruct	0.727	0.635
Mistral-Small-24B	0.782	0.683
Deepseek-v3	0.762	0.603
ChatGPT-4o	0.740	0.668
3 layer nn	0.763	0.644
xgboost	0.826	0.687
lightgbm	0.821	0.680
linear regression	0.784	0.659
vote	0.835	0.706

Table 2: In Track B, the evaluation results on the dev dataset for the Pearson correlation coefficient of the strategies trained on the train dataset for the English and Portuguese (Brazil) languages.

lang	score	lang	score	lang	score
afr	0.698	amh	0.647	aqr	0.668
ary	0.629	chn	0.709	deu	0.739
eng	0.823	esp	0.848	hau	0.750
hin	0.919	ibo	0.600	kin	0.657
mar	0.884	orm	0.581	pcm	0.674
ptbr	0.683	ptmz	0.547	ron	0.794
rus	0.882	som	0.576	sun	0.541
swa	0.384	swe	0.626	tat	0.845
tir	0.538	ukr	0.725	vmw	0.255
yor	0.461				

Table 3: The F1 score of our system on the Track A test dataset.

lang	score	lang	score	lang	score
amh	0.6464	arq	0.6497	chn	0.7224
deu	0.7657	eng	0.8404	esp	0.8080
hau	0.7700	ptbr	0.7100	ron	0.7260
rus	0.9254	ukr	0.7075		

Table 4: The Pearson correlation coefficient of our system on the Track B test dataset.

develop their own judgment standards. This results in biases and gaps between the two. With the emergence and development of large language models (LLMs), and owing to their powerful capabilities, industry applications are increasingly inclined to use untrained models or those only lightly fine-tuned. Therefore, there is a need to explore suitable methods to replace the traditional approach of fitting task labels by training language models extensively. Considering the differences in subjective biases across different large language models, and the generally high accuracy of these models, we were inspired by the concept of Fourier transformations and attempted a ensemble strategy to bridge the gap between these two. From the evaluation results, we observe that the ensemble strategy provides an additional improvement of 0.01 to 0.02 on top of the optimal single-path model's metrics.

In Task 11, the ensemble strategy we employed is based on traditional NLP algorithmic solutions. If similar tasks arise in the future, we aim to explore whether there are applicable solutions within the LLM domain, such as the MoE strategy. Additionally, in the Dev Dataset, we found that transferring the more fine-grained annotation results from Track B to Track A could further improve the performance metrics of Track A. However, due to time constraints, we were unable to test this approach on the Test Dataset, presenting an opportunity for future exploration.

6 Related Work

We select Gemma-9b-it(Gemma Team et al.), Qwen-2.5-32b-Instruct(Qwen et al., 2025), Mistral-Small-24B(noa), DeepSeek-v3(DeepSeek-AI et al., 2024), and ChatGPT-4o as the base models.

Recent advances in emotion detection have primarily focused on two key approaches: leveraging pre-trained large language models (LLMs) (Zhuang et al., 2023; Li et al., 2025) and fine-tuning smaller models for specific tasks (Ren and Sutherland, 2024; Zhang et al., 2023).

Recent studies leverage large, closed-source models like GPT-3 and ChatGPT for zero-shot or few-shot emotion detection, utilizing dynamic prompt generation and optimization to enhance performance without fine-tuning (Amin et al., 2023; Li et al., 2025; Fu et al., 2025).

Techniques like mixture-of-experts (MoE) models and attention-weighted pooling have improved efficiency and accuracy in emotion detection by emphasizing relevant input features (Liu et al., 2024; Zhang et al., 2023).

Traditional ensemble strategies such as XGBoost(Chen and Guestrin, 2016), LightGBM(Ke et al., 2017), and stacking(Ting and Witten, 1997) were applied in our system.

References

Mistral Small 3 | Mistral AI.

Kanhai S. Amin, Linda Mayes, Pavan Khosla, and Rushabh Doshi. 2023. ChatGPT-3.5, ChatGPT-4, Google Bard, and Microsoft Bing to Improve Health Literacy and Communication in Pediatric Populations and Beyond. *arXiv preprint*. ArXiv:2311.10075 [cs].

Tadesse Destaw Belay, Israel Abebe Azime, Abinew Ali Ayele, Grigori Sidorov, Dietrich Klakow, Philip Slusallek, Olga Kolesnikova, and Seid Muhie Yimam. 2025. Evaluating the capabilities of large language models for multi-label emotion understanding. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3523–3540, Abu Dhabi, UAE. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu

Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. DeepSeek-V3 Technical Report. *arXiv preprint*. ArXiv:2412.19437 [cs] version: 1.

Tairan Fu, Javier Conde, Gonzalo Martínez, María Grandury, and Pedro Reviriego. 2025. Multiple Choice Questions: Reasoning Makes Large Language Models (LLMs) More Self-Confident Even When They Are Wrong. *arXiv preprint*. ArXiv:2501.09775 [cs].

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Shreya Pathak, 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 Choquette, 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, Pier Giuseppe Sessa, 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, Clement 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.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Yafu Li, Xuyang Hu, Xiaoye Qu, Linjie Li, and Yu Cheng. 2025. Test-Time Preference Optimization: On-the-Fly Alignment via Iterative Textual Feedback. *arXiv preprint*. ArXiv:2501.12895 [cs].

Ziyue Li and Tianyi Zhou. 2024. Your Mixture-of-Experts LLM Is Secretly an Embedding Model For Free. *arXiv preprint*. ArXiv:2410.10814 [cs].

Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2024. LLMEmb: Large Language Model Can Be a Good Embedding Generator for Sequential Recommendation. *arXiv preprint*. ArXiv:2409.19925 [cs].

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine de Kock, Nirmal Surange, Daniela Teodorescu, Ibrahim Said Ahmad, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino D. M. A. Ali, Ilseyar Alimova, Vladimir Araujo, Nikolay Babakov, Naomi Baes, Ana-Maria Bucur, Andiswa Bukula, Guanqun Cao, Rodrigo Tufino Cardenas, Rendi Chevi, Chiamaka Ijeoma Chukwuneke, Alexandra Ciobotaru, Daryna Dementieva, Murja Sani Gadanya, Robert Geislinger, Bela Gipp, Oumaima Hourrane, Oana Ignat, Falalu Ibrahim Lawan, Rooweither Mabuya, Rahmad Mahendra, Vukosi Marivate, Andrew Piper, Alexander Panchenko, Charles Henrique Porto Ferreira, Vitaly Protasov, Samuel Rutunda, Manish Shrivastava, Aura Cristina Udrea, Lilian Diana Awuor Wanzare, Sophie Wu, Florian Valentin Wunderlich, Hanif Muhammad Zhafran, Tianhui Zhang, Yi Zhou, and Saif M. Mohammad. 2025a. Brighter: Bridging the gap in human-annotated textual emotion recognition datasets for 28 languages. *Preprint*, arXiv:2502.11926.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Seid Muhie Yimam, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine

De Kock, Tadesse Destaw Belay, Ibrahim Said Ahmad, Nirmal Surange, Daniela Teodorescu, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino Ali, Vladimir Araujo, Abinew Ali Ayele, Oana Ignat, Alexander Panchenko, Yi Zhou, and Saif M. Mohammad. 2025b. SemEval task 11: Bridging the gap in text-based emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, Vienna, Austria. Association for Computational Linguistics.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. *arXiv preprint*. ArXiv:2412.15115 [cs].

Yi Ren and Danica J. Sutherland. 2024. Learning Dynamics of LLM Finetuning. *arXiv preprint*. ArXiv:2407.10490 [cs].

Kai Ming Ting and Ian H Witten. 1997. Stacking bagged and dagged models.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning. *arXiv preprint*. ArXiv:2303.10512 [cs].

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023. Open-source Large Language Models are Strong Zero-shot Query Likelihood Models for Document Ranking. *arXiv preprint*. ArXiv:2310.13243.