EMOchecker: An Efficient and Cost-Friendly Affective Judgment Prompting Method for Multimodal LLMs

Tingfu Zhou

University of Michigan, Ann Arbor, USA
Email: tingfu@umich.edu

***Abstract*—Multimodal large language models (MLLMs) represent a promising direction for sentiment analysis. However, training or fine-tuning MLLMs is expensive, which limits their application to specific sub-tasks. As a result, prompt engineering offers a cost-effective alternative. In this paper, I introduce EMOchecker, a prompting method specifically tailored for sentiment analysis to enhance emotional insight and accuracy in MLLMs. Multiple experiments reveal that while EMOchecker provides limited improvement to MLLMs' performance in multimodal sentiment analysis, it significantly enhances performance in text-based sentiment analysis.**

Keywords: Sentiment Analysis, Prompt Engineering, Multimodal Large Language Model

I. Introduction

Sentiment analysis is an important research topic [1], [2]. Initially, sentiment analysis focused primarily on a single modality [3], which limited the completeness and accuracy of affective judgments, as emotional information encompasses text, voice, expressions, and body movements. With the advent of deep learning, researchers have turned their attention to multimodal sentiment analysis [4]. However, most studies focus on the model level, that is, the model used for multimodal sentiment analysis [2], [4]. This results in that when more powerful models become available on the market, GPT-4o [5] for example, these studies quickly become outdated because they cannot be applied to these new models. Fortunately, the rapid advancements in large language models (LLMs) led to the development of multimodal large language models (MLLMs) [6], bringing many innovative models to the market [7]. In this project, I aim to introduce an efficient and cost-effective affective judgment prompting method for these multimodal LLMs.

Previous studies on emotion prompting engineering [8] primarily utilized a single text modality, which can limit the performance of sentiment analysis. In real-life scenarios, emotional information is inherently multimodal. For example, a companion robot equipped with sensors can gather visual and auditory information to accurately assess a user's emotional state, thereby enhancing the user experience. Powerful models like GPT-4o are well-suited for such tasks. However, the large size of MLLMs [5] makes training or fine-tuning them very costly, significantly impacting their application in specific downstream tasks such as sentiment analysis. Thus, employing prompt engineering and utilizing the API of MLLMs can enable accurate emotion assessment while keeping costs manageable.

In this paper, my prompting model synthesizes and modifies existing ideas in prompt engineering and sentiment analysis methodologies. This prompt method offers enhanced emotion judgment in multimodal situations. Furthermore, it leverages powerful foundation models to ensure affordability. The contributions of this project are as follows:

1) I propose EMOchecker, an efficient and cost-effective affective judgment prompting method for multimodal LLMs.
2) Several experiments demonstrate that the improvement of EMOchecker in MLLMs' multi-modal sentiment analysis is limited.
3) Several experiments show that EMOchecker can significantly improve the accuracy of MLLM in text-based sentiment analysis.

II. Related Work

Using multimodal LLM prompt learning for sentiment analysis is an emerging research field. Yu et al. [9] introduced a prompt-based approach to visual perceptual language modeling that leverages a small amount of supervised data. They integrated visual information into an existing language model and employed p-tuning to adapt it for specific downstream tasks. Zhao et al. [10] introduced Memo BERT, a pre-trained model designed for multimodal transformation. This model engages in self-supervised learning using a vast, unlabeled video dataset to conduct multimodal sentiment analysis. Wu et al. [11] utilized trainable templates rather than manual ones and applied an adversarial training method. This approach enables their model to acquire knowledge from various domains, thereby reducing the disparity among domain-specific terminology. Liu et al. [12] introduced an image classification algorithm called PMHANet, which combines pre-trained models with heterogeneous feature alignment. They designed specific strategies for fine-tuning, feature selection, and semantic guidance during the transfer process of pretrained models. Luo et al [13] employ parameter-efficient finetuning to efficiently fine-tune LLMs. Also, they use emotion-cause-aware prompt-based learning and instruction-tuning to improve model performance, enabling large language models (LLMs) to more precisely identify emotions along with their underlying causes.

Most research in multimodal sentiment analysis focuses on model-level methodologies [1], [2], [4]. However, various methods within multimodal sentiment analysis also offer valuable insights. In textual modalities, the use of polar words—terms with positive or negative connotations—helps determine sentiment polarity. In auditory aspects, audio data is crucial for generating accurate transcripts, with key features including pauses. For visual indicators, the combination of visual sentiments with text data helps in emotion analysis, with facial expressions playing a crucial role in the identification of sentiments [14].

III. Preliminaries and Problem Formulation

Prompt engineering for MLLMs can be categorized into three types: multimodal-to-text generation models, image-text-matching models, and text-to-image generation models. This project concentrates on multimodal-to-text generation, which involves creating text-based descriptions or narratives from a blend of input types, such as visual and linguistic information. Regarding the prompting method itself, the prompting method can be divided into two parts that do not overlap: soft prompts and hard prompts. Soft prompts internally add new tokens to the model's architecture, while hard prompts append them to the input [15]. Since this project does not involve the model itself, I will focus on hard prompts in multimodal-to-text generation.

The following are the concepts that need to be used [15]:

Prompt: Additional details or cues supplied to a model to direct its behavior or aid in accomplishing a specific task.

Hard prompts: Hard prompts entail manually created, interpretable text tokens. For example, instead of entering raw input directly, hard prompts can insert "A photo of" before the input for captioning tasks. Hard prompts can be divided into three categories: task instruction prompting, in-context learning, chain-of-thought prompting.

Task Instruction Prompting It employs meticulously crafted prompts that deliver explicit task-related instructions to direct the model's behavior. The formulation for this method can be expressed as $x_{input} = f(x,t)$. Here, f is the task instruction function. x is image. t is the text input. x_{input} is the modified input. In an Instruction Understanding Task, a model receives an input I_x describing the desired output o in natural language. The input I_x comprises a template I, known as instructions, instantiated with a resource x to form I_x. The template may or may not include input-output examples.

A task instruction example is as follows: Suppose a model is asked to summarize a news article. The task instruction prompting could be: "Please summarize the following news article in 3-4 concise sentences, focusing on the main events and their implications." This explicit instruction ensures the model knows exactly what is expected.

In-context Learning It presents the model with a series of related examples or prompts, allowing it to learn and generalize from the given context. The formulation of this method can be expressed as $x_{input} = f(c,x,t)$. Here, f is the task instruction function. And c is the given context. x is image. t is the text input. x_{input} is the modified input.

An in-context learning example is as follows: Suppose a model is tasked with translating sentences from English to Spanish. The in-context learning prompt could be: *"Translate the following sentences from English to Spanish: 1. The weather is beautiful today. → El clima es hermoso hoy. 2. She is reading a fascinating book. →"* By providing example, the model understands the task structure and can generate the correct translation for the second sentence: "Ella está leyendo un libro fascinante."

Chain-of-Thought Prompting It guides the model with a sequence of instructions or questions that gradually build upon one another. Each prompt in the chain adds context or narrows the focus, allowing the model to produce more coherent and contextually appropriate responses. The formulation of this method can be expressed as $f^{l+1} = f^{l}(x,t)$. It is worth to note that f is not the input of CoT. Here, the f represents the prompt function that accepts the image x and text t inputs and guilds the MLLM to generates a response. l is the step index. The output of the l^{th} prompt is the input of the $(l+1)^{st}$ prompt. By progressively building on previous prompts, the iterative nature of the chain-of-thought prompting method aids the model in maintaining coherence and generating responses that align with the evolving context of the conversation.

An chain-of-thought example is as follows: Suppose a model is tasked with solving a math problem: "If a train travels 60 miles in 1 hour, how far will it travel in 3 hours?" A chain-of-thought prompt might be: "1. What is the speed of the train in miles per hour? 2. How many hours is the train traveling? 3. Using the formula Distance = Speed × Time, calculate how far the train travels in 3 hours." The model then combines these steps to conclude: "The train will travel 180 miles in 3 hours."

It is worth noting that these prompting methods are not mutually exclusive, as shown in the Figure 1. they can be combined and utilized together to achieve the desired outcomes across different settings and tasks.

Problem Definition: Given multimodal inputs, comprising both images and text, the task is to design effective prompts and utilize the API of Multimodal Large Language Models (MLLMs) to analyze and interpret the emotional information contained within these inputs.

IV. EMOchecker

Due to the nature of my task—efficient and cost-effective affective judgment—I will not cover the model itself, since training and fine-tuning an MLLM is expensive. Moreover, existing MLLMs' APIs do not support model parameter modification. Thus, soft prompt methods are not suitable for this situation as

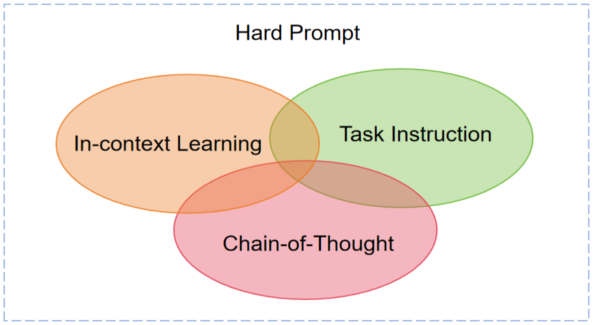

Fig. 1: Prompting Methods

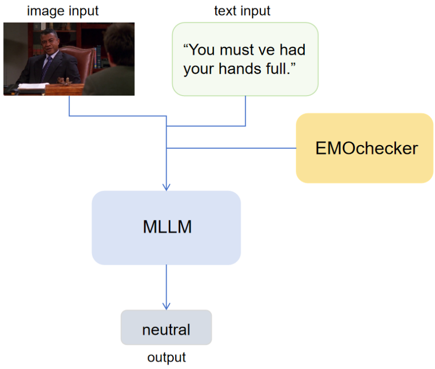

Fig. 2: Prompt Process

it requires modifications to the model itself. I use hard prompt methods which do not involve the model itself, as shown in the Figure 2. The prompted model, called EMOchecker, combines in-context learning, task instruction, and chain-of-thought techniques. In the task instruction component, I assign the API the role of a sentiment analyst. This role defines the emotional analysis task, with a clear and direct instruction for performing sentiment analysis according to specific steps and producing output in a specified format. The chain-of-thought method is used to guide the model through a structured reasoning process. An example is provided within the in-context learning component. The specific prompt is shown in Figure 3.

At the beginning, the task instruction, highlighted in blue, assigns a sentiment analysis role to the MLLM and establishes the task context. This is followed by the context for entering information. The chain-of-thought, shown in orange, guides the MLLM through a sequence of instructions. The pink section combines chain-of-thought with task instruction, breaking down sentiment analysis into four detailed steps: first capturing emotional characteristics in the image, then in the text, followed by combining both sources, and finally making a classification judgment.

In-context learning, shown in purple, provides an example of sentiment analysis, establishing the final output format. The black section further specifies the output format. Clearly defining the output format helps prevent format inconsistencies in the MLLM's responses, so this format requirement is reiterated here.

EMOchecker

You will be performing sentiment analysis based on a scene description and conversation text. Your task is to interpret the speaker's emotions from the provided information and classify them into a single emotion category.
Assume that the image is the scene where the user is having a conversation, and the person in the picture is the user. Text is the content of user conversations. You need to analyze user sentiment according to the following steps.
1. Examine the scene description: Carefully consider the scene description. Look for indicators of emotion such as facial expressions, body language, and environmental factors that might influence the speaker's emotional state.
2. Examine the conversation text: Examine the conversation text for words, phrases, or tone that suggest the speaker's emotional state. Pay attention to the content of what is being said and how it is expressed.
3. Combine your analysis: Integrate your observations from both the visual and textual cues to form a comprehensive understanding of the speaker's emotional state.
4. Classify the emotion: Based on your analysis, determine the most prominent emotion expressed. Choose from the following categories: - joy - sadness - anger - fear - surprise - disgust - neutral
Output your final emotion classification as a single word, without any additional explanation or justification. For example, if you determine the speaker is expressing happiness, your entire output should be:
joy
Remember, you are only to output the emotion category. Do not provide any additional commentary, explanation, or answer to any other questions.

Fig. 3: Prompt for EMOchecker task.

V. Performance Evaluation

A. Experimental Setup and Datasets

The experimental setup is divided into two parts: the APIs of MLLMs and a multimodal emotion dataset. Currently, there are few large models offering multimodal APIs. For this study, I selected two mainstream MLLM APIs: the GPT-4o API [1] and the Claude 3.5 Sonnet API [2]. Both MLLMs support image and text inputs.

[1] https://openai.com/index/hello-gpt-4o/
[2] https://docs.anthropic.com/en/home

For the multimodal emotion dataset, I chose the MELD dataset [16] and the Memotion Dataset 7k [3]. The MELD dataset contains over 1,400 dialogues and 13,000 utterances from the TV series "Friends," featuring multiple speakers. Each utterance is labeled with one of seven emotions: Anger, Disgust, Sadness, Joy, Neutral, Surprise, or Fear. MELD also provides sentiment annotations (positive, negative, and neutral) for each utterance. The Memotion Dataset 7k includes 8,000 annotated memes, each labeled by human annotators with sentiment information and humor types, such as sarcasm, humor, or offensiveness.

It is worth noting that manually labeled multimodal sentiment analysis datasets (incorporating images, videos, and text) are relatively rare due to the high cost of annotation. Therefore, although the Memotion Dataset 7k focuses on internet memes, it is included here to test EMOchecker's robustness, given its similarities to conversational content.

B. Comparison Experiments

The comparison experiments evaluate the performance of EMOchecker against its control group. The control group consists of the original API with additional restrictions on the output format. Apart from the prompt, all other parameters are set to default values. In the following bar charts 4, 5, 6, 7, 8, the API with EMOchecker is denoted as EMO, while the control group is denoted as API. Each experiment uses 100 dialogue data, and the results are shown in Figure 4.

For ChatGPT 4.5o, EMOchecker achieved an accuracy of 65%, while the control group achieved an accuracy of 64%. For Claude 3.5 Sonnet, EMOchecker reached an accuracy of 47%, compared to 42% for the control group. In both cases, EMOchecker has higher accuracy than the control group. However, the difference was not statistically significant at $\alpha = 0.05$. Also, when studying specific cases, you can find that in some cases EMOchecker is correct, but the control group is wrong, while in other cases the opposite is true. For example, in the case "My duties? All right.", the correct answer is surprise, the EMOchecker answer joy, and the control group answer surprise. In other case, like "You or me?", the correct answer is neutral, EMOchecker answer neutral, and control group answer joy. Additionally, in the performance experiments of the model itself, ChatGPT 4.5o statistically outperformed Claude 3.5 Sonnet. The gap between EMOchecker and the control group was also notably wider for Claude 3.5 Sonnet than for ChatGPT 4.5o.

These findings suggest that the performance improvement of EMOchecker is limited to more powerful base models. However, for weaker baseline models, the improvement is more pronounced. We can reasonably infer that for weaker models, EMOchecker can significantly enhance performance.

In addition to emotional analysis, a positive, negative, and neutral classification was conducted. This classification is simpler and does not require identifying specific emotions, instead focusing on broader sentiment categories (positive, negative, neutral). Each experiment uses 100 dialogue data, and the results are shown in Figure 5. For ChatGPT 4.5o, the accuracy of both EMOchecker and the control group was 67%. For Claude 3.5 Sonnet, the accuracy of both EMOchecker and the control group was 45%.

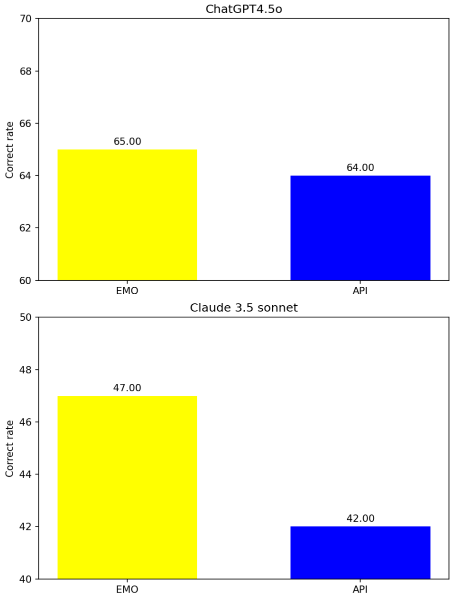

Fig. 4: Emotional Analysis

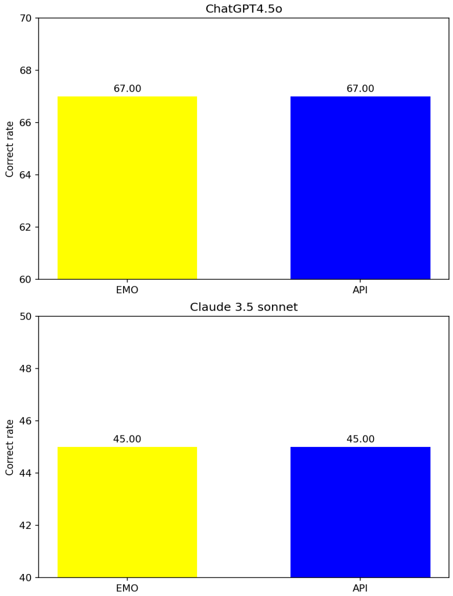

Fig. 5: Positive, Negative, Neutral Analysis

These results indicate that EMOchecker does not show a significant improvement in positive, negative, and neutral classification. This outcome is likely because EMOchecker is specifically optimized for detailed emotional analysis.

It is worth noting that the accuracy of both EMOchecker and the control group was low in this context. To investigate whether this low accuracy is due to the challenges of

[3] https://www.kaggle.com/datasets/williamscott701/memotion-dataset-7k

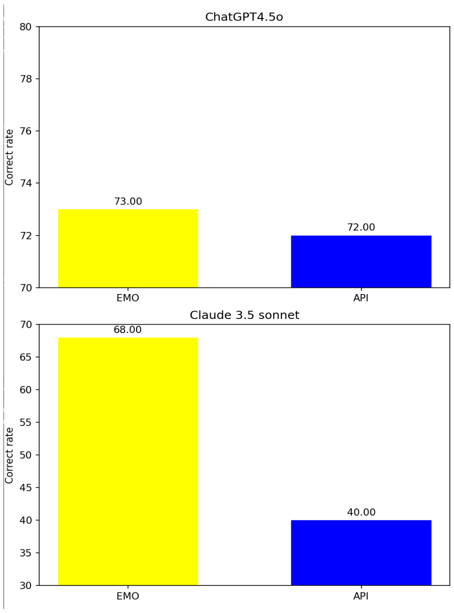

Fig. 6: Text Only Analysis

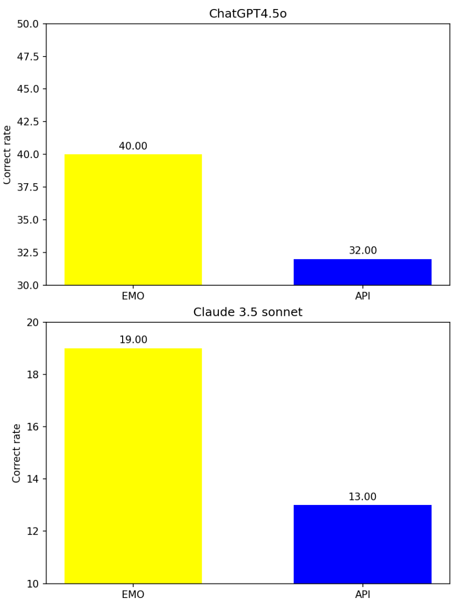

Fig. 7: Image Only Analysis

multimodal data, comparison experiments were conducted for emotional analysis using only text and only images.

The results of the text-only analysis are shown in Figure 6. For ChatGPT 4.5o, the accuracy of EMOchecker was 73%, while the accuracy of the control group was 72%. For Claude 3.5 Sonnet, the accuracy of EMOchecker was 68%, compared to 40% for the control group. These results indicate that in the text-only case, EMOchecker can significantly improve the accuracy of weaker baseline models. However, for stronger baseline models, EMOchecker's improvement is limited.

The results of the image-only analysis are shown in Figure 7. For ChatGPT 4.5o, the accuracy of EMOchecker was 40%, while the control group achieved an accuracy of 32%. For Claude 3.5 Sonnet, EMOchecker reached an accuracy of 19%, compared to 13% for the control group. In the image-only case, EMOchecker slightly improves the performance of both weaker and stronger baseline models. In the specific experiment case, EMOchecker judged some conversations correctly, while the control group did not. However, the improvement is not statistically significant.

From the results of text-only analysis (Figure 6), image-only analysis (Figure 7), and multimodal data analysis (Figure 4), we observe that the performance of text-only analysis generally exceeds that of multimodal analysis under the same conditions. In the case of a weaker baseline model, the gap between text-only and multimodal data analysis is statistically significant. Conversely, the performance of image-only analysis is significantly lower than that of multimodal analysis under comparable conditions.

These findings suggest that existing MLLMs perform best in text analysis, while accuracy remains lower in image analysis. I assume that this is related to the characteristics of transformer-based MLLMs, as image processing requires a separate encoder for the attention mechanism. However, this assumption is outside the scope of the current project.

Based on the results of the above experiments, it can be concluded that EMOchecker does not significantly improve the performance of MLLMs in multimodal emotional analysis, with the degree of improvement largely dependent on the baseline model. The enhancement is more evident for weaker baseline models. EMOchecker also significantly improves performance in text-only emotional analysis. While the improvement in image-only emotional analysis is not statistical significant. In positive-negative-neutral experiments, the baseline model itself appears to be a more influential factor on accuracy than the prompt.

C. *Compatibility Experiments*

Compatibility experiments were conducted using the Memotion Dataset 7k. It is worth noting that a meme is a different type of multimodal communication. A meme consists of pictures and text on those pictures. It is commonly used in online conversations. However, the content of a meme is different from the content (i.e., image and text) of an offline conversation. In a classic offline conversation, the image represents the dialogue situation of the speakers, and the text conveys the conversation content. In a meme, the image and text do not come from the conversation; instead, speakers use memes to express their feelings. Thus, a meme is different from the default application dialogue scene. Therefore, I chose to use memes to test the model's compatibility with varying conversational content. Each experiment uses 100 multimodal memes, and the results are shown in Figure 8.

For ChatGPT 4.5, EMOchecker achieved an accuracy of 22%, whereas the control group reached 18%. For Claude 3.5 Sonnet, EMOchecker achieved an accuracy of 21%, compared to 18% for the control group. In both cases, the accuracy

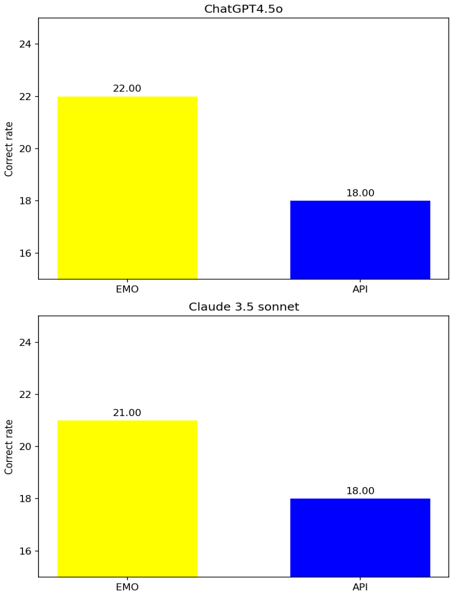

Fig. 8: Compatibility Experiments

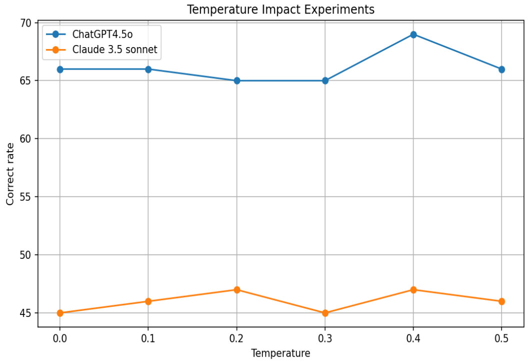

Fig. 9: Temperature Impact Evaluation

of EMOchecker was slightly higher than that of the control group; But, this difference was not statistically significant at $\alpha = 0.05$. Additionally, the accuracy in the compatibility experiment was generally lower for both groups compared to the normal emotion analysis results (Figure 4).

From the above results, we know that EMOchecker's improvement in the accuracy of emotion recognition of memes is not obvious. However, this limit is largely because of the low accuracy of the baseline model.

D. Temperature Impact Evaluation

In the API of LLMs, there is a parameter called temperature. This parameter controls the "randomness" and "creativity" of the generated responses. The temperature ranges from 0 to 1: higher values lead to more diverse and unpredictable responses, incorporating variations in phrasing, while lower values yield more deterministic outputs that closely follow the most likely phrasing and answers. In the temperature impact experiments, I set the temperature to 0, 0.1, 0.2, 0.3, 0.4, and 0.5. Each experiment uses 100 dialogue data, and the results of these experiments are shown in Figure 9.

From the results of the temperature impact experiments, we observe that the accuracy remains generally stable. Different baseline models exhibit distinct accuracy trends as the temperature changes. For ChatGPT 4.5o, accuracy initially decreases, then increases, and subsequently decreases again as the temperature rises, reaching the highest accuracy at a temperature of 0.4. For Claude 3.5 Sonnet, accuracy follows an M-shaped trend as the temperature increases, with peaks at 0.2 and 0.4.

It is important to note that if the temperature is increased further, the model's response time becomes excessively long, which is why the upper temperature limit is set at 0.5. This phenomenon can be explained by the inherent randomness of the baseline model and its originally lower accuracy. Hence, a moderate increase in temperature can lead to slight improvements in EMOchecker's accuracy. Since the default temperature value is 0.2, typically no adjustments are needed.

VI. Conclusion

In this paper, I introduce EMOchecker, a prompt specifically designed for sentiment analysis to enhance emotional understanding and accuracy for MLLMs. Multiple experiments demonstrate that EMOchecker has limited improvement for MLLMs' performance in multimodal sentiment analysis. Additionally, EMOchecker significantly boosts performance in text-based sentiment analysis. It is worth noting that the price of an MLLM API is significantly higher than that of a text-based LLM, while the accuracy of mainstream MLLMs on the market in multimodal sentiment analysis is lower than that of text-based sentiment analysis. Therefore, for commercialization, using text-based LLMs for sentiment analysis is a more cost-effective choice.

References

[1] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, and A. Hussain, "Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions," *Information Fusion*, vol. 91, pp. 424–444, 2023.

[2] R. Das and T. D. Singh, "Multimodal sentiment analysis: a survey of methods, trends, and challenges," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–38, 2023.

[3] G. Chandrasekaran, T. N. Nguyen, and J. Hemanth D, "Multimodal sentimental analysis for social media applications: A comprehensive review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 5, p. e1415, 2021.

[4] S. Lai, X. Hu, H. Xu, Z. Ren, and Z. Liu, "Multimodal sentiment analysis: A survey," *Displays*, p. 102563, 2023.

[5] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[6] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A survey on multimodal large language models," *arXiv preprint arXiv:2306.13549*, 2023.

[7] openAI. (2022) Introducing chatgpt https://openai.com/index/chatgpt/. [Online]. Available: https://openai.com/index/chatgpt/

[8] C. Li, J. Wang, Y. Zhang, K. Zhu, W. Hou, J. Lian, F. Luo, Q. Yang, and X. Xie, "Large language models understand and can be enhanced by emotional stimuli," *arXiv preprint arXiv:2307.11760*, 2023.

[9] Y. Yu and D. Zhang, “Few-shot multi-modal sentiment analysis with prompt-based vision-aware language modeling,” in *2022 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2022, pp. 1–6.

[10] J. Zhao, R. Li, Q. Jin, X. Wang, and H. Li, “Memobert: Pre-training model with prompt-based learning for multimodal emotion recognition,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4703–4707.

[11] H. Wu and X. Shi, “Adversarial soft prompt tuning for cross-domain sentiment analysis,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 2438–2447.

[12] J. Liu, J. Xiao, H. Ma, X. Li, Z. Qi, X. Meng, and L. Meng, “Prompt learning with cross-modal feature alignment for visual domain adaptation,” in *CAAI International Conference on Artificial Intelligence*. Springer, 2022, pp. 416–428.

[13] M. Luo, H. Zhang, S. Wu, B. Li, H. Han, and H. Fei, “Nus-emo at semeval-2024 task 3: Instruction-tuning llm for multimodal emotion-cause analysis in conversations,” in *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 2024, pp. 1599–1606.

[14] A. Gandhi, K. Adhvaryu, and V. Khanduja, “Multimodal sentiment analysis: review, application domains and future directions,” in *2021 IEEE Pune section international conference (PuneCon)*. IEEE, 2021, pp. 1–5.

[15] J. Gu, Z. Han, S. Chen, A. Beirami, B. He, G. Zhang, R. Liao, Y. Qin, V. Tresp, and P. Torr, “A systematic survey of prompt engineering on vision-language foundation models,” *arXiv preprint arXiv:2307.12980*, 2023.

[16] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, “Meld: A multimodal multi-party dataset for emotion recognition in conversations,” *arXiv preprint arXiv:1810.02508*, 2018.