EmoAgent: A Dual-Agent Framework for Short Text Emotion Classification based on Retrieval-Augmented Generation and Chain-of-Thought Reasoning

Zixiang Qi
Guangdong University of Science and Technology, Dongguan, Guangdong, 523668, China
qixiangzi@foxmail.com

ABSTRACT

Emotion classification in short texts is still an important but very difficult task in emotion calculation because of the lack of contextual clues and implied semantic expressions. In order to solve the problem of large-scale language model (LLM) in a domain-specific zero-shot scene, we propose a new dual-agent framework named EmoAgent, which includes a search enhanced generation (RAG) agent and a question query (QA) agent. Specifically, as a knowledge extractor, RAG Agent looks for similar examples in a structured database, while QA Agent uses the Chain-of-Thought (CoT) hint to infer from the discovered context step by step. We created a balanced emotional data set to test the framework using DeepSeek and Doubao models. Many experimental results show that EmoAgent improves the classification accuracy from 37.6% to 80.7%, which proves that it is better to combine sparse search method with deep reasoning mechanism.

Keywords: Emoagent; Short Text Emotion Classification; Retrieval-Augmented Generation; Chain-Of-Thought Reasoning; Sparse Retrieval.

1. INTRODUCTION

Emotional computation and emotional analysis are the important pillars of modern natural language processing (NLP). They help to do things such as robots responding to customers, monitoring people's thoughts, and studying society with data [1-2]. These uses include the classification of emotions-classifying texts into categories such as joy, fear, anger and sadness-which is an area we are working on [3]. However, it is difficult to analyze short texts (such as social media posts): they are often vague, have no context, and hide many meanings, so it is difficult to mark and understand them [4].

Recently, Large Language Model (LLM) has changed NLP. Although they are very large and can be learned without examples [5], they encounter difficulties in strict category restrictions. Research shows that if there is no clear instruction or context, the generated zero-beat model may not be as good as the specially trained small model [6]. This problem mainly comes from illusion and lack of connection with a specific theme in the case of zero shooting.

In order to fill this gap without using expensive fine-tuning, this paper introduces EmoAgent, a complete dual-agent framework. Instead of using LLM as a monolithic zero-beat predictor, we decomposed the cognitive process into two cooperative autonomous agents, which quickly became the frontier paradigm of LLM research [13-15].

Specifically, our framework consists of RAG (Search Enhanced Generation) agent and QA (Problem Solving) agent. RAG agent handles external knowledge retrieval to provide a high-quality context framework. Then, QA agent processes this information by using Chain-of-Thought (CoT) reasoning to make the final emotion classification.

The main contributions of this paper are as follows:

1. We provide EmoAgent, a powerful dual-agent system, which uses RAG agent and QA agent to classify emotions into short texts.
2. We build and clean up a highly balanced and specialized data set to test the performance of LLM.
3. We compare different models, CoT strategies and search algorithms (TF-IDF and BM25) seriously.

2. RELATED WORK

2.1 LLMs in Text Classification

Traditional text classification methods use a large number of deep learning frameworks and pre-training language models. Recently, people are more interested in using LLM [7]. Although the use of LLM for zero-shot classification shows good generalization [8], researchers say that only using context learning or basic self-training may not be enough to meet complex semantic restrictions [9-10]. Our work solves this problem by adding dynamic external knowledge to help LLM's beat boundary, rather than static fine adjustment.

2.2 Agentic LLMs and Chain-of-Thought Reasoning

The NLP paradigm is changing: we are moving from just-in-time engineering to autonomous LLM agents, who perform complex multi-step tasks [14]. At the same time, technologies such as CoT and hierarchical reasoning help the model to write logical steps before giving the final answer, thus making the explanation and reasoning better [11-12]. By using a structure with two agents, our framework directly applies CoT to evaluate the semantic context we found, which makes the logic more consistent.

2.3 Retrieval-Augmented Generation (RAG)

RAG helps to reduce the problem of "illusion" and uses externally verifiable knowledge [16-17]. Recent research shows that using RAG is often more reliable and cheaper than fine adjustment, especially for finding rare or less well-known knowledge [19-20]. Although semantic intensive search is common, search methods such as BM25 and TF-IDF are still very good at finding accurate words and mixed words [18]. In this study, we test how these search algorithms work as part of agenttic RAG for short texts.

3. METHODOLOGY

3.1 The EmoAgent Architecture and Workflow

Definition of agent: In this study, the word "agent" does not mean that a system can do anything by itself, with endless plans, lasting memories, tools found by itself, or discussions with others. No, the agent here means a special part in a fixed path to find and think. For example, the RAG agent is responsible for selecting evidence from the marked knowledge base, while the QA agent is responsible for thinking according to the instructions and expressing the final emotion. This definition shows that EmoAgent is a two-part framework, which comes from the idea of agent, but it is not an autonomous agent system that can do everything.

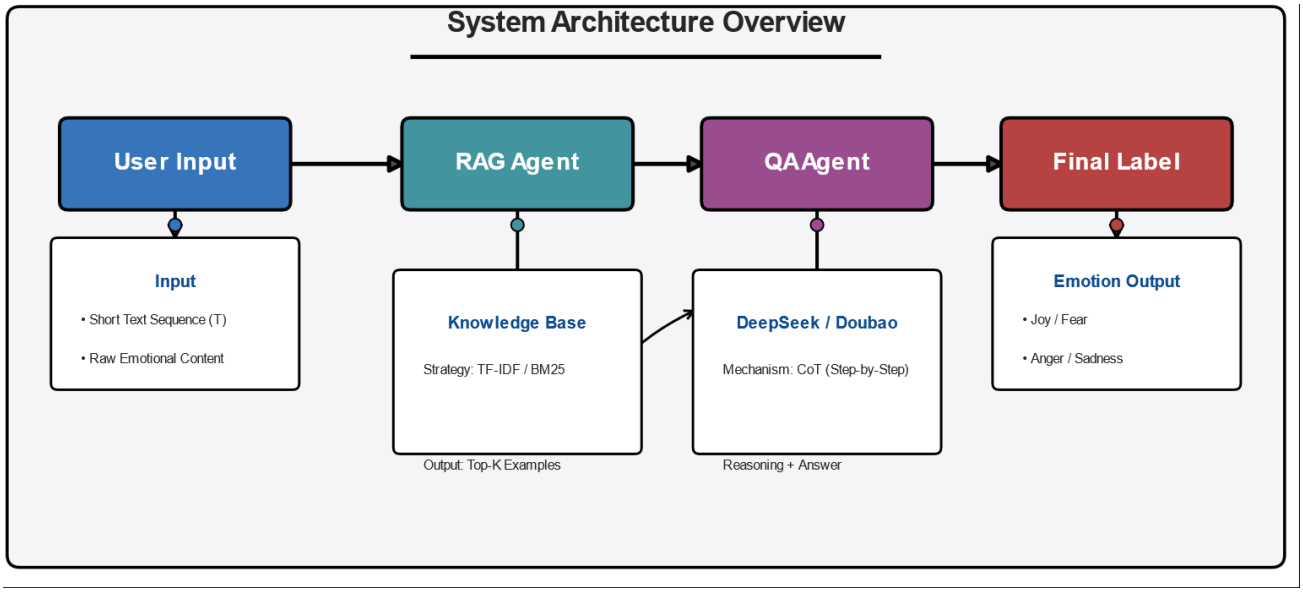

Figure 1. The proposed Dual-Agent framework integrating the RAG Agent and QA Agent.

Designer EmoAgent is more than just an exercise to connect different LLM API. On the contrary, the framework aims to reduce the unpredictability of the generated model through controlled division of labor. As shown in Figure 1, the core idea is to separate evidence retrieval from tag reasoning. In the above operation definition, the dual agent design refers to two limited functional roles: RAG agent retrieves examples of related tags, and QA agent uses the retrieved context for forced reasoning. Therefore, the data flow is organized as a fixed pipeline, which is divided into four stages:

1. Query input: The pipeline starts immediately after receiving a small sentence with emotion. Unlike long documents, these items don't have much context, so the first point is very important.
2. Lexical Knowledge Retrieval: RAG Agent is a simple component for searching words, and important examples will be found in the emotional knowledge base. It turns the question into a keyword list and uses the most similar Top-K example. In our system, this component does not draw plans or select tools by itself. It uses TF-IDF or BM25 instead of semantic embedding.
3. Prompt Assembly: not just putting data into the reasoning engine. The system is like an intelligent assembler: it combines our strict instructions, Top-K examples and user questions to make a single prompt package.
4. Cognitive Reasoning: Finally, QA Agent will collect the tips of this combination. He didn't give a quick answer, but did a step-by-step analysis to find the right emotional label.

3.2 The RAG Agent as a Bounded Retrieval Component: Sparse Lexical Retrieval Mechanisms

According to the definition in Section 3.1, the RAG proxy must be understood as a bounded retrieval component, not an independent proxy. His job is to reduce the lack of context in the essay by looking for marked examples that are very similar to the questions raised; Because the quality of research directly affects the speed of using QA agents, it is important to build and prepare the knowledge base of the whole classification process well. Before searching, every document in the knowledge base is marked and noise elements such as special characters are deleted.

Because the behavior of RAG agent is completely determined by search algorithm and Top-K selection rule, we compare two classical word-for-word search methods, TF-IDF and BM25, to see which mechanism provides the best context marker for emotion classification in short texts.

TF-IDF Implementation:

In TF-IDF, each query and each candidate example is represented as a weighted and clear term vector. Term weights are calculated by $ term frequency and inverse document frequency, which together estimate the importance of a term in a document relative to the whole corpus. For a given item t and document d, the formula is:

$$\mathrm{TF}(\mathrm{t},\mathrm{d}) = \frac{\mathrm{f}_{\mathrm{t},\mathrm{d}}}{\sum_{\mathrm{t}'\in\mathrm{d}} \mathrm{f}_{\mathrm{t}',\mathrm{d}}} \tag{1}$$

$$IDF(t,D) = \log\left(\frac{N}{|\{d\in D: t\in d\}|}\right) \tag{2}$$

Here, tf(t,d) represents the total number of words t in document D, and n is the total number of documents in the corpus. The final lexical similarity score is calculated by scalar product of TF-IDF vector. The search component classifies the candidate examples according to their TF-IDF similarity scores, and selects the best example as the lexical context location of QA Agent.

BM25 Implementation:

We also use Okapi BM25 algorithm as another sparse vocabulary search method. BM25 improves simple word frequency matching by increasing term frequency saturation and standardizing document length. The score function of query Q and document D is defined as:

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{avgdl}\right)} \tag{3}$$

In this equation, |d| represents the length of the document, and avgdl represents the average length of the document in the data set. Parameters k1 and b are adjustable constants. They control the standardization of saturation and length of items. After scoring all the candidate examples, the retrieval component will select the three best examples that match lexically and format them as the context blocks of the QA agent. This process is controlled by algorithms, and does not include autonomous scheduling, memory update or multi-round coordination.

Therefore, the word "agent" in RAG agent refers to its special function of finding evidence in the fixed pipeline of EmoAgent, rather than an independent agent like LLM agent autonomous system.

3.3 The QA Agent: Prompt Engineering and Chain-of-Thought

QA agent is a limited reasoning part that uses DeepSeek or Doubao. As mentioned in Section 3.1, he will not plan or coordinate with other agents alone. Instead, it follows a fixed prompt pattern with Chain-of-Thought (CoT) examples and instructions. We use this design because directly asking LLM to find emotion from a few sentences may give too fast or wrong predictions.

We divide the internal prompt into three specific functional modules:

- Role setting and constraints: First, we pretend that LLM is an expert in psychological and emotional aspects. Most importantly, we set very strict limits on what he can say, forcing him to choose only four categories (happiness, fear, anger and sadness).
- In-context learning (ICL) injection: inject the context block given by the RAG agent. These small examples are like a guide to the model's behavior, which can help it adjust its classification boundaries to match our data set distribution.
- Step-by-Step Execution Directive: But here is the real catch. Instead of allowing a direct classification jump from input X to output Y $(i.e., mapping\ X \rightarrow Y)$, we force a computational detour: $X \rightarrow R \rightarrow Y$

That variable R represents the intermediate reasoning process. Under this very strict rule, the agent must write a paragraph of reasoning process before doing anything else. He must analyze the tone of words and find important themes in the text. Why does it work? By forcing the model to create a token for R reasoning first, we shift our attention to a richer context mechanism than it created itself to better determine the final Y tag.

4. EXPERIMENTS AND RESULTS

4.1 Dataset Construction and Preprocessing

In order to ensure a fair and strict evaluation, we obtain raw data from emotional texts and strictly clean up and annotate them. We deleted incomplete sentences, repeated and unclear text.

The final data set is well organized and perfectly balanced. It contains 1200 test samples, and 300 samples are selected for each emotion category (happiness, fear, anger, sadness). This uniform distribution can prevent the model from benefiting from imbalance and ensure the real ability of the evaluation index display framework.

4.2 Experimental Setup

We use two advanced back-end models to evaluate our framework: DeepSeek and Doubao. The evaluation includes various configurations: the combination of QA agent (with or without CoT/Step-by-step reasoning) and RAG agent (No RAG, TF-IDF and BM25). We measure accuracy, accuracy, recall and F1-Score.

4.3 Experimental Results

All experimental data are summarized in Table 1. The result includes all 12 configurations, which enables us to compare the baseline zero shooting with the improvement of double agent (CoT+RAG).

Table 1. Performance comparison of different models and retrieval strategies.

Model	Strategy	Accuracy	Precision	Recall	F1-Score
Doubao (step-by-step)	TF-IDF RAG	0.8079	0.8154	0.8079	0.8074
Doubao (step-by-step)	BM25 RAG	0.8054	0.8147	0.8054	0.8049
Doubao (step-by-step)	No RAG (Direct)	0.5121	0.5636	0.5121	0.4909
Doubao	TF-IDF RAG	0.8071	0.8147	0.8071	0.8062
Doubao	BM25 RAG	0.8054	0.8163	0.8054	0.805
Doubao	No RAG (Direct)	0.3767	0.5401	0.3767	0.3271
DeepSeek (step-by-step)	TF-IDF RAG	0.7854	0.7929	0.7854	0.7846
DeepSeek (step-by-step)	BM25 RAG	0.7708	0.781	0.7708	0.77
DeepSeek (step-by-step)	No RAG (Direct)	0.5204	0.5558	0.5204	0.4976

Table 1. (continued) Performance comparison of different models and retrieval strategies.

Model	Strategy	Accuracy	Precision	Recall	F1-Score
DeepSeek	TF-IDF RAG	0.7821	0.79	0.7821	0.7812
DeepSeek	BM25 RAG	0.7742	0.7842	0.7742	0.7732
DeepSeek	No RAG (Direct)	0.4621	0.5507	0.4621	0.4297

4.4 Experimental Analysis

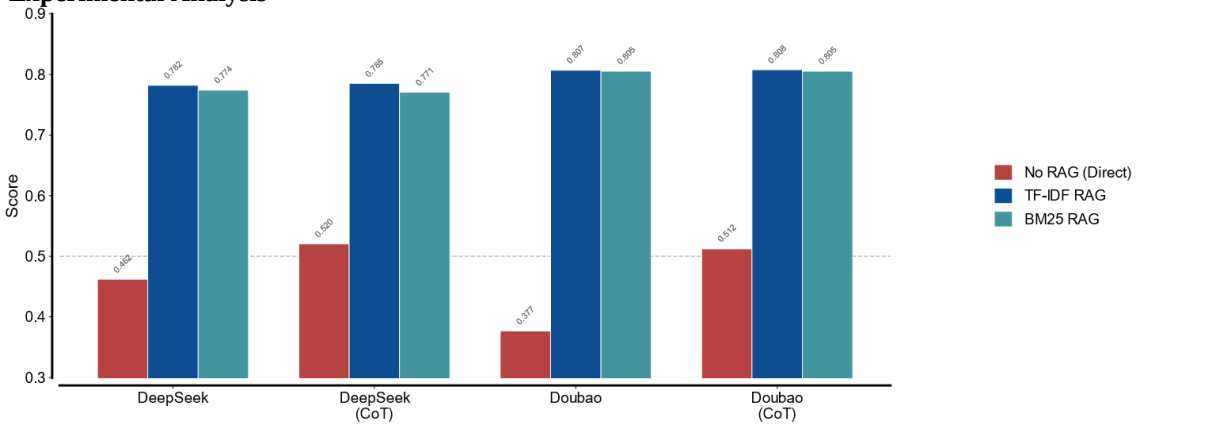

Figure 2. Performance comparison across different models and retrieval strategies.

Based on the results reported in Table 1, we analyze the effects of CoT reasoning, sparse retrieval methods, and backend models.

Effect of Chain-of-Thought Reasoning

When retrieval is not used, CoT prompting will produce better results than direct beat classification. For Du Bao, the accuracy of CoT tips increased from 37.67% to 51.21%, an absolute increase of 13.54 percentage points. The accuracy of DeepSeek increased from 46.21% to 52.04%, an absolute increase of 5.83 percentage points. These results show that in the zero-shot framework of the test, CoT reasoning improves the classification performance by pushing the model to make a clear intermediate analysis before giving the final answer.

Effect of Sparse Retrieval Methods

Adding the retrieval part is the biggest improvement in the test configuration. For Du Bao, when using TF-IDF-based retrieval and CoT prompt, the accuracy is improved from 37.67% (direct zero-beat baseline) to 80.79%. Comparing these two sparse retrieval methods, TF-IDF is slightly better than BM25 in Dubao CoT setting, with accuracy rates of 80.79% and 80.54% respectively. This slight difference shows that the standardization of BM25 document length does not provide additional advantages compared with TF-IDF for this short text data set.

Model-Level Comparison: DeepSeek vs. Doubao

Using retrieval plus CoT configuration, the accuracy of Doubao is slightly higher than that of DeepSeek:Doubao is 80.79%,DeepSeek is 78.54%. However, in the direct zero-beat test without backtracking and CoT, the accuracy of DeepSeek is 46.21%, while that of Doubao is 37.67%. This shows that DeepSeek does better in direct zero-beat testing, while Doubao benefits more from the examples found in the retrieval enhancement configuration of the test.

5. CONCLUSION

In this paper, we put forward EmoAgent, which is a double-agent framework, using RAG agent and QA agent to classify emotions into short texts. We tested our ideas in 1200 examples of DeepSeek and Doubao. Our experience shows that with the help of external knowledge and Chain-of-Thought reasoning, the accuracy is improved from less than 50% to more than 80%. EmoAgent provides a very clear, powerful and fair solution for LLM text classification.

REFERENCES

[1] Kusal S, Patil S, Choudrie J, et al. A systematic review of applications of natural language processing and future challenges with special emphasis in text-based emotion detection[J]. Artificial Intelligence Review, 2023, 56(12): 15129-15215.

[2] Alam M S, Mrida M S H, Rahman M A. Sentiment analysis in social media: How data science impacts public opinion knowledge integrates natural language processing (NLP) with artificial intelligence (AI)[J]. American Journal of Scholarly Research and Innovation, 2025, 4(01): 63-100.

[3] Alslaity A, Orji R. Machine learning techniques for emotion detection and sentiment analysis: current state, challenges, and future directions[J]. Behaviour & Information Technology, 2024, 43(1): 139-164.

[4] Krommyda M, Rigos A, Bouklas K, et al. An experimental analysis of data annotation methodologies for emotion detection in short text posted on social media[C]//Informatics. MDPI, 2021, 8(1): 19.

[5] Chae Y, Davidson T. Large language models for text classification: From zero-shot learning to instruction-tuning[J]. Sociological Methods & Research, 2026, 55(2): 501-567.

[6] Bucher M J J, Martini M. Fine-tuned 'small' LLMs (still) significantly outperform zero-shot generative AI models in text classification[J]. arXiv preprint arXiv:2406.08660, 2024.

[7] Sun X, Li X, Li J, et al. Text classification via large language models[C]//Findings of the Association for Computational Linguistics: EMNLP 2023. 2023: 8990-9005.

[8] Gretz S, Halfon A, Shnayderman I, et al. Zero-shot topical text classification with LLMs-an experimental study[C]//Findings of the Association for Computational Linguistics: EMNLP 2023. 2023: 9647-9676.

[9] Gera A, Halfon A, Shnarch E, et al. Zero-shot text classification with self-training[C]//Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. 2022: 1107-1119.

[10] Edwards A, Camacho-Collados J. Language models for text classification: Is in-context learning enough? [C]//Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). 2024: 10058-10072.

[11] Sanwal M. Layered chain-of-thought prompting for multi-agent llm systems: A comprehensive approach to explainable large language models[J]. arXiv preprint arXiv:2501.18645, 2025.

[12] Chen Q, Qin L, Liu J, et al. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models[J]. arXiv preprint arXiv:2503.09567, 2025.

[13] Barua S. Exploring autonomous agents through the lens of large language models: A review[J]. arXiv preprint arXiv:2404.04442, 2024.

[14] Zhang Z, Yao Y, Zhang A, et al. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents[J]. ACM Computing Surveys, 2025, 57(8): 1-39.

[15] Plaat A, van Duijn M, Van Stein N, et al. Agentic large language models, a survey[J]. Journal of Artificial Intelligence Research, 2025, 84.

[16] Sharma C. Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers[J]. arXiv preprint arXiv:2506.00054, 2025.

[17] Chen L C, Pardeshi M S, Liao Y X, et al. Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model[J]. Computer Standards & Interfaces, 2025, 94: 103995.

[18] Sawarkar K, Mangal A, Solanki S R. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers[C]//2024 IEEE 7th international conference on multimedia information processing and retrieval (MIPR). IEEE, 2024: 155-161.

[19] Friel R, Belyi M, Sanyal A. Ragbench: Explainable benchmark for retrieval-augmented generation systems[J]. arXiv preprint arXiv:2407.11005, 2024.

[20] Soudani H, Kanoulas E, Hasibi F. Fine tuning vs. retrieval augmented generation for less popular knowledge[C]//Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region. 2024: 12-22.