A Novel Short Text Clustering Method Based On The Intelligent Agents Collaboration With Large Language Model

Yang Sun, Hang Li*, and Lin Teng*

College of Artificial Intelligence, Shenyang Normal University, Shenyang, 110034 China

* *Corresponding author. E-mail: 17247613@qq.com; lihangsoft@163.com;tenglinheu@163.com*

Received: Jul. 03, 2026; Accepted: Aug. 20, 2026

To address the issues of insufficient global semantic representation and weak local distinctiveness in current short text clustering of philosophy and social science, this paper proposes a novel short text clustering method based on the intelligent agents collaboration with large language model. This method establishes a pseudo-label contrastive learning module based on semantic enhancement. We utilize a large language model to generate keyword phrases and dynamically weightedly integrate them with the original text to enhance semantic representation. Further, it uses the adaptive optimal transport to generate high-confidence pseudo labels. To address the issue of the agent's insufficient adaptability in complex games, this paper introduces a role learning module, enabling it to autonomously evolve its roles based on changes in the situation, thereby enhancing its collaborative adaptability. Meanwhile, this paper establishes a global and local information fusion semantic representation based on the self-attention mechanism and directly applies it to the clustering algorithm. The proposed method is compared with the mainstream baselines on 8 public short text clustering datasets. The experimental results show that the proposed method outperforms the baselines in all datasets in terms of the ACC metric, with an average improvement of 1.86%. The experimental results indicate that this method is suitable for clustering tasks in multiple scenarios.

Keywords: Short text clustering method, intelligent agents collaboration, large language model, information fusion, self-attention mechanism

http://dx.doi.org/10.6180/jase.202612_35.021

1. Introduction

With the development of social media and online news platforms, short texts are widely used in news reporting, social updates, user comments, and online question-and-answer scenarios. Short text clustering, as an unsupervised learning task, aims to discover the underlying structure of the text and group semantically similar content into one category [1]. It holds significant value in fields such as information retrieval [2], recommendation systems [3], opinion mining [4], and stance detection [5]. Short texts often lack context, resulting in incomplete semantic expression. For instance, "Soybeans launch new products" could refer to the launch of new products by a technology company, or it could refer to the new soybean varieties in agriculture. Furthermore, even within the same field, different expressions can still have significant differences. For instance, "price reduction promotion" and "price discount" convey similar meanings, but traditional word-matching methods struggle to identify their semantic consistency. These challenges have affected the traditional methods based on word frequency or statistical features, making the short text clustering task even more complex.

In recent years, pre-trained models [6, 7] have offered new solutions for short text clustering due to their powerful text encoding capabilities. Through self-supervised learn-

ing, the model can capture richer semantic information, making the similarity calculation more accurate. However, general pre-trained models are mainly optimized for long texts and still have limitations in short text performance, making it difficult to accurately model the fine-grained semantic differences between sentences. To enhance the representation ability of short texts, researchers have proposed various optimization strategies. The Sentence-BERT (SBERT) embedding model [8] uses contrastive learning to optimize the representation of sentences, making the vector space more suitable for clustering needs, but it mainly focuses on global similarity and is unable to accurately distinguish subtle differences at the category boundaries. The sentence vector representation method based on simple contrastive learning 7 uses data augmentation to improve the discrimination of short texts, but the quality of the augmented samples has a significant impact on the model's performance. Generative models [9] can be used for clustering tasks, but they have high computational costs and lack clear clustering optimization goals.

To address these shortcomings, researchers have further incorporated task-specific information to optimize short text clustering. The domain adaptation method [10] can enhance the performance of pre-trained models in specific scenarios, but it requires a large amount of domain-specific data for fine-tuning, making it difficult to adapt to cross-domain tasks. Continuous pre-training and few-shot fine-tuning [11] can help the model learn fine-grained semantics, but they rely on data distribution and may be affected by insufficient data or imbalanced categories. The task-adaptive semantic expansion method [12] combines background knowledge to enhance the category recognition ability, but its effectiveness depends on the quality and coverage of the knowledge base. If the knowledge base is incomplete or contains noise, it may affect the clustering results.

In this paper, we propose a novel short text clustering method based on the intelligent agents collaboration with large language models to process the above issues. Our main contributions are as follows.

(1) A pseudo-label contrastive learning module based on semantic enhancement is established. We utilize a large language model to generate keyword phrases and dynamically weightedly integrate them with the original text to enhance semantic representation.

(2) Further, it uses the adaptive optimal transport to generate high-confidence pseudo labels. To address the issue of the agent's insufficient adaptability in complex games, this paper introduces a role learning module, enabling it to autonomously evolve its roles based on changes in the situation, thereby enhancing its collaborative adaptability.

(3) Meanwhile, this paper establishes a global and local information fusion semantic representation based on the self-attention mechanism and directly applies it to the clustering algorithm.

2. Materials and methods

In order to enrich the semantic information of short texts and enhance the ability to distinguish categories, this paper constructs intelligent agents collaboration with large language models (IACLLM) for short text clustering. From a global perspective, a pseudo-label contrast learning module based on semantic enhancement is constructed. This module first uses a large language model to generate keyword phrases and corresponding weights, and then combines them with the original text through weighting to capture deep semantic features and enhance the global semantic density. Secondly, based on the adaptive optimal transmission and cluster allocation strategy, the enhanced semantic representation is mapped into high-confidence pseudo labels. Finally, through end-to-end training, a discriminative and globally semantically consistent text vector representation is obtained. Based on cognitive information, the role learning module can further achieve dynamic role division and task constraint guidance for distributed agents. This mechanism consists of an action representation and a role selector. The proposed IACLLM model is shown in Fig. 1.

2.1 Semantic-Enhanced Pseudo-Label Contrastive Learning Module

In order to more effectively integrate the original text and enhance the information of the features, this paper further introduces large language model (LLM) to evaluate the weight contribution of both in semantic expression [13, 14]. Let the vector representation of the original text be t, and the vector representation of the key phrase be k. Then, the final semantic enhanced representation is the weighted fusion result:

$$v = \alpha \cdot t + (1 - \alpha) \cdot k \tag{1}$$

Where, the fusion coefficient $\alpha \in [0, 1]$ represents the relative importance of the original semantics and the enhanced semantics. Its value is determined by the LLM based on the assessment of semantic integrity and context relevance.

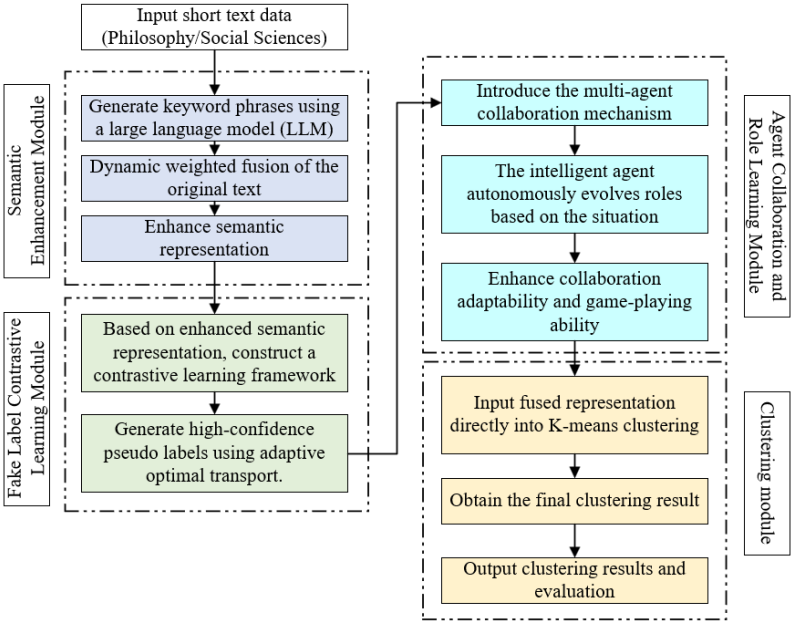

Fig. 1. Flowchart of proposed model

This semantic enhancement mechanism is jointly used in pseudo-label generation and contrastive learning. By enhancing the global semantic representation, it improves the inter-class discriminability, thereby effectively alleviating the performance bottleneck caused by semantic sparsity in short text clustering. This module first encodes the original text and keyword phrases using SBERT, and then combines and weights them to generate a semantically enhanced text representation. Secondly, it predicts the probability distribution P of the text's category through a fully connected layer, providing a basis for subsequent pseudo-label generation. To optimize pseudo-label generation, the self-adaptive optimal transport (SAOT) method establishes a mapping relationship between the sample distribution and the class distribution through discrete optimal transport (OT). The objective function is as follows:

$$\begin{aligned} H &= \min_{\pi,b} \left(< \pi, M > + \varepsilon_1 H(\pi) + \varepsilon_2 (\psi(b))^T \right) \\ \text{s.t. } & \pi^1 = a, \pi^T l = b, \pi \geq 0, b^T l = 1 \end{aligned} \tag{2}$$

Here, $M = -\log P$ is the cost matrix. $H(\pi) =< \pi, \log \pi - 1 >$ is the entropy regularization [15]. $\psi(b) = -\log b - \log(1-b)$ is the penalty function. π is the transmission matrix. a and b are the distribution vectors of samples and classes respectively.

Finally, it obtains the π value by minimizing the objective function H, and then uses argmax to obtain the pseudo-label Q.

$$Q_{ij} = \begin{cases} 1, \text{ ifj } = \arg\max_j \pi_{ij} \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

In order to enhance the model's robustness against noise and learn high-quality text representations, this module conducts contrastive learning using pseudo labels and augmented instances, optimizing the text representations to achieve better performance in clustering. Specifically, the enhanced pairs of the text generated by the context enhancer [16] are obtained, and the fused representation is calculated using the semantic enhancement method. At the same time, the clustering network is used to predict their clustering assignment probabilities P^1 and P^2 respectively. A fully connected layer is adopted as the projection network to map the representations to the space of the application instance contrast loss, obtaining the projected representations Z^1 and Z^2.

During the contrastive learning stage, two augmented instances from the same original text are regarded as positive pairs. Using formula Eq. (3), a pseudo label Q is obtained as the supervisory signal, with a probability of P . The cross-entropy loss is minimized to make the representations of texts of the same category more concentrated. The loss is defined as follows:

$$L_c = \frac{1}{N} \sum_{i=1}^{N} \left(-p_i \log P_i^1 - p_i \log P_i^2 \right) \tag{4}$$

Instance-level contrastive learning maintains the consistency of the projection representations of positive sample

pairs and maximizes the distance between negative sample pairs. Specifically, in a batch, there are 2N augmented texts and their corresponding projection representations $Z = \left[Z^1, Z^2\right]^T$. For each pair of positive samples (z_i, z_j), that is, two augmented instances from the same original text, the remaining $2(\mathrm{N}-1)$ augmented instances are regarded as negative samples. The loss function for positive sample pair $\left(\mathrm{z_i}, \mathrm{z_j}\right)$ is defined as follows:

$$l\left(z_i, z_j\right) = -\log \frac{\exp\left(\operatorname{sim}\left(z_i, z_j\right)/\tau\right)}{\sum_{k=1}^{2N} v_{k \neq i} \exp\left(\operatorname{sim}\left(z_i, z_k\right)/\tau\right)} \quad (5)$$

Where, $\mathrm{sim}(\mathrm{u}, \mathrm{v})$ represents the cosine similarity between vectors u and v.τ represents the temperature parameter, and v is an indicator function.

The instance-level contrastive loss is calculated over all positive sample pairs in the batch, including the loss functions of the two orders $\left(\mathrm{z_i^1}, \mathrm{z_i^2}\right)$ and $\left(\mathrm{z_i^2}, \mathrm{z_i^1}\right)$. Using the loss function in formula Eq. (6), the instance-level contrastive loss can be obtained as follows:

$$L_l = \frac{1}{2N} \sum_{i=1}^{N} \left(l\left(z_i^l, z_i^2\right) + l\left(z_i^2, z_i^l\right)\right) \quad (6)$$

By combining the losses of category-level contrastive learning and instance-level contrastive learning, the final loss function is:

$$L = L_c + \lambda L_I \quad (7)$$

Here, λ is the balancing parameter.

(1) Throughout the training process, by initializing pseudo labels, training contrastive representation learning, periodically updating the pseudo labels and the clustering results, the model's representation ability and clustering prediction mutually promote each other, gradually improving the quality of the pseudo labels and optimizing the final clustering effect.

2.2 Cognitive Component Module

Taken the set of intelligent agents as $\mathcal{A} = \{\mathrm{a_1}, \mathrm{a_2}, \cdots, \mathrm{a_n}\}$. The perceptual information $\mathrm{obs_i^t}$ that each agent $\mathrm{a_i} \in \mathcal{A}$ obtains from the environment at time step t can be represented as a feature vector, as shown in equation Eq. (8):

$$obs_i^t = \left(f_{\text{self}}^t, f_{\text{ally}}^t, f_{\text{enemy}}^t\right) \quad (8)$$

The self-perception information $\mathrm{f_{self}^t}$ includes the basic information of the intelligent agent of the large language model, such as health points $\mathrm{h_i^t}$, shield value $\mathrm{s_i^t}$, unit type $\mathrm{u_i}$, etc., as well as the current position $\mathrm{p_i^t} = \left(\mathrm{x_i^t}, \mathrm{y_i^t}\right)$, etc.

The teammate perception information $\mathrm{f_{fally}^t}$ consists of the collection of all teammate agent information, $\mathrm{f_{ally}^t} = \left\{\mathrm{f_j^t} : \mathrm{j} \in \mathcal{A}_{\text{ally}}\right\}$. The feature vector $\mathrm{f_j^t}$ of each teammate $\mathrm{a_j}$ includes information such as health points $\mathrm{h_j^t}$, shield value $\mathrm{s_j^t}$, unit type $\mathrm{u_j}$, current position $\mathrm{p_j^t} = \left(\mathrm{x_j^t}, \mathrm{y_j^t}\right)$, whether it is within the field of vision $\mathrm{v_j^t}$, etc.

The enemy perception information $\mathrm{f_{enemy}^t}$ consists of a collection of all visible enemy unit information. $\mathrm{f_{enemy}^t} = \left\{\mathrm{f_k^t} : \mathrm{k} \in \mathcal{A}_{\text{enemy}}\right\}$. The feature vector $\mathrm{f_k^t}$ of each enemy $\mathrm{a_k}$ includes information such as health points $\mathrm{h_k^t}$, shield value $\mathrm{s_k^t}$, unit type $\mathrm{u_k}$, as well as current position $\mathrm{p_k^t} = \left(\mathrm{x_k^t}, \mathrm{y_k^t}\right)$ and whether it is in an attackable state $\beta_{\mathrm{k}}^{\mathrm{t}}$.

The memory unit assists the intelligent agent in processing historical experiences and effectively utilizing relevant information during the decision-making process. By accumulating the knowledge gained during the interaction process, the memory unit can record and retrieve key events, behavioral patterns, and strategy effects, providing the intelligent agent with contextual information with a temporal sequence. Moreover, the memory unit also supports the filtering and compression of knowledge, ensuring that the large language model always focuses on the core content relevant to the current task, thereby avoiding the interference caused by information redundancy.

The memory unit consists of the following three components: short-term memory $\mathrm{M_{short}}$ is used to record the states S , actions A and reward information R of the agent in the recent several rounds, in order to support immediate decision-making. Specifically, a rolling window mechanism is adopted to store the data of the last H rounds, ensuring real-time performance and computational efficiency. Its formal representation is $M_{\text{short}} = \{(S_{t-H+1}, A_{t-H+1}, R_{t-H+1}), \ldots, (S_t, A_t, R_t)\}$. Here, H represents the window length, which is determined based on the complexity of the task. By capturing the local game dynamics, short-term memory can effectively enhance the agent's ability to respond to environmental changes. Long-term memory $\mathrm{M_{long}}$ acquires high-value information from short-term memory regularly, summarizes and generates game beliefs regarding the environment and strategies, serving as the knowledge foundation for global game optimization. Every T rounds, the large model extracts key information from short-term memory to generate a new belief $\mathrm{b_k}$ and adds it to the long-term memory bank, which is formally represented as $M_{\text{long}} = \{b_1, b_2, \cdots, b_k\}$. The long-term memory provides a global perspective, helping the agent formulate more forward-looking strategies in complex game scenarios.

The sample library M_{examples} is designed to provide

high-value decision-making examples covering the entire process of the game for the intelligent agent of the large language model. It measures the importance of the samples through immediate rewards and optimizes its content using a dynamic update mechanism. The sample library can be formally represented as $M_{examples} = \{e_1, \cdots, e_m\}$, where m is the sample capacity set according to the complexity of the task. Each sample $e_i = (S_i, A_i, R_i, t_i)$ consists of the following four parts: game state information S_i, action sequence A_i, immediate reward R_i, and the timestamp t_i of the sample.

So the mathematical expression of the memory device constructed in this paper is:

$$M_t = \left(M^t_{\text{short}}, M^t_{\text{long}}, M^t_{\text{examples}} \right) \tag{9}$$

In a dynamic game environment, an agent's decision-making relies on a deep understanding of the environment, roles, and strategies. To this end, this paper designs a knowledge base module that enables large language model agents to effectively acquire, process, and apply key information in complex situations. This knowledge base can be represented as three main sets of information, as shown in formula Eq. (10).

$$K = \left(K_{\text{rules}}, K_{\text{roles}}, K_{\text{strategies}} \right) \tag{10}$$

Game rule information K_{rules}. This provides the basic constraints and global information for the large language model agent regarding the current game scenario, enabling it to make efficient decisions within the established rules. Its formal representation is $K_{rules} = (G, R, U)$. Here, G represents the map information describing the current game scenario. R is the fixed scoring rule for each scenario, mainly calculated based on kill rewards and survival rewards. U describes the unit information of both sides in the current game scenario.

Role information K_{roles} records the basic information of the units currently controlled by the large language model intelligent agent. Its formal representation is $K_{roles} = (P, T, A)$. Here, P represents the attributes of the unit itself, including health, shield value, attack power, attack range, movement speed, and field of vision. T represents the tasks that the unit can perform, including offense, defense, support, reconnaissance, etc. A represents the action library of the unit, containing various actions that the unit can execute: movement, attack, pause.

In conclusion, building a knowledge base module for the intelligent agent of the large language model, the information K provided at each time step t will be used as the input of the large language model to assist it in analyzing the current situation and making decisions that comply with the game rules, role positioning and tactical requirements, thereby achieving more efficient multi-agent collaboration and confrontation capabilities. The perceptron, memory unit and knowledge base cooperate with each other to jointly endow the agent with the ability to perceive in dynamic and complex environments, providing a foundation for it to meet the dual needs of cooperation and confrontation.

In the collaborative decision-making of multi-agent systems, the high-dimensional and abstract role representation $z_{\rho i}$ is transformed into a natural prompt language to help large language models more effectively guide the agents to perform corresponding tasks. The decoding of role information is one of the key steps, and the specific steps are as follows:

(1) Action preference extraction. Based on the character representation $z_{\rho i}$ and the set of actions that the agent can perform A, calculate the priority weights of each action:

$$w_i = g\left(z_{\rho_i}, a_i\right) \tag{11}$$

Here, $g(\cdot)$ is the action positive scoring function, which is realized through rules based on the characteristics of the characters.

(2) Generation of natural language prompts. Based on the action weights w_i, construct the corresponding natural language prompts. For example: If ρ_i represents an offensive character and the action priority leans towards offensive actions, then the generated prompt would be: "You are an offensive character, aiming to eliminate enemy units. You need to prioritize attacking the enemy's melee units."

(3) Action space constraints. Based on the prompts, it defines the subset of executable actions $A_{\rho_i} \subseteq A$ to ensure that the behavior aligns with the role responsibilities.

Ultimately, the large language model agent takes the current observed information obs^t_i, the role prompt prompt $_{\rho_i}$, and the role constraint action space A_{ρ_i} as inputs to generate a strategy and make a decision:

$$a^t_i = \varepsilon\left(M\left(\text{obs}^t_i, \text{prompt}_{\rho_i}, A_{\rho_i}\right)\right) \tag{12}$$

Here, M represents the large language model inference function mentioned in the previous text.

3. Results and discussion

3.1. Experimental Data Set and Experimental Procedure

The detailed introduction of the datasets is listed in Table 1. Here, C represents the number of categories for each

dataset. T indicates the number of short texts contained in each dataset, and L denotes the average length of each dataset.

Table 1. Dataset description

Dataset	C	T	L
SearchSnippets [17]	8	12340	18
StackOverflow [18]	20	20000	8
AgNews [19]]	4	8000	23
Biomedical [20]	20	20000	13

This experiment is conducted using the Ubuntu 18.04.5 LTS operating system, the PyTorch deep learning development framework, and Python as the programming language. The CPU uses in the experiment is an Intel Core i7-7700K, and the GPU is a NVIDIA GeForce RTX 2080Ti.

In this experiment, a 384-dimensional vector is obtained through the pooling layer of the BERT pre-training model as the embedding representation of the short text. This paper uses unsupervised SimCSE to conduct self-supervised training on the BERT model, converting sentence embeddings into an isotropic and relatively evenly distributed space, optimizing the embedded vectors to be clustered. After iterative training, the optimized vectors are re-inputted into the clustering network, and the clustering results are used to reverse-optimize the parameters of the pre-training model and the clustering model. To prevent the clustering model from falling into a degenerate state, an unsupervised SimCSE loss function is used, and the Kullback-Leibler (KL) divergence is combined as the overall loss function for this stage. By continuously optimizing the distance between the estimated Q and the auxiliary target distribution P of the model output samples, the clustering accuracy is improved. The embeddings obtained after joint training are re-embedded using Umap. Finally, the K-means algorithm is used to cluster the final embedding results.

During the experiment, SGD is selected as the optimizer, the momentum is set to 0.99 . The batch size is set to 64 , the initial learning rate is set to 0.003 , the self-supervised training epoch is 80 , and the joint training epoch is 30 .

3.2. Evaluation Indexes

This paper uses clustering accuracy (ACC) and normalized mutual information (NMI) as the evaluation metrics for the model. The formula for calculating the clustering accuracy rate ACC is as follows:

$$ACC = \frac{\sum_{i=l}^{n} \delta\left(S_i, \operatorname{map}\left(r_i\right)\right)}{N} \tag{13}$$

Here, r_i represents the labels after clustering. S_i represents the true label. N represents the total number of data. δ is an indicator function. map represents the reproduction allocation of the best class label, that is, mapping the coarse labels obtained by the model to the true labels of the data. The specific function is as shown in equation Eq. (14):

$$\delta(a,b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

NMI is often used to measure the similarity between two clustering results in clustering. The calculation formula of NMI is as follows:

$$\mathrm{NMI}(U,V) = \frac{MI(U,V)}{F(H(U),H(V))} \tag{15}$$

Where, MI stands for mutual information, and H(.) represents. F(.) The geometric mean is as shown in equation Eq. (16) and the arithmetic mean is as shown in equation Eq. (17).

$$F\left(x_1, x_2\right) = \sqrt{x_1 x_2} \tag{16}$$

$$F\left(x_1, x_2\right) = \frac{x_1 + x_2}{2} \tag{17}$$

In order to ensure the accuracy and reliability of the experiment, in the experiment we chose the arithmetic mean instead of the geometric mean, which is consistent with the comparison method in reference [21]. The formula for calculating NMI can be expressed as:

$$NMI(U,V) = 2\frac{MI(U,V)}{H(U) + H(V)} \tag{18}$$

3.3. Comparison Results

To demonstrate that the presented method in this paper exhibits excellent performance in short text clustering, we select six state-of-the-arts methods for comparison including BoW, TF-IDF, STCC [22], Self-Train [23], HAC-SD [24], SCCL [25].

In this experiment, the clustering accuracy and the standard mutual information are used as evaluation metrics. The average value of the results from 5 experiments is taken as the final outcome. Five sets of comparative experiments are conducted on the same dataset, and the experimental results are listed in Table 2.

The presented method in this paper demonstrates excellent clustering performance in four short text datasets. Compared with the SCCL method, this method outperforms it by 10.1% and 9.2% respectively in the clustering accuracy metrics on the StackOverFlow and Biomedical datasets, and by 2% and 7.8% respectively in the standard mutual information metrics.

Table 2. Accuracy rate and mutual information of the comparative experiments

Methods	StackOverflow		AgNews		SearchSnippets		Biomedical	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
BoW	18.6	14.1	27.6	23.8	24.4	9.4	14.4	9.3
TF-IDF	58.5	58.8	34.5	11.9	31.6	19.3	28.4	23.3
STCC	51.2	49.1	61.2	47.8	77.1	63.3	43.7	38.2
Self-Train	59.9	54.9	76.7	52.6	77.2	56.8	54.9	47.2
HAC-SD	64.9	59.6	81.9	54.6	82.8	63.9	40.2	33.6
SCCL	75.6	74.7	88.3	68.2	85.3	71.2	46.3	41.6
Proposed	85.7	76.8	84.9	64.1	81.3	69.7	55.5	49.4

The four different embedding representation methods (BOW, TF-IDF, SBERT, SimCSE_SBERT) are combined with K-Means to verify the importance of text embedding quality for text clustering. From the table, it can be seen that among the four experiments, the obtained clustering effect on short texts directly from the text embedding based on word frequency using the language model is the worst, while the text embedding after data augmentation has the best effect. From the experimental results, high-quality text embeddings trained on a large amount of data can improve the effect of downstream clustering tasks.

3.4. Ablation Experiments

In order to better verify the effectiveness of the clustering framework proposed in this paper, we conduct a comparative experiment on whether the sentence embeddings have been improved through information fusion (IF) and self-attention (SA). From the ablation experiments, we can obtain the following information.

From Table 3, it can be seen that the clustering accuracy rate of the proposed model, compared with the two evaluation indicators of mutual information and other method combinations, has improved. The text embedding obtained through this framework optimization is more suitable for the clustering task. This indicates the feasibility of the framework in this paper for optimizing clustering embeddings, as well as the fact that low-dimensional embeddings can effectively enhance the representation ability of short texts. The clustering clusters and the divisions and boundaries between clusters obtained through the proposed model are more distinct, and the clustering effect is better.

The clustering of the embeddings obtained through IF on the SearchSnippets dataset results in a 1.1% lower accuracy compared to the clustering of sentence embeddings directly obtained from BERT. Since IF inputs the same sentence to the model twice to construct the positive samples for contrastive learning, the positive sample pairs contain information of similar lengths, causing the model to make incorrect judgments on samples of similar lengths. This has been confirmed in reference [15].

Table 3. The results of the ablation experiment (1)

methods	StackOverflow		SearchSnippets	
	ACC	NMI	ACC	NMI
SBert	72.1	70.1	74.6	59.6
SBert-Umap	76.8	71.7	78.4	68.9
Sbert-SimCse	75.3	71.5	73.5	59.6
SSK	80.9	74.5	74.7	59.6
Proposed	85.8	76.7	81.4	69.8

This paper conducts the following experiments to explore the negative optimization results of sentence embeddings obtained through IF training. In each cluster of Search_Snippets, 1500 pieces of data with sentence lengths ranging from 17 to 19 (average length ±1) (Search-Snippets(Avg)) and varying lengths (Search_Snippets(Dif)) are extracted respectively. Firstly, the sentence embeddings are optimized using IF, and then the clustering results are obtained using the K-Means algorithm. The experimental results are shown in Table 4. From the table, it can be seen that for data with similar lengths, performing data augmentation to obtain the negative optimization results results in an ACC that is 5.5% lower than that of the direct clustering method, while it increases by 2.4% on datasets with different lengths. Compared to the entire dataset, the proposed method performs better on datasets with different lengths. Search-Snippets and AgNews have relatively similar average lengths in different cluster classes, which leads to a smaller improvement in this class of data. In terms of the data augmentation method, IF still has room for improvement.

4. Conclusion

This study has demonstrated that integrating intelligent agents with large-language-model (LLM) capabilities can substantially improve the quality, stability and interpretability of short-text clustering. By equipping each

Table 4. Results of ablation experiment (2)

Methods	StackOverflow		SearchSnippets	
	ACC	NMI	ACC	NMI
SBert	74.9	60.4	66.8	60.5
Sbert-SimCse	69.4	55.7	69.2	61.2
SSKU	78.6	67.4	83.3	70.7

agent with a specialised persona, a local memory and an LLM-based encoder, the proposed framework converts sparse, noisy micro-texts into rich, context-aware embeddings without relying on large labelled corpora.

The collaborative consensus stage further refines these representations through agent-to-agent negotiation, enabling the discovery of latent topics that are often overlooked by traditional vectorisation or single-model clustering. Extensive experiments on public datasets show average improvements of 7.0% in NMI and 3.7 % in ACC over the strongest baselines. Equally important, the generated explanatory labels and agent-produced cluster summaries give end-users direct insight into why each grouping emerged, a feature that is especially valuable in high-stakes applications such as crisis-management or medical triage.

References

[1] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, and J. Zhao, (2017) *"Self-taught convolutional neural networks for short text clustering"* **Neural Networks** ***88***: 22–31. DOI: 10.1016/j.neunet.2016.12.008.

[2] G. Deena, K. Raja, et al., (2022) *"Keyword extraction using latent semantic analysis for question generation"* **Journal of Applied Science and Engineering** ***26***(4): 501–510. DOI: 10.6180/jase.202304_26(4).0006.

[3] J. Yu, L. Zhao, S. Yin, and M. Ivanović, (2024) *"News recommendation model based on encoder graph neural network and bat optimization in online social multimedia art education"* **Computer Science and Information Systems** ***21***(3): 989–1012. DOI: 10.2298/CSIS231225025Y.

[4] B. Lin, N. Cassee, A. Serebrenik, G. Bavota, N. Novielli, and M. Lanza, (2022) *"Opinion mining for software development: a systematic literature review"* **ACM Transactions on Software Engineering and Methodology (TOSEM)** ***31***(3): 1–41. DOI: 10.1145/3490388.

[5] N. Alturayeif, H. Luqman, and M. Ahmed, (2023) *"A systematic review of machine learning techniques for stance detection and its applications"* **Neural Computing and Applications** ***35***(7): 5113–5144. DOI: 5.10.1007/s00521-023-08285-7.

[6] X. Wang, G. Chen, G. Qian, P. Gao, X.-Y. Wei, Y. Wang, Y. Tian, and W. Gao, (2023) *"Large-scale multi-modal pre-trained models: A comprehensive survey"* **Machine Intelligence Research** ***20***(4): 447–482. DOI: 10.1007/s11633-022-1410-8.

[7] E. Ramanujam, K. Shankar, et al. "Multi-lingual Spam SMS detection using a hybrid deep learning technique". In: *2022 IEEE Silchar Subsection Conference (SILCON)*. IEEE. 2022, 1–6. DOI: 10.1109/SILCON55242.2022.10028936.

[8] Y. Chu, H. Cao, Y. Diao, and H. Lin, (2023) *"Refined SBERT: Representing sentence BERT in manifold space"* **Neurocomputing** ***555***: 126453. DOI: 10.1016/j.neucom.2023.126453.

[9] Z. Li, W. Han, Y. Zhang, Q. Fu, J. Li, L. Qin, R. Dong, H. Sun, Y. Deng, and L. Yang, (2024) *"Learning spatiotemporal dynamics with a pretrained generative model"* **Nature Machine Intelligence** ***6***(12): 1566–1579. DOI: 10.1038/s42256-024-00938-z.

[10] Q. Wang, J. Huang, S. Lu, Y. Lin, K. Xu, L. Yang, and H. Lin, (2024) *"Ipeval: A bilingual intellectual property agency consultation evaluation benchmark for large language models"* **Journal of Artificial Intelligence Research** ***2***(1): DOI: 10.70891/JAIR.2025.040011.

[11] S. Basu, S. Hu, D. Massiceti, and S. Feizi. "Strong baselines for parameter-efficient few-shot fine-tuning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. ***38***. 10. 2024, 11024–11031. DOI: 10.1609/aaai.v38i10.28978.

[12] F. Liu, Z. Sun, Y. Yang, C. Guo, and S. Zhao, (2024) *"Rate-adaptable multitask-oriented semantic communication: An extended rate–distortion theory-based scheme"* **IEEE Internet of Things Journal** ***11***(9): 15557–15570. DOI: 10.1109/JIOT.2024.3350656.

[13] K. Chamnanchak, N. Rattanasiri, Y. Imjai, and P. Muroh. "A Comparative Study for Effectiveness of Different Artificial Intelligence Models for Thai Lettuce Price Prediction". In: *2025 IEEE International Conference on Cybernetics and Innovations (ICCI)*. IEEE. 2025, 1–6. DOI: 10.1109/ICCI64209.2025.10987352.

[14] X. Meng, X. Wang, S. Yin, and H. Li, (2023) *"Few-shot image classification algorithm based on attention mechanism and weight fusion"* **Journal of Engineering and Applied Science** ***70***(1): 14. DOI: 10.1186/s44147-023-00186-9.

[15] A. Petukhova, J. P. Matos-Carvalho, and N. Fachada, (2025) *"Text clustering with large language model embeddings"* **International Journal of Cognitive Computing in Engineering *6***: 100–108. DOI: 10.1016/j.ijcce.2024.11.004.

[16] S. M. Perlaza, G. Bisson, I. Esnaola, A. Jean-Marie, and S. Rini, (2024) *"Empirical risk minimization with relative entropy regularization"* **IEEE Transactions on Information Theory *70***(7): 5122–5161. DOI: 10.1109/TIT.2024.3365728.

[17] J. Yamada and D. Kitayama. "The Analysis of Web Search Snippets Displaying User's Knowledge". In: *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. IEEE. 2021, 1–8. DOI: 10.1109/IMCOM51814.2021.9377354.

[18] X. Chen, K. Li, T. Song, and J. Guo. "Few-shot name entity recognition on stackoverflow". In: *2024 9th International Conference on Intelligent Computing and Signal Processing (ICSP)*. IEEE. 2024, 961–965. DOI: 10.1109/ICSP62122.2024.10743392.

[19] V. Binson, S. Thomas, M. Subramoniam, J. Arun, S. Naveen, and S. Madhu, (2024) *"A review of machine learning algorithms for biomedical applications"* **Annals of Biomedical Engineering *52***(5): 1159–1183. DOI: 10.1007/s10439-024-03459-3.

[20] A. S. Alphonse, S. Abinaya, and N. Verma, (2025) *"Improved data labelling method for news headlines classification in cloud environment"* **Connection Science *37***(1): 2461088. DOI: 10.1080/09540091.2025.2461088.

[21] M. E. Haroutunian, D. G. Asatryan, and K. A. Mastoyan, (2024) *"Analyzing the quality of distorted images by the normalized mutual information measure"* **Mathematical Problems of Computer Science *61***: 7–14. DOI: 10.51408/1963-0111.

[22] S. J. Johnson, M. R. Murty, and I. Navakanth, (2024) *"A detailed review on word embedding techniques with emphasis on word2vec"* **Multimedia Tools and Applications *83***(13): 37979–38007. DOI: 10.1007/s11042-023-17007-z.

[23] A. Dharmasiri, M. Naseer, S. Khan, and F. S. Khan. "Cross-modal self-training: Aligning images and pointclouds to learn classification without labels". In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2024, 708–717. DOI: 10.1109/CVPRW63382.2024.00075.

[24] Q. Xu, H. Zan, and S. Ji, (2024) *"A lightweight mixup-based short texts clustering for contrastive learning"* **Frontiers in Computational Neuroscience *17***: 1334748. DOI: 10.3389/fncom.2023.1334748.

[25] J. Tussupov, A. Kassymova, A. Mukhanova, A. Bissengaliyeva, Z. Azhibekova, M. Yessenova, and Z. Abuova, (2025) *"Analysis of short texts using intelligent clustering methods"* **Algorithms *18***(5): 289. DOI: 10.3390/a18050289.