A Two-Stage Progressive Intent Clustering for Task-Oriented Dialogue

Bingzhu Du and **Nan Su** and **Yuchi Zhang** and **Yongliang Wang**
Ant Group, Beijing, China

Abstract

Natural Language Understanding (NLU) is one of the most critical components of task-oriented dialogue, and it is often considered as an intent classification task. To achieve outstanding intent identification performance, system designers often need to hire a large number of domain experts to label the data, which is inefficient and costly. To address this problem, researchers' attention has gradually shifted to automatic intent clustering methods, which employ low-resource unsupervised approaches to solve classification problems. The classical framework for clustering is deep clustering, which uses deep neural networks (DNNs) to jointly optimize non-clustering loss and clustering loss. However, for new conversational domains or services, utterances required to assign intents are scarce and the performance of DNNs is often dependent on large amounts of data. In addition, although re-clustering with k-means algorithm after training the network usually leads to better results, k-means methods often suffer from poor stability. To address these problems, we propose an effective two-stage progressive approach to refine the clustering. Firstly, we pre-train the network with contrastive loss using all conversations data and then optimize the clustering loss and contrastive loss simultaneously. Secondly, we propose adaptive progressive k-means to alleviate the randomness of vanilla k-means, achieving better performance and smaller deviation. Our method ranks second in DSTC11 Track2 Task 1, a benchmark for intent clustering of task-oriented dialogue, demonstrating the superiority and effectiveness of our method.

1 Introduction

Task-oriented dialogue technique is of great popularity and is widely adopted in various applications, such as Microsoft's XiaoIce (Zhou et al., 2020), Google Assistant etc. Natural Language Understanding (NLU) is one of the most crucial components of task-oriented dialogue, and it is often considered as an intent classification task. Supervised deep networks are the most widely used methods for intent classification due to their high performance, such as classification head based on pre-trained language models. However, such kind of methods require a large amount of labeled data that is collected through painstaking analysis of conversation transcripts by domain experts. This is not feasible for many commercial operations, especially in emerging domains or services.

Therefore, researchers have proposed low-resource unsupervised intent clustering methods to address this problem, as described in Table 1. An intent clustering procedure assigns an intent label to each dialogue turn labeled with "informIntent" dialogue act based on the conversation between the customer and the agent. Such kind of unsupervised approach has tackled the above mentioned low resource dilemma for task-oriented dialogue systems. The most commonly used clustering method is k-means (Macqueen, 1967; Lloyd, 1982) due to its simplicity and efficiency. However, its clustering performance depends on data representation, which is often ineffective for high-dimensional data. To alleviate this problem, Principal Component Analysis (PCA) and Non-negative Matrix Factorization (NMF) were proposed. They are non-parametric linear transformations and followed by clustering algorithms sequentially. Now there is more work focusing on nonlinear dimensionality reduction and co-optimization of clustering objectives using DNNs, called deep clustering. Two surveys provide detailed descriptions of this approach. Min et al. (2018) gives a review from the perspective of network structure. Aljalbout et al. (2018) presents a systematic taxonomy of clustering methods using DNNs.

Xie et al. (2016) proposed Deep Embedded Clustering (DEC), a method for the first time combining DNN with clustering, which uses deep neural networks to iteratively optimize sentence represen-

Proceedings of The Eleventh Dialog System Technology Challenge, pages 48–56

Speaker Role	Utterances	Dialog Acts	Gold Intent
Agent	How can I help you today?		
Customer	**I want to pay my auto insurance bill.**	InformIntent	Intent1
Agent	I will need your account number.		
Customer	It's two three one five six four eight seven nine.		
Agent	Okay. Thank you.		
Customer	**So what is my payment?**	InformIntent	Intent2
Agent	Okay. I'm showing that no payment is due until April.		

Table 1: Intent Clustering. The utterance written in bold is target utterance required to assign intents and its proportion is relatively small from the conversations.

tation and cluster assignment. Hadifar et al. (2019) applied DEC approach with sentence embedded by Smooth Inverse Frequency (SIF) to perform short text clustering following a multi-stage approach. To facilitate better data separation, Supporting Clustering with Contrastive Learning (SCCL) (Zhang et al., 2021) is the first method to introduce contrastive learning into clustering by jointly optimizing contrastive loss and clustering loss.

However, DNNs usually rely on large amounts of data to update model parameters or proper initialization to speed up convergence. In addition, the acquisition of intent utterances is limited and difficult, especially in emerging dialogue domains or services, so we improve the clustering performance from the perspective of model initialization. Meanwhile, in order to get better cluster assignments, re-run clustering using k-means is generally performed after training the network. It is known that K-means suffer from poor stability due to the way of obtaining initial centroids. To solve these problems, we propose an effective two-stage progressive intent clustering from conversations. Our main contributions are as follows:

- Pre-training with contrastive learning. To obtain a proper parameter initialization, we pretrain the network before jointly learning utterance representations and clustering. Due to the lack of target utterances, contrastive learning is introduced and trained with all conversations. The introduction of pre-training increases the robustness of the model and also outputs better initial parameters for subsequent joint training.

- Progressive k-means. To alleviate the stability of vanilla k-means, we propose progressive k-means, an improved cluster algorithm that is adaptive to deep clustering. As the deep representation is learned, the distance loss is gradually minimized until convergence. The cluster centroids are no longer chosen randomly but are replaced by high-confidence cluster centers.

- Our approach ranks second in ACC metrics and first in F1 metrics on the DSTC11 Track2 task1, a benchmark for intent clustering from conversations for task-oriented dialogue, demonsrating the effectiveness and superiority on two different domain datasets.

2 Proposed Approach

To refine the performance of automatic intent clustering for task-oriented dialogues, we propose a two-stage progressive approach. Our specific method pipeline is shown in Figure 1.

- Estimating number of intent. According to the task description, the number of clusters is unknown, so we need to automatically estimate the number of clusters. We consider it as a parameter and explore the search space based on a predefined objective function to decide the optimal value. The utterance representation is extracted by all-mpnet-base-v2, which is the best model for learning sentence representations. The specific parameter estimation methods are described in 3.2.

- Stage one. Due to the shortage of target utterances, we used the entire conversations to pretrain the network with the help of contrastive learning (see 2.1). During training, we use progressive K-means (see 2.4) to assign intent labels to the target utterance. Besides, the progressive k-means outputs a high-confidence cluster centers named as High-Score-Center,

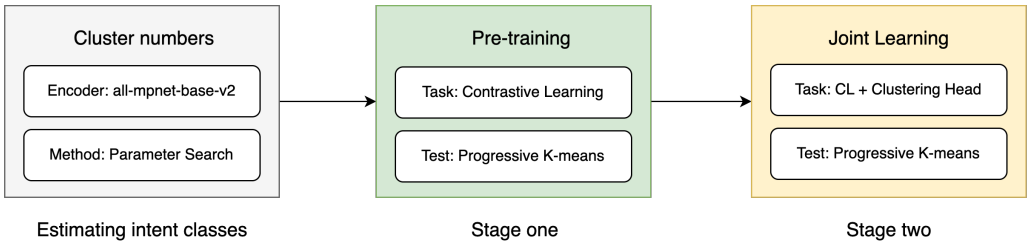

Figure 1: Task Pipeline

which is used as the initial center of clustering head in the second stage. Pre-training can provides better initial parameters for subsequent joint learning and thus converge to a better position.

- Stage two. During the second stage of the approach, we jointly optimize contrastive loss and clustering loss on target utterances, which can iteratively refine the cluster assignments alongside improving utterance representations (see 2.2; 2.3). The parameters of linear layer of clustering head are initialized by High-Score-Center which is obtained in the first stage. In order to get better cluster assignments, progressive k-means is performed on target utterances during the training process.

2.1 Contrastive Learning

Learning universal sentence embedding is a significant problem in natural language processing (Kiros et al., 2015; Hill et al., 2016; Conneau et al., 2017; Logeswaran and Lee, 2018). Much recent research has shown that contrastive learning can be extremely effective to advance the sentence embeddings (Henderson et al., 2020; Zhang et al., 2020). Contrastive learning aims to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Hadsell et al., 2006). It refines representation learning by using a contrastive prediction task. The training data consists of positive pairs and negative pairs constructed from unlabeled or labeled datasets.

We follow the contrastive learning framework proposed in SimCLR by Chen et al. (2020), using the normalized temperature-scaled cross-entropy loss as the training loss which is also called InfoNCE in the previous literature (Hjelm et al., 2019). The specific contrastive learning method we adopt is SimCSE (Gao et al., 2021), which uses dropout coupled with pre-trained language models as the minimal data augmentation strategy compared with previous explicit data augmentation strategies. This simple strategy proved to be superior to many traditional data augmentation techniques in natural language processing such as word deletion, reordering and substitution and back translation, etc. Given a minibatch of N examples $D = \{x_i\}_{i=1}^{N}$. Let h_i and h_i^+ denote the representation of example x_i, which is the output obtained by feeding x_i into the encoder twice by applying different dropout masks z_i and z_i^+, constituting a positive pair. The negative pair is derived from two different examples within the minibatch D. The contrastive learning objective x_i is as follows,

$$l_i^{CL} = -log(\frac{e^{sim(h_i,h_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(h_i,h_j^+)/\tau}}) \quad (1)$$

$$h_i = f_\theta(x_i, z_i), h_i^+ = f_\theta(x_i, z_i^+) \quad (2)$$

$f_\theta(.)$ denotes the encoder, τ denotes the temperature parameter which we set as 0.5, $sim(h_1, h_2)$ is the cosine similarity $\frac{h_1^T h_2}{||h_1||.||h_2||}$.

The contrastive learning approach can also be replaced by EsimCSE (Wu et al., 2021), which refines the positive and negative construction method called Enhanced SimCSE. Since positive pairs are of the same length, EsimCSE breaks this length pattern by using word repetitions that do not destroy the meaning of the sentence. In addition, by introducing momentum contrast (He et al., 2020), more negative pairs can be leveraged to guide the model to achieve a better representation.

2.2 Clustering with KL divergence

DEC was proposed by Xie et al. (2016), which trains the auto-encoder by minimizing reconstruction loss and then fine-tunes the encoder network by optimizing KL-divergence with an auxiliary target distribution. Let e_i denote the representation of instance x_i, $u_k, k \in 1, ..., K$ denote the cluster centroid. Following Laurens and Hinton (2008), we use the Student's t-distribution to measure the similarity between embedded point e_i and centroid

k,

$$q_{ik} = \frac{(1 + ||e_i - u_k||^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{k'=1}^{K}(1 + ||e_i - u'_k||^2/\alpha)^{-\frac{\alpha+1}{2}}} \quad (3)$$

q_{ik} can be regarded as the probability of assigning x_i to the k cluster and then be used as a soft assignment of embeddings to cluster centers, α denotes the degree of freedom of the Student's t-distribution which we set as 1.

We use a linear layer to approximate the centroids of each cluster and optimize it iteratively by leveraging an auxiliary distribution. let p_{ik} denote the auxiliary probability which has stricter probabilities compared to the similarity score q_{ik},

$$p_{ik} = \frac{q_{ik}^2/f_k}{\sum_{k'=1}^{K} q_{ik}^2/f'_k} \quad (4)$$

Here $f_k = \sum_{i=1}^{N} q_{ik}, k = 1, ..., K$ can be interpreted as the soft cluster frequencies. By quadratic sharpening of the soft assignments q_{ik}, this target distribution can assign more weight to high-confidence instances and less weights to low-confidence instances. In addition, by using soft cluster frequencies for normalization, the bias caused by large clusters can be avoided.

Finally, we use KL divergence, which can measure the similarity between two probability distributions P and Q, as the training objective to achieve pushing the soft assignments to the target distribution. The specific loss is as follows,

$$l_i^C = KL[p_i||q_i] = \sum_{k=1}^{K} p_{ik} log \frac{p_{ik}}{q_{ik}} \quad (5)$$

Unlike SCCL, in which the parameters of the linear layer that fits the clustering centroids are initialized by the cluster centers generated by standard K-means in the hidden space h. We instead use the clustering centers computed from the high-scoring instances generated by the progressive k-means (detailed as 2.4) we iterated during pre-training.

2.3 Joint Learning

We adopt joint learning where simultaneous optimization of representation learning and clustering is performed. In the second stage of our method, we adopt joint learning. Its specific structure is shown in Figure 2. First, the encoder maps the input data to the representation space, followed by

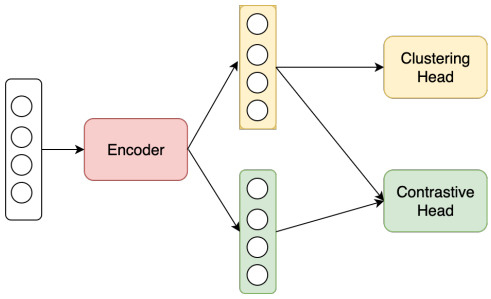

Figure 2: Joint Modeling Framework

the clustering head and contrastive head modules described in detail above. Inspired by SimCSE, we pass the input data into the encoder twice with different dropout masks. The overall training objective is the weighted sum of the contrast loss and clustering loss as follows.

$$L = L_{Instance-CL} + \eta L_{Cluster} \quad (6)$$

$$L_{Instance-CL} = \sum_{i=1}^{N} l_i^{CL}/N \quad (7)$$

$$L_{Cluster} = \sum_{i=1}^{N} l_i^{C}/N \quad (8)$$

$L_{Instance-CL}$ denotes the contrastive loss, the average loss over the minibatch. $L_{Cluster}$ denotes the average clustering loss on minibatch.

2.4 Progressive K-means

Deep clustering is a method that uses deep neural networks to jointly optimize representation learning and clustering. We propose progressive K-means, which is adaptive to this method, introducing an updated representation of instances for iteration in the outer layer of the original K-means iteration.

Except for the first starting centers, we use the improved cluster centroids generated by the last optimization. K-means is a well-known clustering algorithm that is widely used due to its simplicity and efficiency. However, the arbitrary nature of its initial centers leads to relatively large differences between cluster assignments, which ultimately does not guarantee the accuracy of the results. K-means++ (Arthur and Vassilvitskii, 2007) improves the method of selecting the starting center. The centroids are selected based on the weight of the data points, which are calculated based on their squared distance from the nearest center. Despite the significant improvements, the clustering

Dataset	Conversation		Cluster			
	Nums	Turns	Turns	Len	Labels	L/S
Insurance	948	66875	1205	17	22	10
Finance	2000	130274	1597	27	39	45
Banking	1000	59180	1503	23	29	63

Table 2: Dataset statistics. Nums: number of conversations; Len: average number of words in each cluster query; Labels: number of clusters; L/S: the ratio of the size of the largest cluster to that of the smallest cluster.

Method	ACC	Precision	Recall	F1	NMI	ARI	#cluster
Baseline	55.80	64.97	62.98	63.26	62.98	39.85	29
T23(best)	**69.79**	76.09	76.12	76.00	75.05	59.23	30.5
T07(ours)	69.59	72.01	81.64	**76.50**	73.48	60.13	27.5

Table 3: Overall performance. Average experimental results of different methods on two test datasets; ACC-ARI is metric introduced as 3.3; Cluster is the predicted number of clusters and the true cluster numbers is 34; The highest value for each metric is written in bold

performance is still inadequate. To solve this problem, we propose an improved K-means which is adaptive to successive iterative representations.

Algorithm 1 Progressive K-means

for $epoch = 1, Ts//Interval$ **do**
  Obtain the representation from the network
  if $epoch = 1$ **then**
    Kmeans++
    Save High-Score-Centers
  else
    Kmeans++ initialized with High-Score-Centers
    Update High-Score-Centers
  end if
end for

Algorithm 1 summarizes the progressive K-means, where Ts denotes the total training steps, $Interval$ denotes the frequency of cluster updates, K-means++ denotes the original complete iterative process, and High-Score-Centers are the average embedding of the first 20 high silhouette coefficient score instances in each cluster, which will be used as the initial centroids of subsequent iterations. This approach can be used in combination with deep clustering methods. Except for the first iteration, K-means++ uses the latest representation of the instance with better separation in each new iteration and uses High-Score-Centers as the initial centroid. Thus, it results in an asymptotic and stable cluster assignment that greatly mitigates the randomness and deviation of K-means.

3 Experiments and Results

3.1 Datasets

We evaluated the proposed method on one development dataset and two test datasets provided by the organizers of DSTC11 Track2. The performance on the test dataset validates the effectiveness of our method. The ablation studies performed on the development dataset demonstrate the properties of different parts of our method.

The three datasets mentioned above are from different domains, namely insurance, finance and banking. The dataset contains about 1K customer-supported spoken conversations with manual transcription and annotation. Both customers and agents are real people. Each conversation has an average of 70 turns. Detailed statistics are presented in Table 2.

3.2 Experimental Setup

We adopt all-mpnet-base-v2 (Song et al., 2020) in the Sentence Transformers Library (Reimers and Gurevych, 2019) as an encoder, which is a pre-trained language model that can learn state-of-the-art universal sentence representations. Same with Zhang et al. (2021), the clustering head is implemented by a linear layer with size $768 \times K$, where K denotes the number of clusters, and the contrastive head is implemented using MLP with an input hidden size of 768 and an output hidden size of 128. We use the Adam optimizer with batch size

Method	ACC	Precision	Recall	F1	NMI	ARI	#cluster
Baseline	51.85	69.25	53.98	60.67	65.71	33.61	46
(T23)Best	**68.13**	70.95	76.52	73.63	72.85	55.52	32
(T07)Ours	67.06	71.82	78.71	**75.11**	74.48	**55.77**	41

Table 4: Performance of Finance dataset; the gold cluster numbers is 39; The highest value for each metric is written in bold

Method	ACC	Precision	Recall	F1	NMI	ARI	#cluster
Baseline	59.75	60.68	71.99	65.85	60.26	46.10	12
(T05)Best	**75.25**	78.78	82.50	**80.60**	**78.45**	**70.69**	26
(T07)Ours	72.12	72.19	84.56	77.89	72.47	64.50	14

Table 5: Performance of Banking dataset; the gold cluster numbers is 29; The highest value for each metric is written in bold

of 400 and set the maximum sentence length to 64, 100, and 100 for insurance, finance, and banking, respectively. We use a constant learning rate of 1e-6 to optimize the encoder while setting a learning rate of 1e-4 to optimize the rest of our model.

Hyperparameter Optimization (Hyperopt) (Bergstra et al., 2013) is used to search for the optimal number of clusters between 5 and 50. In this paper, we use Tree of Parzen Estimators (TPE) to search for the best parameters in a specified search space based on a predefined objective function. Specifically, we perform k-means with different numbers of clusters and then output the number with the highest silhouette coefficient score. Considering that the choice of initial centroids has a considerable impact on the clustering performance when applying the k-means algorithm, we conducted multiple search experiments. First we performed 30 Hypteropt search experiments with different seeds to obtain 30 optimal clustering numbers, and then we chose the mode of data points greater than the 60th percentile as the final clustering number. Outliers are considered here, and not all data points are used.

In the ablation study, K-means++ is executed after training the network to obtain clustering results. Similar to Xie et al. (2016), we run K-means++ with 20 restarts and select the best centroids. The experimental results are the average of 7 trials, and the standard deviation is also shown as Hadifar et al. (2019).

We measure the performance of our method on test datasets using fusion results of 7 trials. We fuse the results by voting. If the number of votes is greater than half, the result is used, otherwise the result of the model with the largest sample size is used. In this way, the more authoritative the model, the more likely it is to be chosen.

3.3 Experiment results

We evaluate the clustering performance from two perspectives and six standard metrics. Clustering accuracy (ACC), Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) focus more on model performance itself (Huang et al., 2014; Wang et al., 2017) while Precision, Recall and F1 forcus more on performance of practical application (Hadifar et al., 2019). They are used together to provide a comprehensive measure of overall system performance. ACC is used as the primary metric for evaluating system submission rankings and is calculated using the following logic:

$$ACC = \frac{\sum_{i=1}^{N} \delta(map(c_i) = y_i)}{N} \quad (9)$$

$\delta(.)$ is an indicator function, c_i is the clustering label for x_i, $map(.)$ is a mapping between the predicted labels and ground-truth labels using the Hungarian algorithm (Kuhn, 1955) for 1:1 assignment. It is a mapping between predicted labels and ground truth labels using the Hungarian algorithm for 1:1 assignment. For Precision, Recall and F1, the predicted categories will be assigned to the most frequent target categories, allowing a many-to-one assignment to ensure maximum sample matching.

The organizers ranked the submitted systems using the average ACC metric calculated on the

Datase	ACC	Precision	Recall	F1	NMI	ARI
SCCL	56.57±2.38	65.37±1.68	61.67±1.97	63.46±1.66	64.33±1.33	41.71±1.76
+Pretraining	62.16±1.68	67.94±1.58	75.46±0.92	71.49±0.78	69.65±0.65	46.90±2.01
+Refined centers	65.15±1.78	71.22±1.82	74.70±0.19	72.90±0.87	70.86±0.59	51.12±1.26
Our Method	**66.72±0.59**	**72.96±0.41**	**74.20±0.51**	**73.57±0.42**	**71.30±0.23**	**51.89±0.70**

Table 6: Ablation results on Insurance dataset. Performance was assessed by the mean results and standard deviation of 7 trials

financial and banking datasets. We first compare our approach with the baseline and the best model, and the results are presented in Table 3. Obviously, our model outperforms the baseline model by a large margin.

- **Baseline**. It adopts a traditional sequential modeling approach where feature representations are first extracted before using K-means. Clustering is performed directly in the feature space generated by the pre-trained language model all-mpnet-base-v2 in the sentence transformation library. Hyperopt is used to estimate the number of clusters.

- **Submission System**. T23 is the model ranking first and our method is denoted as T07.

As shown in Table 3, our system ranks second with the ACC (0.2% lower than the first place) and first with the F1 (0.5% higher than T23), indicating that our model is more competitive in the actual dialogue system development. In terms of other metrics, it has good and bad. In addition, our predicted number of clusters is 27.5, which differs from the ground truth figure by 6.5 (ground truth is 34).

To further analyze the performance of our system, we report the performance on the Finance and Banking datasets, respectively, as shown in Table 4 and Table 5. On the Finance dataset, our system ranks second in terms of ACC (1.07% lower than first) and outperforms T23 on the other five metrics. Notably, our model ranks first in the F1 and ARI metrics, outperforming T23 by 1.48% and 0.25%, respectively. The target cluster count is 39, while our prediction is 41. On the banking dataset, our system is 3.13% lower than the best model on the acc metric, ranking sixth. The performance is not satisfactory compared to the financial dataset, which may be caused by the bias brought by the estimated clusters counts. The ground truth label is 29 and our predicted value is 14, resulting in a difference of 15 data points.

Furthermore, T23 has the best result in the financial dataset but ranks 8th in the banking dataset, while T05 has the best result in the banking dataset but ranks 6th in the financial dataset. It demonstrates that none of these models guarantee accuracy and that there is much room for improvement, particularly in terms of stability. In addition, it also reflects that this task is difficult and the performance of clustering can be affected by a variety of factors. The two approaches we propose: improving the clustering algorithm and improving the initialization, are stable and effective in various domains. In the future, we will concentrate on doing more work on the estimation of the number of clusters.

3.4 Ablation Study

In order to improve the effectiveness of the automatic intent clustering method in dialogues, we refined the SCCL method from different perspectives. To investigate the effectiveness of each part of our proposed method, we performed an ablation study on the insurance dataset, by adding each module to SCCL in turn. Details are shown in Table 6. **+Pretraining:** A pre-training module with contrastive loss is added to the SCCL approach, which is trained using the entire conversations. It shows an improvement of almost 6% in the Acc metric and significant improvements in other metrics. In addition, the bias of the model is reduced due to better learned representations. **+Refined centers:** The parameters in the clustering head are initialized with the High-Score-Centers generated by the progressive k-means in pre-training. The results show an improvement of nearly 3%, demonstrating the great impact of initialization and the effectiveness of our proposed progressive k-means. **Our Method:** In addition to the above modifications, we performed progressive k-means after training the network to replace the previous

k-means method. The performance is also slightly improved by 1.6% and the bias is reduced to within 1%.

4 Conclusion

In this paper, to refine the performance of automatic intent clustering for task-oriented dialogue, we propose a two-stage progressive approach. To alleviate the lack of target utterance, we introduce pre-training before joint learning to increase the robustness of the model. In addition, we propose progressive K-means, an enhanced K-means method, which is compatible with deep clustering. Also, we use the High-Score-Center obtained by performing progressive K-means in the first stage to initialize the parameters of the clustering head in the second stage. Ultimately, our model results in better performance and lower deviation. Our proposed system proved its effectiveness by ranking second in ACC metrics and first in F1 metrics in DSTC11 Track2 Task 1. In addition, we conducted an ablation study to verify the performance of each component.

References

E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers. 2018. Clustering with deep learning: Taxonomy and new methods.

D. Arthur and S. Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*.

James Bergstra, Daniel Yamins, and David D. Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 115–123. JMLR.org.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Amir Hadifar, Lucas Sterckx, Thomas Demeester, and Chris Develder. 2019. A self-training approach for short text clustering. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 194–199, Florence, Italy. Association for Computational Linguistics.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and accurate conversational representations from transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online. Association for Computational Linguistics.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California. Association for Computational Linguistics.

R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. 2014. Deep embedding network for clustering. In *2014 22nd International Conference on Pattern Recognition*, pages 1532–1537.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28:*

Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 3294–3302.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics*, 2(1):83–98.

Van Der Maaten Laurens and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2605):2579–2605.

S. P. Lloyd. 1982. Least square quantization in pcm. *IEEE Transactions on Information Theory*, 28.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

J. Macqueen. 1967. Some methods for classification and analysis of multivariate observations. *Proc. Symp. Math. Statist. and Probability, 5th*, 1.

Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. 2018. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

K. Song, X. Tan, T. Qin, J. Lu, and T. Y. Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Wang, Peng, Xu, Bo, Zheng, Suncong, Zhao, Jun, Jiaming, and Tian. 2017. Self-taught convolutional neural networks for short text clustering. *Neural Networks the Official Journal of the International Neural Network Society*.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2021. ESimCSE: Enhanced Sample Building Method for Contrastive Learning of Unsupervised Sentence Embedding. *arXiv e-prints*, page arXiv:2109.04380.

Junyuan Xie, Ross B. Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 478–487. JMLR.org.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430, Online. Association for Computational Linguistics.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. 2020. An unsupervised sentence embedding method by mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1601–1610, Online. Association for Computational Linguistics.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of XiaoIce, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.