Reminding the incremental language model via data-free self-distillation

Han Wang · Ruiliu Fu · Chengzhang Li · Xuejun Zhang · Jun Zhou · Xing Bai · Yonghong Yan · Qingwei Zhao

Accepted: 22 April 2022 / Published online: 8 August 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Incremental language learning, which involves retrieving pseudo-data from previous tasks, can alleviate catastrophic forgetting. However, previous methods require a large amount of pseudo-data to approach the performance of multitask learning, and the performance decreases dramatically when there is significantly less pseudo-data than new task data. This decrease occurs because the pseudo-data are learned inefficiently and deviate from the real data. To address these issues, we propose reminding the incremental language model via data-free self-distillation (DFSD), which includes 1) self-distillation based on the Earth mover’s distance (SD-EMD) and 2) hidden data augmentation (HDA). SD-EMD can increase the efficiency of the model by adaptively estimating the knowledge distribution in all GPT-2 layers and effectively transferring data from the teacher model to the student model via adaptive self-multilayer-to-multilayer mapping. HDA can reduce deviations by decomposing the generation process via data augmentation and bootstrapping. Our experiments on decaNLP and text classification tasks with low pseudo-data sampling ratios reveal that the DFSD model outperforms previous state-of-the-art incremental methods. The advantages of DFSD become more apparent when there is less pseudo-data and larger deviations.

Keywords Incremental language learning · Self-distillation · Hidden data augmentation · Data-free constraint · Pseudo data

1 Introduction

Human learning is a long-term and continuous behavior [1] that represents the ability of human beings to retrieve previous knowledge through comprehension while acquiring new knowledge. However, traditional machine learning paradigms suffer from catastrophic forgetting [1–3] and thus cannot retain previously learned knowledge when acquiring new knowledge. Therefore, incremental learning has been proposed to improve model performance for both new tasks and previously learned tasks when a model learns different tasks sequentially without retrieving data from previous tasks, regardless of the task order [1, 4, 5]. Many works on mitigating catastrophic forgetting are based on appropriately relaxing restrictions on retrieving previous data [6–15]. In this paper, we investigated how to address catastrophic forgetting in incremental learning with data-free constraints [4] in the field of natural language processing (NLP), which involves training new tasks without retrieving real data from previous tasks.

A robust incremental learning model should ignore the type of task and be able to learn different fields from one task (e.g., text classification in different fields) as well as different tasks (e.g., question-answering, dialogue systems, and semantic parsing). With decaNLP [16], incremental learning models can learn a variety of tasks by regarding different NLP tasks as question-answer (QA) tasks. Recently, LAMOL [11], a generative lifelong learning framework that uses a language model (LM) to learn various NLP tasks in QA style, was developed. Instead of using real data from previous tasks, LAMOL generates pseudo-data for previous tasks, which it then uses to joint train with a new task. L2KD [12] is an improved LAMOL model that distills the output of the model at both the word and sequence levels. DnR [13] is an improved LAMOL model that distills the attention layer and the hidden states of some GPT-2 layers [17].

However, these LAMOL-based methods require a large amount of pseudo-data to be generated to approach the
performance level of multitask learning (MTL) methods. The cost of generating and training a large amount of pseudo-data is high in terms of both time and memory. Is it possible for the model to approach the performance of MTL with less pseudo-data? Regrettably, in the aforementioned works, the answer to this question is no. The performance decreased dramatically when less pseudo-data were trained with the new task in the abovementioned works. There were two reasons for this result: (1) The task-specific knowledge distribution in all GPT-2 layers and the differences across tasks were rarely considered, resulting in inefficient training of pseudo-data and clear catastrophic forgetting with less pseudo-data. (2) The model was overbiased toward new tasks due to the imbalanced training of the new task data with the pseudo-data, resulting in the pseudo-data deviating from the corresponding tasks in the next generation. As more tasks were learned, the accumulated overbias caused the performance of the model to deteriorate. This deviation was defined as “chaos” in [11]. In this paper, we define this deviation as noise.

Inspired by [18–20], we propose a reminding incremental language model via data-free self-distillation (DFSD) that uses two methods to address the above issues: (1) self-distillation based on the Earth Mover’s Distance [20] (SD-EMD), which adaptively estimates the distribution of the task knowledge in all layers and effectively transfers data from the teacher model to the student model via adaptive self-multilayer-to-multilayer mapping; SD-EMD allows the model to train the pseudo-data more efficiently; and (2) hidden data augmentation (HDA), which involves decomposing pseudo-data generation into an uncertain weighted mixture of all the trained task data. Noise is caused by an overestimation of unspecified tasks and is corrected by bootstrapping. HDA can reduce deviations between pseudo-data and real data. The above methods are presented in Fig. 1.

In addition, to evaluate the quality of the pseudo-samples, we propose the confusion of pseudo-samples (CPS) technique, a BLEU-based method for evaluating the weights of the previous tasks in the pseudo-samples. The CPS method can be used to determine the deviations in the pseudo-data by evaluating whether the data acquired from the learned tasks is uniform.

Our main contributions in this paper are detailed as follows: (1) We propose DFSD, which allows more useful knowledge to be learned from a small amount of noisy pseudo-data for incremental language learning and significantly outperforms previous state-of-the-art methods. (2) We propose SD-EMD for adaptively estimating the distribution of task knowledge in all layers and effectively transferring data from the teacher model to the student model via adaptive self-multilayer-to-multilayer mapping. This method can significantly reduce the dependence on retrieving pseudo-data. (3) We propose HDA for reducing deviations between pseudo-data and real data by decomposing the generation process and bootstrapping. (4) We investigate the effectiveness of SD-EMD and HDA. Moreover, we propose using CPS to evaluate the quality of the pseudo-data. The experimental results indicate that SD-EMD and HDA are complementary and can alleviate catastrophic forgetting by improving the training efficiency of the pseudo-data and reducing the noise, respectively.

2 Related work

Incremental learning is essential for advancing general artificial intelligence [21], and it has been applied in some real-world applications [22, 23]. Previous studies have investigated how to prevent catastrophic forgetting from different perspectives [24].

2.1 Data-based incremental learning

Data-based methods usually retrieve a small amount of previous task data during the training or inference process.
Gradient episodic memory (GEM) [7] preserves a small amount of data from previous tasks to limit loss at previous tasks. Average-GEM (A-GEM) [8] is a faster and more efficient version of GEM. Deep generative replay (DGR) [25] and FearNet [26] use generative models to remember the knowledge of previous tasks. Both models generate pseudo-data for previous tasks to alleviate catastrophic forgetting. MbPA++ [9] uses episodic memory to save some real data from previous tasks. The episodic memory data is retrieved during the training process and used for local interference adaptation. Meta-MbPA [10] improves MbPA++ by applying a meta-lifelong framework. Recently, LAMOL [11], which uses a language model (LM) to learn various kinds of NLP tasks in QA style, has been developed. In LAMOL, the pseudo-data generated by the model are trained alongside the new task.

Knowledge distillation has also been applied in the field of incremental learning. Learning without forgetting [6] minimizes the difference between shared parameters by constraining the outputs of old tasks for new data. Progress & Compress (P&C) [27] implements two networks: one network learns a new task, which is then distilled into another network, which stores all learned tasks. In the field of computer vision, Lifelong GAN [28] and generative replay with distillation [29] have been used for memory replay or generation-based approaches with distillation. In the field of natural language processing, L2KD [12] has been used for word-level and sequence-level knowledge distillation to transfer information from a teacher model to a student model that contains knowledge of previous tasks. L2KD trains a single-task model with a new task as a teacher model before it implements incremental learning. DnR [13] distills the attention and hidden state of some GPT-2 layers [17]. Both L2KD and DnR use knowledge distillation to improve LAMOL. In contrast to previous work, our SD-EMD model adaptively estimates the distribution of the task knowledge in all layers and effectively transfers this knowledge from the teacher model to the student model. SD-EMD improve the pseudo-data training efficiency, reducing the dependence on the number of pseudo-samples.

2.2 Regularization-based incremental learning

Regularization-based methods typically estimate the importance of the parameters. Elastic weight consolidation (EWC) [30] estimates the importance of the parameters for each task by calculating the Fisher information. Online EWC [27] is an improved method for estimating the relative importance of weights in a network. Synaptic intelligence (SI) [31] estimates the sensitivity of the parameters to the loss function to determine the importance of the parameter. Memory aware synapses (MAS) [32] estimate parameter importance based on the sensitivity of the output function. Instead of estimating the importance of the parameters, incremental moment matching (IMM) [33] incrementally matches the moment of the posterior distribution of the neural network. However, IMM is inefficient in terms of memory because the Gaussian posterior distributions must be obtained for all parameters. Thus, we propose HDA, a more computationally and memory efficient method for decomposing the generation of pseudo-data into an uncertain weighted mixture of all trained task data; we use this method to reduce the amount of noise in the pseudo-data.

3 Methodology

Before discussing DFSD in detail, we briefly introduce LAMOL [11], on which DFSD is based. Then, we introduce SD-EMD and compare it to other LAMOL-based distillation methods in Section 3.2. Following that, we describe how HDA can be used to decompose the generation of pseudo-data and reduce the amount of noise in Section 3.3, and we discuss how CPS can be used to evaluate the quality of the pseudo-data in Section 3.4. Finally, we describe how to use SD-EMD and HDA during the training process in Section 3.5.

3.1 LAMOL

LAMOL uses a single language model to complete QA and LM tasks. In a QA task, a sequence containing the context (C) and question (Q) is used as an input, with the model generating the corresponding answer (A). In the LM task, a complete sentence with C+Q+A is generated, starting from a task-specific token [TASK]. These two tasks are illustrated in Fig. 2. The loss function of LAMOL is $L_{\text{base}} = L_{\text{QA}} + \beta L_{\text{LM}}$, where $\beta$ is the weight of the LM task.

Beginning with the second task, whenever a new task is learned, LAMOL first generates pseudo-samples of previous tasks and then jointly trains the pseudo-data with the new task. As the representative of the learned tasks, these pseudo-data can prevent the model from forgetting the learned tasks. In this scenario, we are given a task order $\{D_1, D_2, \ldots\}$; however, we do not know the number of tasks. When training a new task $D_{\tau} (\tau > 1)$, the task-specific tokens of the previous tasks are used as the first token to input into the model; then, the pseudo-samples $X_{\tau-1}$ are generated via greedy decoding. $D_{\tau}$ is jointly trained with $X_{\tau-1}$ rather than real samples from previous tasks to alleviate catastrophic forgetting. The amount of pseudo-data is $\gamma |D_{\tau}|$, where $\gamma$ denotes the sampling ratio and $|D_{\tau}|$ is the amount of data in task $D_{\tau}$. Since all $\tau-1$ tasks share the $\gamma |D_{\tau}|$ pseudo-samples, the model generates $\frac{\gamma}{\tau-1} |D_{\tau}|$ pseudo-samples for each previous $\tau-1$ task.
3.2 Self-distillation based on EMD

For our DFSD method, we propose self-distillation based on EMD (SD-EMD) for LAMOL to reduce the dependence on the amount of pseudo-data. SD-EMD, one of the key concepts of our proposed DFSD method, adaptively transfers knowledge stored in some layers of the teacher model to other layers in the student model when learning a new task since the knowledge learned in different tasks is distributed in different layers, a process that is not considered in L2KD [12] and DnR [13], which used distillation to improve LAMOL. In this paper, we define this task-specific knowledge distribution as the inner-knowledge distribution (IKD). Formally, we let $W_{\text{in}} = \{\omega_i | i \in [1, K]\}$ denote IKD, where $\omega_i$ represents the knowledge weight of the $i$-th layer, $K$ is the number of GPT-2 layers and $\sum_{i=1}^{K} \omega_i = 1$. Furthermore, neither L2KD nor DnR focus on task transferability, and the transferability between layers differs. In this paper, we define the transferability between tasks for each layer as the outer-knowledge distribution (OKD). Formally, we let $W_{\text{out}} = \{f_{ij} | i, j \in [1, K]\}$ denote OKD, where $f_{ij}$ represents the knowledge weight transferred from the $i$-th layer of the teacher model to the $j$-th layer of the student model and $\sum_{j=1}^{K} f_{ij} \leq \omega_i$. The IKD and OKD of L2KD and DnR can be expressed as follows:

$$W^{L2KD}_{\text{in}} = \begin{cases} \omega_i = 0 & i < K \\ \omega_i = 1 & i = K \end{cases}$$

$$W^{L2KD}_{\text{out}} = \begin{cases} f_{ij} = 0 & \text{others} \\ f_{ij} = \xi & i, j = K; \exists \in [0, 1] \end{cases}$$

$$W^{DnR}_{\text{in}} = \begin{cases} \omega_i = 0 & i < K - 1 \\ \omega_i = \frac{1}{2} & i \in [K - 1, K] \end{cases}$$

$$W^{DnR}_{\text{out}} = \begin{cases} f_{ij} = 0 & \text{others} \\ f_{ij} = \frac{\xi}{2} & K - 1 \leq i = j \leq K; \exists \in [0, 1] \end{cases}$$

As shown in the above equations, the IKD is assumed to be concentrated in some layers in L2KD and DnR, and the $\omega_i$ values of these layers are the same. In DnR, the value of OKD is nonzero only in the selected layers, and there is no cross-layer knowledge transfer. However, [34, 35] illustrated that the knowledge distribution varies across different GPT-2 layers. Li et al. [36] demonstrated that this difference in the knowledge distribution is related to specific tasks in BERT [37]. Therefore, in this paper, we propose that the knowledge distribution of all GPT-2 layers should be considered during incremental language learning. However, the knowledge distribution of previous tasks shifts when the model learns a new task. As we cannot predict the next task during incremental learning, it becomes very expensive to manually design the IKD and OKD values to aid in distillation.

Thus, an iterable optimization algorithm that can obtain the mapping relationship for transferring knowledge from T to S with the lowest possible cost must be found. Moreover, to satisfy the data-free constraint, the incremental model should utilize as little data as possible from previously learned tasks when training a new task. The EMD [20] can measure the difference between two distributions by transferring knowledge between the distributions with the minimal accumulated cost. The pseudo-data containing the previously learned knowledge can be input into the model for distillation. Therefore, we propose SD-EMD as a method for transferring pseudo-data knowledge from the teacher model to the student model. We refer to the model that has been trained on all previous tasks as the teacher (T) model and the model that is being trained on the new task as the student (S) model. S is initialized based on the parameters of T. The student model learns new tasks while retaining the performance of previous tasks.
As a result, our SD-EMD model has three modules: the embedding layer, the attention layer, and the hidden state.

**Embedding Layer** Word embedding, a most fundamental task in NLP, maps words into semantic spaces.

The semantics of words are affected by the context and differ for different tasks. We minimized the semantic distance between words with similar contexts in T and S with the mean squared error. Let \( X = (x_1, ..., x_N) \) denote a pseudo-sample X with length N; let \( E(X)^T = (e_1^T, ..., e_i^T, ..., e_N^T) \) and \( E(X)^S = (e_1^S, ..., e_i^S, ..., e_N^S) \) denote the embedding of X in the teacher model and student model, respectively.

\[
L_{emb} = \frac{1}{N} \sum_i^N \| e_i^T - e_i^S \|^2
\]  

(1)

**Attention and Hidden State** GPT2 includes stacked transformer decoders, with the masked self-attention layer instructing the model to focus on only the information from the previous tasks. We transform the linguistic knowledge that is distributed in the masked attention matrix and hidden state matrix of different layers from T to S based on EMD. Let \( \mathbb{U}^T = \{U_1^T, \omega_1^T\}, ..., \{U_K^T, \omega_K^T\} \) denote the matrix and IKD of all layers in the teacher model and \( \mathbb{U}^S = \{U_1^S, \omega_1^S\}, ..., \{U_K^S, \omega_K^S\} \) denote the matrix and IKD of all layers in the student model. Next, let \( \mathbb{D} = \{d_{ij}\} \) denote the cost of transferring attention or hidden state knowledge from \( U_i^T \) to \( U_j^S \). Because the Jensen Shannon divergence (JSD) is symmetric and can be used to compare two different distributions, we can apply it to calculate \( d_{ij} \):

\[
d_{ij} = JSD(U_i^T, U_j^S) = \frac{1}{2} \left( U_i^T \log \frac{U_i^T}{U_j^S} + U_j^S \log \frac{U_j^S}{U_i^T} \right)
\]  

(2)

where \( i, j \in [1, K] \).

Next, we minimize the cumulative transformation cost to transfer the knowledge from T to S by solving the following optimization problem:

\[
\min \sum_i^K \sum_j^K d_{ij} f_{ij}
\]  

(3)

where \( f_{ij} \) represents the knowledge weight transfer from the \( i \)-th layer of T to the \( j \)-th layer of S. Therefore, we can define the result of the EMD process as follows:

\[
\text{EMD}(U^T, U^S) = \frac{\sum_i^K \sum_j^K d_{ij} f_{ij}}{\sum_i^K \sum_j^K f_{ij}}
\]  

(4)

Both the attention matrix and the hidden state matrix can be calculated with Eq. (4). The objective function can be written as follows:

\[
\mathcal{L}_{SD-EMD} = \text{EMD}(A^T, A^S) + \text{EMD}(H^T, H^S)
\]  

(5)

where \( A^T \) and \( A^S \) are the attention matrices of T and S, respectively, and \( H^T \) and \( H^S \) are the hidden state matrices of T and S, respectively.

We initialize \( \omega_i^T \) and \( \omega_i^S \) with \( \frac{1}{K} \) without any prior knowledge of the task. However, the IKDs of different NLP tasks have their own distributions. Thus, we use the attention cost [36] to update \( \omega_i^T \) and \( \omega_i^S \) to allow the linguistic knowledge of each layer to be mapped more accurately.

### 3.3 Hidden data augmentation

In addition to SD-EMD, we propose hidden data augmentation (HDA), another essential component of DFSD, as an innovative solution for reducing noise in the pseudo-samples. In Table 10 of Appendix A, we show real samples, good pseudo-samples, and noisy pseudo-samples (NPS). As an example of an NPS, for the task order SST → WOZ → SRL, the pseudo-data of SST and WOZ can be generated prior to training SRL. Thus, we can find that the content about WOZ is generated when the task-specific token is “_sst_”. This indicates that in the semantic space, “_sst_” is biased towards “_woz.en_” for SST, they are noisy samples which can cause the model to forget the previously learned SST data. Moreover, the NPS can cause irreversible catastrophic forgetting because the cascading errors accumulate as the number of learned tasks increases.

In previous studies on LAMOL, the negative impact of NPS has not been investigated. In this paper, we explore a method for reducing the noise by decomposing the generation of pseudo-data into an uncertain weighted mixture of all trained task data. GPT-2 is a parameter-fixed model that uses the same parameters to represent all learned tasks. Therefore, when GPT-2 is applied to learn different tasks sequentially, we can consider the knowledge of a new task to be weighted the same as the knowledge of previous tasks. Formally, let \( \phi_i = \sum_{i=1}^T \lambda_i \varphi_i \) denote the knowledge of \( \tau \) tasks in GPT-2, where \( \lambda_i \) represents the weight of the \( i \)-th task and \( \varphi_i \) represents the knowledge of the \( i \)-th task. Therefore, the three types of pseudo-samples can be simplified, as shown in Fig. 3.

According to Fig. 3, NPS are caused by the knowledge of previous tasks gradually being replaced by the knowledge
Fig. 3 Geometric illustration of a good pseudo-sample and a noisy pseudo-sample. \(e_i\) indicates the real sample of the \(i\)-th task, \(e_i^{\text{good}}\) and \(e_i^{\text{noisy}}\) indicate the good and noisy pseudo-samples of the \(i\)-th task, respectively. \(\lambda_i\) and \(\lambda_i'\) indicate the weight of the \(i\)-th task

of new tasks. This indicates that \(\lambda_i > \lambda_{i-1}\). For a specific task, the knowledge of other tasks can be regarded as noise. Therefore, we propose addressing the catastrophic forgetting due to NPS based on noisy labels and data augmentation. Since \(\lambda_i\) is a hidden variable, this method is referred to as hidden data augmentation (HDA).

Ideally, the task-specific token \(\psi_{\text{task}}\), the starting token for generating pseudo-samples, can produce task-specific pseudo-samples \(\chi_{\text{task}}\) that are similar to real task-specific samples \(\chi_{\text{task}}^{\text{real}}\) in terms of content and grammar. However, previous task-specific token embedding can shift after new tasks are learned, resulting in uncontrollable scale noise in the pseudo-samples. Since this noise is generated by learning new tasks, we regard the deviation of \(\psi_{\text{task}}\) in the semantic space as a data mixing process. This process also implies that the knowledge of a new task is weighted with the knowledge of previous tasks. Formally, let \(\chi^\tau = \{\chi^\tau_i | i \in [1, \tau]\}\) denote the pseudo-samples of all previous tasks after the model has learned \(\tau\) tasks, and let \(i\) denote the index of a previous task. Let \(\Psi^\tau = \{\psi_1, ..., \psi_{\tau}\}\) and \(\Phi^\tau = \{\phi_1, ..., \phi_{\tau}\}\) denote the set of task-specific tokens and the set of each task in the semantic space, respectively, where \(\tau\) denotes the number of tasks. Then, the pseudo-data can be generated as follows:

\[
\chi_i^\tau = \sum_{\psi_i = \phi_1} G(E_{\tau-1}(\psi_i), \theta_{\tau-1}) \\
E_{\tau}(\psi_i) = \rho(\psi_i | \theta_{\tau}) \\
\theta_{\tau} = \sum_{i=1}^{\tau} \lambda_i \phi_i (s.t. \sum_{i=1}^{\tau} \lambda_i = 1)
\]

Equation (6) indicates that the generation of pseudo-samples for each task is affected by all learned tasks. (7) represents the embedding of \(\psi_i\) based on \(\theta_{\tau}\), which represents the parameters of the model after learning \(\tau\) tasks. Therefore, information from previous tasks is combined in \(\chi_i^\tau\) in a specific ratio after \(\tau\) tasks are trained.

We define \(\Lambda_i^\tau = \{\lambda_{i1}, ..., \lambda_{i\tau}\}\) to represent the weights of each previous task when generating pseudo-samples for the \(i\)-th task after \(\tau\) tasks have been learned. The \(i\)-th task pseudo-sample can be generated as follows:

\[
\hat{x}_i^\tau = \sum_{j=1}^{\tau} \lambda_{ij} x_{ij}, \quad \hat{x}_i^\tau \in \chi_i^\tau \\
x_{ij} = G(E_{\tau}(\psi_i), \phi_{\tau})
\]

where \(x_{ij}\) represents the pseudo-sample generated by a model trained only on the \(j\)-th task. We assume that the generation of \(x_{ij}\) is an unobservable hidden process. Since this mixing process is performed by the model automatically, we do not know the true value of \(\Lambda^\tau\). Therefore, each word in the pseudo-sample can be regarded as a combination of information from previous tasks, which can prevent \(S\) from forgetting the knowledge learned from previous tasks. These words are regarded as pseudo-authentic labels for \(S\) provided by \(T\). Therefore, the loss of \(\hat{x}_i^\tau\) can be calculated by the standard cross-entropy loss as follows:

\[
\hat{\ell}(\hat{x}_i^\tau) = \sum_{j=1}^{\tau} \lambda_{ij} \ell_{ij} \\
\ell_{ij} = -y_{ij}^\tau \log(h(x_{ij}|\theta_S^\tau))
\]

However, according to [38], the standard cross-entropy loss causes the model to label noise. Thus, the model fit is \(x_{ij} | j \in [1, \tau], j \neq i\) when \(\hat{x}_i^\tau\) is learned. The bootstrapping loss proposed in [39] is useful for correcting the training objective. Therefore, the bootstrapping loss is used to ensure that the model fits \(x_{ij}\). (12) can be improved as follows:

\[
\ell_{ij} = -\zeta(i, j) \log(h(x_{ij}|\theta_S^\tau)) \\
\zeta(i, j) = \begin{cases} 
(\rho y_{ij}^\tau + (1 - \rho) y_i^S), & i = j \\
((1 - \rho) y_{ij}^\tau + \rho y_i^S), & i \neq j 
\end{cases}
\]
where $\rho$ weights the prediction of $T$. Then, (11) can be represented as follows:

$$\hat{\ell}(\hat{x}_i^T) = - (\alpha y_i^T + \alpha S y_i^S) \log(h(\hat{x}_i^T | \theta_i^S))$$

(15)

$$\alpha^T = \rho \lambda_{ii} + \sum_{j \neq i} (1 - \rho) \lambda_{ij}$$

(16)

$$\alpha^S = (1 - \rho) \lambda_{ii} + \sum_{j \neq i} \rho \lambda_{ij}$$

(17)

We find that $\alpha^T + \alpha^S = 1$. To increase the smoothness, we use the predicted temperature-softmax probabilities $\hat{h}^T$ and $\hat{h}^S$ rather than the token predictions $y^T$ and $y^S$. Therefore, (15) can be simplified to (18):

$$\hat{\ell}(\hat{x}_i^T) = - (\alpha \hat{h}_j^T + (1 - \alpha) \hat{h}_i^S) \log(h(\hat{x}_i^T | \theta_i^S))$$

(18)

Finally, the objective function of HDA can be defined as follows:

$$\mathcal{L}_{HDA}(X_t) = \frac{1}{|X_t|} \sum_{i=1}^{r} \sum_{j=1}^{s_i} \hat{\ell}(\hat{x}_i^T)$$

(19)

### 3.4 Evaluation metric for the pseudo samples

In this subsection, we introduce a BLEU-based method for evaluating the quality of the pseudo-samples by calculating the weights of all learned tasks for each pseudo-sample. In Section 3.3, we regarded each pseudo-sample as containing a mixture of data from previous tasks. However, this mixture is hidden and thus cannot be evaluated directly because only generated samples can be directly observed and discretely compared to the data of previous tasks. This judgment is not smooth. Moreover, it is expensive to manually evaluate each pseudo-sample. Therefore, a quantitative analysis method for evaluating the amount of knowledge from each previous task in the pseudo-samples must be developed. BLEU [40], which is widely used for evaluating the quality of generated text, can be used to evaluate the pseudo-data. Let $P = \{p_1, ..., p_r\}$ denotes the weights of the previous tasks in the pseudo-samples after the model has learned $T$ tasks. We can calculate $P$ with the following equations:

$$p_k = \frac{s_k}{\sum_{i=1}^{r} s_i}, \quad p_k \in P$$

(20)

$$s_k(\mathcal{X}_t, D_k) = \frac{1}{|\mathcal{X}_t|} \sum_{i=1}^{r} \sum_{j=1}^{s_i} \text{BLEU}(\hat{x}_i^T, D_k)$$

(21)

where $D_k$ denotes the training set of the $k$-th task. In a set of good pseudo-samples, the proportion of knowledge from each previous task should be approximately uniform because each task is equally important. Without considering the task boundary, we consider the pseudo-data to be acceptable even if the pseudo-data of the tasks deviate as long as the pseudo-data set contains uniform knowledge from all learned tasks.

We choose the JSD, as in (2), to evaluate the difference between $P$ and the uniform distribution $U = \{\frac{1}{r}, ..., \frac{1}{r}\}$. This difference is known as the Confusion of Pseudo-Samples (CPS):

$$CPS = JSD(P, U)$$

(22)

In Table 1, (1) and (2) are good and noisy pseudo-samples for SST, respectively, while (3) and (4) are good and noisy pseudo-samples for WOZ, respectively. In Table 2, we show some combinations of different examples from Table 1 to calculate the CPS. Each example is classified as either good pseudo-data or noisy pseudo-data based on the corresponding task. The goal is for the model to generate an ideal set of pseudo-data, as in case 1. However, in reality, some pseudo-data from previous tasks are similar to the newly learned task, which is similar to cases 2 and 3. In the two cases mentioned above, the CPS of the actual set of pseudo-samples was significantly higher than that of the ideal set of pseudo-samples. This indicates that if the pseudo-data are more biased towards a certain task, the CPS value is higher. Therefore, our proposed CPS method can evaluate the quality of the pseudo-samples.

### 3.5 Training process

Based on LAMOL, our DFSD model trains the QA and LM tasks simultaneously. The complete objective function is expressed as follows:

$$\mathcal{L} = \mathcal{L}_{base} + \mu(\mathcal{L}_{SD-EMD}^{QA} + \mathcal{L}_{EMD}^{LM})$$

$$+ \delta(\mathcal{L}_{HDA}^{QA} + \mathcal{L}_{HDA}^{LM})$$

(23)

where $\mu$ and $\delta$ denote the SD-EMD and HDA parameters, respectively. The training process is detailed in Algorithm 1.

| Table 1 BLEU of the pseudo-samples after training SST and WOZ |
|-----------------|----------|----------|
| Pseudo-samples  | BLEU$_{SST}$ | BLEU$_{WOZ}$ |
| (1) _sst_. the movie is simply too stupid to be worth your time. is this review negative or positive? _ans_. negative | 0.815 | 0 |
| (2) _sst_. can you recommend me a cheap place to eat? what is the change in state? _ans_. price range : cheap | 0 | 0.871 |
| (3) _woz.en_. a restaurant that serves european food. what is the change in state? _ans_. food : european | 0 | 0.987 |
| (4) _woz.en_. it’s not a surprise, as the film’s premise is simply too familiar. is this review negative or positive? _ans_. negative | 0.695 | 0 |
Table 2  Some cases for calculating the Confusion of Pseudo-Samples (CPS)

| Case ID | Combination | \( \mathcal{P} \)          | CPS  |
|---------|-------------|-----------------------------|------|
| 1       | \{(1) × 3, (3) × 3\} | \{(0.452, 0.548)\}          | 0.005|
| 2       | \{(1) × 2, (2) × 1, (3) × 3\} | \{(0.298, 0.702)\}          | 0.086|
| 3       | \{(1) × 1, (2) × 2, (3) × 3\} | \{(0.148, 0.852)\}          | 0.309|

Algorithm 1  Data-free self-distillation.

**Input:** a new task dataset \( D_t \), a teacher model with parameters \( \Theta^T_{t-1} \), the LAMOL loss function \( \mathcal{L}_{base} \), a self-distillation function based on the EMD loss function \( \mathcal{L}_{SD-EMD} \) and its factor \( \mu \), the hidden data augmentation loss function \( \mathcal{L}_{HDA} \) and its factor \( \delta \), and the pseudo-data sampling ratio \( \gamma \).

**Output:** the student model with parameters \( \Theta^S_t \).

1:  Initialize \( \Theta^S_t = \Theta^T_{t-1} \).
2:  Sample \( k = \gamma \cdot |D_t| \) pseudo-data from \( \Theta^T_{t-1} \) to build \( \mathcal{X}_{t-1} \).
3:  for all training samples \( \mathcal{X}^y \in D_t \) and \( \hat{\mathcal{X}}^c \) do
   4:      \( \Theta^S_t \leftarrow \mathcal{L}_{base}(\mathcal{X}^y; \hat{\mathcal{X}}^c; \Theta^S_t) \)
   5:      \( \Theta^S_t \leftarrow \mu \mathcal{L}_{SD-EMD}(\hat{\mathcal{X}}^c; \Theta^T_{t-1}; \Theta^S_t) \)
   6:      \( \Theta^S_t \leftarrow \delta \mathcal{L}_{HDA}(\hat{\mathcal{X}}^c; \Theta^T_{t-1}; \Theta^S_t) \)
7:  end for
8:  return \( \Theta^S_t \)

4 Experimental setup

To compare the results, we built our DFSD based on LAMOL.\(^1\) We used a task-specific token as the starting token for the pseudo-samples. The HDA and SD-EMD factors, \( \delta \) and \( \mu \), were 0.5 and 0.08, respectively. We set the learning rate to 1e-4 and \( \alpha \), the confidence of \( T \) in HDA, to 0.9.

4.1 Experimental data

Based on LAMOL [11], we used five different NLP tasks from decaNLP [16] to evaluate our proposed DFSD model: SQuAD, WikiSQL, WOZ, QA-SRL, and SST. These five datasets include different NLP tasks trained in a random order, which we used to evaluate the robustness of our proposed method for various NLP tasks. In addition, we also evaluated the DFSD model on Amazon, AGNews, DBPedia, Yahoo and Yelp, which are five classic text classification datasets [41]. We used balanced versions of the datasets from [9]. The normalized F1 (nF1) metric, which reduces the amount of text and removes the punctuation and articles, was used to evaluate SQuAD and SRL. The logical form exact match (lFEM) was used to evaluate WikiSQL. The exact match (EM) was used to evaluate SST and text classification. The turn-based dialogue state EM (dsEM) was used to evaluate WOZ. The dataset information is shown in Table 3.

4.2 Baselines

- **Fine-tuned:** Fine-tune GPT-2 individually based on the task order.
- **LAMOL** [11]: Only naive QA and LM tasks are used to train the general incremental language model. In LAMOL\(G\) and LAMOL\(T\), the first token is the [GEN] token and the task-specific token, respectively. LAMOL\(R\) indicates that real samples from previous tasks were trained with the new task instead of pseudo-samples.
- **L2KD** [12]: An improved version of LAMOL with distillation; a single-task model is trained on the current task for the teacher model, while it is trained on the previous tasks for the student model. This model uses the task-specified token.
- **DnR** [13]: An improved version of LAMOL with distillation and replay that distills some layers. We determine the optimal implementation by distilling the final two layers with naive matching. This model uses the [GEN] token.
- **MbPA++** [9]: MbPA++ stores some data from previous tasks in episodic memory and retrieves this data during training and inference. We determined the optimal setting and performance for text classification.

Table 3  Dataset sizes and evaluation metrics. EM denotes an exact match between different texts. nF1 is a normalized version of F1 with lower text and remove punctuation and articles. lFEM and dsEM are the logical form EM and turn-based dialogue state EM, respectively

| Dataset | #Train | #Test | Metric |
|---------|--------|-------|--------|
| Different NLP Tasks | | | |
| SQuAD | 87599 | 10570 | nF1 |
| WikiSQL | 56355 | 15878 | lFEM |
| WOZ | 2536 | 1646 | dsEM |
| QA-SRL | 6414 | 2201 | nF1 |
| SST | 6920 | 1821 | EM |
| Different Datasets for Text Classification | | | |
| Amazon | 115000 | 7600 | EM |
| AGNews | 115000 | 7600 | |
| DBPedia | 115000 | 7600 | |
| Yahoo | 115000 | 7600 | |
| Yelp | 115000 | 7600 | |

\(^1\)https://github.com/jojotenya/LAMOL
– **Meta-MbPA** [10]: Meta-MbPA uses a meta-lifelong framework to improve MbPA++. We determined the optimal setting and performance for text classification.
– **Multitask**: Multitask learning trains all tasks simultaneously and is often regarded as the upper bound of lifelong learning.

### 5 Results and analysis

First, we evaluated our DFSD model with three experiments; these settings of these experiments have been used in previous works [9–13]. Then, we validated the effectiveness of SD-EMD and HDA.

#### 5.1 Results on three DecaNLP tasks

To comprehensively evaluate the performance of our proposed method, we first selected three datasets [SST, SRL, and WOZ] in decaNLP [16], following [11, 13]. To eliminate the influence of the training order, we trained the models in six different orders. When training each order, we tested the model after the final task was trained and calculated the average of the three task scores to determine the performance. The average of all orders was used to indicate the performance of the current model.

Sun et al. [11] demonstrated that a larger sampling ratio $\gamma$ had a positive effect on the performance of the model, while the gain decreased when $\gamma$ was approximately 0.1 to 0.3. When the value of $\gamma$ is less than 0.1, the performance of the model decreases significantly as $\gamma$ decreases. To validate the effectiveness of our proposed DFSD model for lower values of the sampling ratio $\gamma$, we conducted experiments for $\gamma \in [0.01, 0.02, 0.03, 0.05, 0.2]$, where $\gamma = 0.2$ is chosen because it was the best setting in LAMOL-based baselines’ paper. The results are shown in Table 4.

As shown in Table 4, the fine-tuned model performed poorly on the three task datasets, and the order of learning had a considerable impact on the task performance. The incremental language model LAMOL-based baselines performed better than the fine-tuned model. However, the performance still lags behind multitask learning (MTL). This gap increased significantly as the sampling ratio $\gamma$ decreased. When $\gamma = 0.2$, all the baselines performed well, with DnR approaching the performance of MTL. However, DFSD outperformed all LAMOL-based baselines and even surpassed MTL. This demonstrates that DFSD can capture more relevant information between tasks, achieving positive forward and backward transfer between tasks. When $\gamma = 0.05$, L2KD was the best baseline; however, the performance was lower than LAMOL$_R$ and MTL by 3% and 5.5%, respectively. However, DFSD exceeded all LAMOL-based baselines and was only 0.12% worse than MTL. The standard deviation (std.) of DFSD was 0.6, which was lower than the std of the other LAMOL-based baselines. As $\gamma$ decreased, the performance of all LAMOL-based baselines decreased significantly; however, DFSD had a robust performance. We use subscripts to indicate the value of $\gamma$. We find that DFSD$_{0.02}$ outperformed all LAMOL-based baselines in terms of in average performance and std when $\gamma \in [0.02, 0.03, 0.05]$. Furthermore, DFSD$_{0.01}$ outperformed all LAMOL-based baselines in terms of in average performance and std when $\gamma \in [0.01, 0.02, 0.03]$. This indicates that DFSD is more sample-efficient. Thus, DFSD can generate nearly perfect quality pseudo-data and learn the pseudo-data more efficiently.

In terms of single tasks, WOZ was forgotten more easily in all LAMOL-based baselines, especially when WOZ was the first task. We conclude that WOZ is a more difficult task because the dialogue system is complex in terms of semantics and context. A good WOZ pseudo-sample is shown in Table 10; our goal was to generate pseudo-samples similar to this example. However, inefficient pseudo-data, such as “thank you” and “goodbye”, were generated. These inefficient pseudo-data are easily replaced by new tasks. As shown in Table 4, when WOZ was learned first, DFSD exceeded all LAMOL-based baselines by 5-51%. The smaller the value of $\gamma$, the clearer the advantage of DFSD. Therefore, DFSD is more sample-efficient when learning more knowledge from a small number of pseudo-samples to alleviate catastrophic forgetting. Additional experiments and analyses about the two components of DFSD are discussed in Sections 5.4 and 5.5.

Overall, DFSD has potential in practical applications with data-free constraints. When compared with other LAMOL-based methods, DFSD can approach MTL, with 75% less pseudo-data while working with much less pseudo-data.

#### 5.2 Results on longer sequences and larger datasets in different NLP tasks

As the number of learned tasks increases, previously learned tasks are more likely to be forgotten. We verified the effectiveness of DFSD after more tasks had been learned. We choose [SQuAD, WikiSQL, SST, SRL, and WOZ] based on [11, 13]. Because SQuAD and WikiSQL are both significantly larger than the other datasets, when they are a non-first task, $\gamma|D_{\text{task}} \in \{\text{SQuAD, WikiSQL}\} \geq |D_{\text{task}} \in \{\text{SST, SRL, WOZ}\}|$. Therefore, due to the large amount of generated pseudo-data, the model becomes similar to multitask learning (the only difference is whether the model is trained with real data or pseudo-data). Therefore, when SQuAD and WikiSQL are first, the training difficulty increases. We investigated SQuAD and WikiSQL as the beginning or end of all the orders; this results in 24 orders...
Table 4 Three decaNLP tasks under five sampling ratios $\gamma$. Each column represents a permutation of [SST, SRL, and WOZ]. After training the model in each order, the model of the last task was determined by averaging across the three tasks. Average and Std represent the average and standard deviation of the six orders, respectively. The better performance is in boldface.

| $\gamma$ | Methods | SST WOZ | SRL WOZ | SRL SST | WOZ SRL | WOZ SST | Average | Std |
|---------|---------|---------|---------|---------|---------|---------|---------|-----|
|        | Finetune | 50.2 | 24.7 | 62.9 | 31.3 | 32.8 | 33.9 | 39.3 | 12.0 |
| 0.01   | LAMOL$_G$ | 72.5 | 71.2 | 63.6 | 70.8 | 52.3 | 57.7 | 64.7 | 7.6 |
|        | LAMOL$_T$ | 73.7 | 72.2 | 64.2 | 70.1 | 51.9 | 60.9 | 65.5 | 7.5 |
|        | LAMOL$_R$ | 74.0 | 70.5 | 73.5 | 74.5 | 59.1 | 64.2 | 69.3 | 5.7 |
|        | L2KD | 72.1 | 72.3 | 61.6 | 70.4 | 53.8 | 60.8 | 65.2 | 6.9 |
|        | DnR | 76.5 | 71.9 | 66.7 | 70.3 | 52.8 | 67.9 | 67.7 | 7.4 |
|        | DFSD | **80.7** | **79.9** | **77.8** | **77.4** | **72.7** | **73.1** | **76.9** | **3.0** |
| 0.02   | LAMOL$_G$ | 76.8 | 71.5 | 68.0 | 70.8 | 60.1 | 69.7 | 69.5 | 5.0 |
|        | LAMOL$_T$ | 68.3 | 73.8 | 70.3 | 71.9 | 52.6 | 67.3 | 67.4 | 6.9 |
|        | LAMOL$_R$ | 76.1 | 74.1 | 75.5 | 73.8 | 71.5 | 76.5 | 74.6 | 1.7 |
|        | L2KD | 72.5 | 70.8 | 72.3 | 73.4 | 67.0 | 74.9 | 71.8 | 2.5 |
|        | DnR | 78.1 | 78.4 | 69.8 | 74.0 | 60.8 | 73.7 | 72.5 | 6.0 |
|        | DFSD | **81.3** | **80.8** | **79.6** | **78.9** | **79.3** | **78.4** | **79.7** | **1.0** |
| 0.03   | LAMOL$_G$ | 77.3 | 73.6 | 70.3 | 72.3 | 59.2 | 71.2 | 70.7 | 5.6 |
|        | LAMOL$_T$ | 76.0 | 74.8 | 71.6 | 74.3 | 57.5 | 70.6 | 70.8 | 6.2 |
|        | LAMOL$_R$ | 77.3 | 75.2 | 76.1 | 75.4 | 73.5 | 77.3 | 75.8 | 1.3 |
|        | L2KD | 77.3 | 76.8 | 74.4 | 75.9 | 73.0 | 74.6 | 75.3 | 1.5 |
|        | DnR | 78.7 | 78.0 | 71.2 | 72.6 | 68.0 | 76.5 | 74.2 | 3.9 |
|        | DFSD | **81.2** | **81.4** | **80.2** | **80.1** | **79.9** | **80.4** | **80.5** | **0.6** |
| 0.05   | LAMOL$_G$ | 79.6 | 78.9 | 73.1 | 73.7 | 68.6 | 75.7 | 74.9 | 3.4 |
|        | LAMOL$_T$ | 77.3 | 76.9 | 78.1 | 74.7 | 73.4 | 75.8 | 76.0 | 1.5 |
|        | LAMOL$_R$ | 81.0 | 78.9 | 80.1 | **80.9** | 77.7 | 78.0 | 79.4 | 1.2 |
|        | L2KD | 78.3 | 77.4 | 77.9 | 78.4 | 73.0 | 76.8 | 77.0 | 1.9 |
|        | DnR | 78.7 | 78.1 | 73.8 | 74.7 | 70.1 | 77.5 | 75.5 | 3.0 |
|        | DFSD | **81.5** | **82.4** | **80.8** | 80.5 | **81.7** | **81.5** | **81.4** | **0.6** |
| 0.2    | LAMOL$_G$ | 80.0 | 80.7 | 79.6 | 78.7 | 78.4 | 80.5 | 79.7 | 0.8 |
|        | LAMOL$_T$ | 79.4 | 79.9 | 80.1 | 78.7 | 79.8 | 79.0 | 79.5 | 0.5 |
|        | LAMOL$_R$ | 81.8 | 80.6 | **81.6** | 81.2 | 80.4 | 80.5 | 81.0 | 0.5 |
|        | L2KD | 80.1 | 79.6 | 79.5 | 79.7 | 79.9 | 80.4 | 79.9 | **0.3** |
|        | DnR | 81.4 | 82.1 | 81.2 | 81.1 | 80.9 | 81.6 | 81.4 | 0.4 |
|        | DFSD | **81.9** | **82.6** | 81.4 | **81.5** | **82.3** | **82.1** | **82.0** | 0.4 |

/ Multitask 81.5

in total. We choose $\gamma = 0.05$ because DFSD approached multitask learning with this value, as shown in Table 4. Table 5 shows that the experimental results are consistent with the expected results. The order from large datasets to small datasets is more challenging than the order from small datasets to large datasets. However, DFSD still has a robust performance for these sequences.

According to Table 4, the performance of DFSD begins to decrease when $\gamma = 0.01$. To demonstrate the effectiveness of DFSD for longer learning orders, we used a smaller value of $\gamma = 0.01$ and the most difficult order to Table 5 Performance of all orders of [SQuAD (0), WikiSQL (1), SST (2), SRL (3), WOZ (4)] when SQuAD and WikiSQL are first or last. # is a placeholder. We train the model with DFSD in these orders for $\gamma = 0.05$. Each score in the table indicates the performance of the model on the last task by averaging the five tasks.

| Sub-order | 234 | 243 | 324 | 342 | 423 | 432 |
|-----------|-----|-----|-----|-----|-----|-----|
| 01### | 74.8 | 75.4 | 75.1 | 75.2 | 75.1 | 75.6 |
| 10### | 75.7 | 75.5 | 75.2 | 75.1 | 75.2 | 75.1 |
| ####01 | 75.0 | 75.2 | 75.4 | 75.1 | 75.1 | 75.1 |
| ####10 | 76.8 | 76.7 | 76.7 | 76.7 | 76.2 | 75.8 |
compare with previous methods. The order was from the largest to the smallest dataset: SQuAD → WikiSQL → SST → SRL → WOZ. As the value of $\gamma |D_T|$ decreases and more tasks require the generation of pseudo-data, the order becomes more difficult to learn. As shown in Fig. 4, our DFSD model is also robust when $\gamma = 0.01$. When $\gamma = 0.01$, we observe that the other LAMOL-based baselines forget the SQuAD and WikiSQL datasets after more tasks are learned, while the DFSD model remains effective, especially when learning WOZ. When WOZ is learned, $\gamma |D_{WOZ}| = 25$, and there are only 6 pseudo-samples corresponding to each previous task. The DFSD model can learn and transfer knowledge efficiently with less pseudo-samples. Moreover, the SST results show that the DFSD model can transfer more knowledge from the SQuAD and WikiSQL datasets than the other methods. The SRL results

Fig. 4 Model performance after each epoch in the sequence SQuAD → WikiSQL → SST → SRL → WOZ for the sampling ratio $\gamma = 0.01$.
show that the DFSD model can transfer more knowledge in the SQuAD, WikiSQL and SST datasets than the other methods.

To investigate the influence of $\gamma$, we used smaller and larger values of $\gamma$ in the DFSD model. As shown in Fig. 5, DFSD is highly robust to $\gamma$ and can stably alleviate catastrophic forgetting. When $\gamma \leq 0.002$, the DFSD performance begins to deteriorate significantly. This is due to the fact that when $\gamma \leq 0.002$, $|D_{WOZ}| \leq 5$. Thus, there is at most one pseudo-sample that corresponds to each previous task. This indicates that DFSD has the potential to be used for one-shot or zero-shot learning. One-shot and zero-shot pseudo-samples will be explored in future work.

**5.3 Results on different datasets for text classification**

We validated the effectiveness of our DFSD method for training different datasets on one NLP task. We compared the DFSD model with LAMOL, L2KD, DnR, MbPA++ and Meta-MbPA. We trained the models in 4 different orders based on [9]. The orders are detailed in Table 6. For the
DFSD model, when we trained a new task, we randomly sampled only 10% of the data in each epoch due to a lack of computing resources. Thus, although the model was studied for 9 epochs, this is equivalent to learning the whole dataset only once. In addition, for the DFSD model, the same γ is equivalent to 10% of the γ of the other methods. We compared the results with γ = 0.2 because it performed better in LAMOL, L2KD, and DnR.

As shown in Table 7, our DFSD method outperformed previous state-of-the-art methods. As shown in Fig. 6, our DFSD method improved the previous task by learning the new task when training different datasets on the same NLP task. For instance, the Amazon and Yelp performance improved due to the positive information transfer from the other tasks. This indicates that DFSD can use similar tasks to accelerate the training process and improve the performance. The training curves for all orders are provided in Appendix B.

### 5.4 Effectiveness of self-distillation based on EMD

In this subsection, we present experiments on EMD-based self-distillation to prove the effectiveness of the method. We compare our proposed distillation method with two previous state-of-the-art methods, L2KD and DnR, which use other distillation methods to improve LAMOL. The order SST→WOZ→SRL is used in this subsection. We used 3 different values of γ to obtain the average score, with each method run 10 times with different seeds. As shown in Table 8, when γ = 0.01, both L2KD and DnR barely improved LAMOL, while SD-EMD increased the performance of LAMOL by at least 5%. This indicates that SD-EMD can obtain knowledge from fewer pseudo-samples. When γ = 0.03, while both L2KD and DnR improved LAMOL, SD-EMD increased the LAMOL performance by at least 2% more than these methods. When γ = 0.05, the results are similar to when γ = 0.01, with SD-EMD increasing the LAMOL performance by 2.4-4.4%. SD-EMD is effective regardless of the number of pseudo-samples used. We investigated the following reasons to analyze the results of the 3 sampling ratio γ values: (1) In L2KD, the teacher model and student model include the new task and the previous tasks, respectively. Because only the knowledge of the new task is distilled, the knowledge of previously learned tasks is not sufficient when γ is small. (2) DnR, similar to SD-EMD, is opposite to L2KD, with the teacher model learning the previous tasks. However, DnR only distills the attention layer and hidden state of a subset of the layers and does not consider the \( W_{in}^{\text{in}} \) and \( W_{out}^{\text{out}} \) of different tasks in GPT-2 (see \( W_{in}^{\text{in}} \) and \( W_{out}^{\text{out}} \) in Section 3.2). Thus, DnR has no effect on the condition that \( W_{in}^{\text{in}} \) is far from the ideal IKD=\( W_{in}^{\text{in}} \). (3) SD-EMD not only considers the previous tasks in self-distillation but also dynamically updates the IKD and OKD, allowing the knowledge of the pseudo-data to be obtained more accurately; thus, SD-EMD is still effective when γ is small (e.g., γ = 0.01).

SD-EMD can determine and update IKD and OKD during training. We visualized the IKD and OKD of all orders of \([\text{SST, WOZ, SRL}]\) for SD-EMD\(_{0.05}\). In Fig. 7(1)-(4), we show the IKD of the attention layer (A) and the hidden states (H) for the order SST→WOZ→SRL. We discover that the IKDs of different tasks are mainly distributed in the first three layers, regardless of the task order. It should be noted that \( o_{1} \), the knowledge weight of the first layer, has the highest value in all orders. Our analysis indicates that this result can be attributed to the fact that the knowledge extracted from the shallow layers is more generalizable. The knowledge extracted from the shallow layers is shared by different tasks, whereas the knowledge extracted from deeper layers is more task-specific. The IKDs of all orders are essentially consistent with Fig. 7(1)-(4) and are provided in Appendix C.

For both A and H, the OKD results shown in Fig. 7b, d, f and h flow between the teacher layers and the corresponding student layers, with no cross-flow between the layers. This finding shows that the knowledge learned in each GPT-2 layer is from different levels. Figure 7a, e shows the distance distribution of H between the teacher model and the student model. According to this figure, the first two layers and the last layer are very similar and the middle layers are similar, indicating that the features extracted from the shallow layers are similar. As the output layer needs to map the features to a vector with the same length as the vocabulary, the knowledge contained in the output layer is similar to the knowledge in the shallow layers. This process is similar to human learning in that

---

**Table 6** Four orders for the text classification task

| ID | Task order                        |
|----|----------------------------------|
| i  | Yelp → AGNews → DBPedia → Amazon → Yahoo |
| ii | DBPedia → Yahoo → AGNews → Amazon → Yelp |
| iii| Yelp → Yahoo → Amazon → DBPedia → AGNews |
| iv | AGNews → Yahoo → Amazon → DBPedia → Yahoo |

**Table 7** Summary of the text classification results. The superscript indicates the percent of the dataset used during training, and the subscript indicates the sampling ratio γ. The better performance is in boldface

| Order          | i          | ii         | iii         | iv          | Avg.       |
|----------------|------------|------------|-------------|-------------|------------|
| MbPA+++\(_{1.0}\) | 70.8       | 70.9       | 70.2        | 70.7        | 70.7       |
| Meta-MbPA\(_{1.0}\) | 77.9       | 76.7       | 77.3        | 77.6        | 77.4       |
| LAMOL\(_{1.0}\)  | 76.7       | 77.2       | 76.1        | 76.1        | 76.5       |
| L2KD\(_{1.0}\)   | 76.8       | 77.0       | 75.8        | 77.8        | 76.9       |
| DnR\(_{1.0}\)    | 77.4       | 77.2       | 77.1        | 76.9        | 77.2       |
| DFSD\(_{0.1}\)   | 77.6       | **77.4**   | **77.8**    | **77.9**    | **77.7**   |

---

Springer
simple and common knowledge is learned first; then, complex knowledge is deeply learned; and finally, complex knowledge is simplified and transformed into common knowledge. In Fig. 7c, g, the attentions of the different layers are far apart, which indicates that different layers pay attention to the semantics of different levels.

### 5.5 Effectiveness of hidden data augmentation

In this subsection, HDA is applied to previous state-of-the-art (SOTA) methods, and the performance with and without HDA is compared. The order SST→WOZ→SRL was tested in this subsection. We chose 3 different values of \( \gamma \) and determined the average score, with each HDA method run 10 times with different seeds. As shown in Table 9, HDA improved all baselines as well as our SD-EMD model. This overall improvement demonstrates the effectiveness of HDA. Moreover, the gap between LAMOL\(_T\) and LAMOL\(_G\) can be narrowed by applying HDA. This indicates that HDA can mitigate the impact caused by the difference between the first tokens \( \psi_{\text{task}} \) and \( \psi_{\text{GEN}} \).

| Methods  | \( \gamma \) | 0.01       | 0.03       | 0.05       |
|----------|-------------|------------|------------|------------|
| LAMOL\(_G\) | 71.9±0.14   | 73.6±0.39  | 78.9±0.34  |
| LAMOL\(_T\) | 72.2±0.58   | 74.8±0.40  | 76.9±0.25  |
| L2KD      | 72.3±0.45   | 76.8±0.36  | 77.4±0.28  |
| DnR       | 71.9±0.36   | 78.0±0.17  | 78.1±0.17  |
| SD-EMD    | 77.5±0.57   | 80.6±0.75  | 81.3±0.40  |
Fig. 7 Visualization of self-distillation based on EMD when the sampling ratio $\gamma = 0.05$. For the learning order SST $\rightarrow$ WOZ $\rightarrow$ SRL, Fig. 7(1)-(4) shows the inner-knowledge distribution (IKD) of the attention layer (A) and the hidden states (H) for each task. Figure 7a, e and b, f show the distance H between the teacher model and the student model for each layer and the OKD in the QA task, respectively. Figure 7c, g and d, h illustrate the results of A task boundary for the pseudo-samples if the first token is $\psi_{GEN}$. This is detrimental to ensuring that previous tasks are not forgotten. However, the performance of LAMOL$_G$ and DnR, which use $\psi_{GEN}$ as the first token, is increased by 6.1% when HDA is applied. Thus, HDA remains effective when the task boundaries are not clear. Although both

| Methods     | $\gamma = 0.01$ | $\gamma = 0.03$ | $\gamma = 0.05$ |
|-------------|-----------------|-----------------|-----------------|
|             | score ↑         | CPS-2 ↓         | score ↑         | CPS-2 ↓         | score ↑         | CPS-2 ↓         |
| LAMOL$_G$   | 71.9±0.14       | 1.061           | 73.6±0.39       | 1.043           | 78.9±0.34       | 0.92            |
| +HDA        | 76.1±0.91       | 0.894           | 79.3±0.73       | 0.673           | 80.4±0.25       | 0.582           |
| LAMOL$_T$   | 72.2±0.58       | 1.197           | 74.8±0.40       | 0.839           | 76.9±0.25       | 0.731           |
| +HDA        | 76.7±0.85       | 0.977           | 79.6±0.73       | 0.768           | 80.1±0.55       | 0.366           |
| L2KD        | 72.3±0.45       | 0.258           | 76.8±0.36       | 0.091           | 77.4±0.28       | 0.049           |
| +HDA        | 73.9±1.06       | 0.245           | 79.3±0.62       | 0.078           | 79.9±1.00       | 0.046           |
| DnR         | 71.9±0.36       | 1.239           | 78±0.17         | 1.489           | 78.1±0.17       | 1.006           |
| +HDA        | 78.0±0.20       | 1.098           | 80.8±0.50       | 0.764           | 81.0±0.38       | 0.761           |
| SD-EMD      | 77.5±0.57       | 0.984           | 80.6±0.75       | 0.93            | 81.3±0.40       | 0.589           |
| +HDA        | 79.9±0.33       | 0.736           | 81.4±0.26       | 0.492           | 82.4±0.11       | 0.309           |
L2KD and DnR are improved versions of LAMOL, there is almost no improvement when $\gamma = 0.01$. However, the performance of L2KD and DnR can be increased by 3.1% and 6.1%, respectively, when HDA is applied. When $\gamma = 0.03$, L2KD and DnR both perform better than LAMOL$_G$ and LAMOL$_T$ because they use distillation. HDA can bridge the gap caused by distillation. When $\gamma = 0.05$, the results are same as when $\gamma = 0.03$.

We propose CPS as a method for evaluating the weights of previous tasks in the pseudo-samples. The lower the CPS value is, the more evenly distributed the weight of each task in the pseudo-data. A good pseudo-dataset for incremental learning should be a homogeneous mixture that contains the task knowledge learned. Let CPS-n denote the CPS of the pseudo-data generated after the model has learned n tasks. CPS-1 is usually zero and thus does not need to be considered. As shown in Table 9, the CPS-2 of all baselines decreased when HDA was applied. In addition, all the baseline scores were significantly improved. This indicates that HDA can prevent the model from becoming overbiased to a new task. Therefore, catastrophic forgetting can be alleviated by applying HDA, which can result in higher quality pseudo-samples.

### 6 Conclusion and future work

In this paper, we propose data-free self-distillation (DFSD), a simple and efficient method that approaches the performance of multitask learning with fewer and noisier pseudo-samples. DFSD includes two components: (1) We apply an EMD-based self-distillation method to adaptively estimate the knowledge distribution of the tasks in all layers and transfer this data from the teacher model to the student model via adaptive self-multilayer-to-multilayer mapping. The experimental results indicate that the model with SD-EMD can achieve the same or better performance with fewer pseudo-samples. (2) HDA can reduce deviations between the pseudo-data and the real data by decomposing the generation process and bootstrapping. We propose the confusion of pseudo-samples (CPS) method to evaluate the quality of the pseudo-samples. In terms of the CPS, HDA reduces the bias of the model toward new tasks while retaining previously learned task knowledge. SD-EMD and HDA are complementary and can be applied in practical scenarios.

In future work, we plan to investigate incremental language learning under data-free constraints, focusing on one-shot or zero-shot pseudo-data replay and longer task sequences.

### Appendix A: Examples of Real Samples and Pseudo-samples

| Task | Real Sample | Good Pseudo Sample | Noisy Pseudo Sample |
|------|-------------|--------------------|--------------------|
| SQuAD | In 1882, Albert Zalm (John Zalm’s brother) built an early wind tunnel used to compare lift to drag of aeronautical models. Around 1899, Professor Jerome Green became the first American to send a wireless message. In 1931, Father Julius Nieuwland performed early work on basic reactions that was used to create neoprene... What did the brother of John Zalm construct at Notre Dame? | an early wind tunnel | |
| SQuAD | In 1882, Albert Zalm (John Zalm’s brother) built an early wind tunnel used to compare lift to drag of aeronautical models. Around 1899, Professor Jerome Green became the first American to send a wireless message. In 1931, Father Julius Nieuwland performed early work on basic reactions that was used to create neoprene... What did the brother of John Zalm construct at Notre Dame? | a new breed of fur | |
| SQuAD | but it may also prove to be a worthwhile endeavor. is this review negative or positive? | positive (similar to SST) | |
| SQuAD | but it may also prove to be a worthwhile endeavor. is this review negative or positive? | positive (similar to SST) | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
| SQuAD | the table has columns driver, grid, and so on... what was the grid when the grid was larger than 13? | the grid was larger than 13 | |
Table 10 (continued)

| Task | real sample | good pseudo sample | noisy pseudo sample (chaos) |
|------|-------------|-------------------|---------------------------|
| WikiSQL | _wikisql_ the table has columns player, no., nationality, position, years in toronto, school / club team and key words max, min, count, sum, avg, =, >, <, op, select, where, and, col, table, caption, page, section, op, cond, question, agg, aggops, condops - - what’s dell curry nationality? what is the translation from english to sql? | _wikisql_ the table has columns year, entrant, chassis, engine, points and key words max, min, count, sum, avg, =, >, <, op, select, where, and, col, table, caption, page, section, op, cond, question, agg, aggops, condops - - which chassis had a year smaller than 1991? what is the translation from english to sql? | (similar to SST) _wikisql_ this study examines the impact of gangs on friendships, families and workplace relationships ... is this review negative or positive? _ans_ positive |
Appendix B: Text Classification

For our DFSD method, we show the order of tasks and the corresponding training curves for the classification tasks shown in Fig. 8.

Fig. 8 The training curves for the text classification tasks. The graph plots the performance of the model in each epoch for each task.
Appendix C: Visualization of Self-Distillation Based on EMD

For the three decaNLP tasks, we show the IKDs of the hidden states and attention layer during incremental language learning for each task for all learning orders when the sampling ratio $\gamma = 0.05$ in Fig. 9.

Fig. 9  The IKDs of the three decaNLP tasks. We show the IKDs of the hidden state and attention layer during incremental language learning for each task for all learning orders when the sampling ratio $\gamma = 0.05$. 
References

1. Ring MB (1997) CHILD: A first step towards continual learning. Mach Learn 28(1):77–104. https://doi.org/10.1023/A:1007331723572
2. McCloskey M, Cohen NJ (1989) Catastrophic interference in connectionist networks: the sequential learning problem. In: Psychology of learning and motivation, vol 24, Elsevier, pp 109–165
3. French RM (1999) Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences 3(4):128–135
4. Polikar R, Upda L, Upda SS, Honavar VG (2001) Learn++: an incremental learning algorithm for supervised neural networks. IEEE Trans Syst Man Cybern Part C 31(4):497–508. https://doi.org/10.1109/5326.983933
5. Chen Z, Liu B (2018) Lifelong Machine Learning, Second Edition. Synthesis Lectures on Artificial Intelligence and Machine Learning Morgan & Claypool Publishers. https://doi.org/10.2503/cl-2018-0021
6. Li Z, Hoiem D (2017) Learning without forgetting. IEEE Trans Pattern Anal Mach Intell 40(12):2935–2947
7. Lopez-Paz D, Ranzato M (2017) Gradient episodic memory for continual learning. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp 6647–6647. https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html
8. Chaudhry A, Ranzato M, Rohrbach M, Elhoseiny M (2019) Efficient lifelong learning with A-GEM. In: 7th International conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net. https://openreview.net/forum?id=Hlk2c5FX
9. de Masson d’Autume C, Ruder S, Kong L, Yogatama D (2019) Episodic memory in lifelong language learning. In: Wallach HM, Larochelle H, Beygelzimer A, d’Alch´e-Buc F, Fox EB, Garnett R (eds) Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp 13122–13131. https://proceedings.neurips.cc/paper/2019/hash/8fd2e80c1458e250f198a2cfaa9b397-Abstract.html
10. Wang Z, Mehta SV, P´oczos B, Carbonell J (2020) Efficient meta lifelong-learning with limited memory. In: EMNLP
11. Sun FK, Ho CH, Lee HY (2019) Lamol: Language modeling for lifelong language learning. Proceedings of the ICLR 2020
12. Chang YS, Su SY, Chen YN (2020) Lifelong language knowledge distillation. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, Online, pp 2914–2924. https://doi.org/10.18653/v1/2020.emnlp-main.233
13. Sun J, Wang S, Zhang J, Zong C (2020) Distill and replay for continual language learning. In: Proceedings of the 28th international conference on computational linguistics. International Committee on Computational Linguistics, Barcelona, Spain (Online), pp 3569–3579. https://doi.org/10.18653/v1/2020.coling-main.318
14. Kanwchurcha K, Horsuwann T, Lertvitayakumjorn P, Kijjirubel K, Vateekul P (2021) Rational LAMoL: a rationale-based lifelong learning framework. In: Zong C, Xia F, Li W, Navigli R (eds) Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, ACL/ICJNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021. Association for Computational Linguistics, pp 2942–2953. https://doi.org/10.18653/v1/2021.acl-long.229
15. Li C, Li Y, Zhao Y, Peng P, Geng X (2021) SLER: Self-generated long-term experience replay for continual reinforcement learning. Appl Intell 51(1):185–201. https://doi.org/10.1007/s10489-020-01786-1
16. McCann B, Keskar NS, Xiong C, Socher R (2018) The natural language decathlon:. Multitask learning as question answering. arXiv:1806.08730
17. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I (2019) Language models are unsupervised multitask learners. OpenAI blog 18(9)
18. Furlanello T, Lipton Z, Tschannen M, Itti L, Anandkumar A (2018) Born again neural networks. In: International conference on machine learning, PMLR, pp 312–321
19. Rubner Y, Tomasi C, Guibas LJ (2000) The earth mover’s distance as a metric for image retrieval. Int J Comput Vis 40(2):99–121
20. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (2019) Continual lifelong learning with neural networks: a review. Neural Netw 115:53–71. https://doi.org/10.1016/j.neunet.2019.01.012
21. Heinrich S, Yao Y, Hinz T, Liu Z, Hummel T, Kerzel M, Weber C, Wermter S (2020) Crossmodal language grounding in an embodied neurocognitive model. Frontiers NeuroRobotics 14:52. https://doi.org/10.3389/fnроб.2020.00052
22. Capuano N, Greco L, Ritrovato P, Vento M (2021) Sentiment analysis for customer relationship management: an incremental learning approach. Appl Intell 51(6):3339–3352. https://doi.org/10.1007/s10489-020-01984-x
23. Cossu A, Carta A, Lomonaco V, Bacciu D (2021) Continual learning for recurrent neural networks: an empirical evaluation. Neural Netw 143:607–627. https://doi.org/10.1016/j.neunet.2021.07.021
24. Shin H, Lee JK, Kim J, Kim J (2017) Continual learning with deep generative replay. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp 2990–2999. https://proceedings.neurips.cc/paper/2017/hash/0ebe98067c6c73db1250d2eaa81-Abstract.html
25. Kemker R, Kanan C (2018) Fearnet: Brain-inspired model for incremental learning. In: 6th International conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net. https://openreview.net/forum?id=SJ1Xmf-Rb
26. Schwarz J, Czarnecki W, Luketina J, Grabska-Barwinska A, Teh YW, Pascual R, Hasdell R (2018) Progress & compress: A scalable framework for continual learning. In: Dy JG, Krause A (eds) Proceedings of the 35th international conference on machine learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, Proceedings of Machine Learning Research, vol 80, PMLR, pp 4535–4544. http://proceedings.mlr.press/v80/schwarz18a.html
27. Zhai M, Chen L, Tung F, He J, Nawah M, Mori G (2019) Lifelong gan: Continual learning for conditional image generation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2759–2768
28. van de Ven GM, Siegellmann HT, Tolias AS (2020) Brain-inspired replay for continual learning with artificial neural networks. Nat Commun 11. https://doi.org/10.1038/s41467-020-17866-2
29. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A et al (2017) Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences 114(13):3521–3526
31. Zenke F, Poole B, Ganguli S (2017) Continual learning through synaptic intelligence. In: Precup D, Teh YW (eds) Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, Proceedings of Machine Learning Research, vol 70, PMLR, pp 3987–3995. http://proceedings.mlr.press/v70/zenke17a.html

32. Aljundi R, Babiloni F, Elhoseiny M, Rohrbach M, Tuytelaars T (2018) Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 139–154

33. Lee S, Kim J, Jun J, Ha J, Zhang B (2017) Overcoming catastrophic forgetting by incremental moment matching. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp 4652–4662. https://proceedings.neurips.cc/paper/2017/hash/708f064faaf32a43e4d3784ef6af9e9-Abstract.html

34. nostalgebraist (2020) Interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed 31st Aug 2020

35. Alammar J (2021) Finding the words to say: Hidden state visualizations for language models. https://jalammar.github.io/hidden-states/

36. Li J, Liu X, Zhao H, Xu R, Yang M, Jin Y (2020) BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover’s distance. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics, Online, pp 3009–3018. https://doi.org/10.18653/v1/2020.emnlp-main.242, https://www.aclweb.org/anthology/2020.emnlp-main.242

37. Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, pp 4171–4186. https://doi.org/10.18653/v1/n19-1423

38. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2021) Understanding deep learning (still) requires rethinking generalization. Commun ACM 64:107–115

39. Reed SE, Lee H, Anguelov D, Szegedy C, Erhan D, Rabinovich A (2015) Training deep neural networks on noisy labels with bootstrapping. In: Bengio Y, LeCun Y (eds) 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings. http://arxiv.org/abs/1412.6596

40. Papineni K, Roukos S, Ward T, Zhu W (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, July 6-12, 2002, Philadelphia, PA, USA, ACL, pp 311–318. https://doi.org/10.3115/1073083.1073135, https://aclanthology.org/P02-1040/

41. Zhang X, Zhao JJ, LeCun Y (2015) Character-level convolutional networks for text classification. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems 28: annual conference on neural information processing systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pp 649–657. https://proceedings.neurips.cc/paper/2015/hash/250c8b51c773f308c8b4be067a9a02-Abstract.html

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Han Wang received the B.E. degree in Communication Engineering from Minzu University of China in 2016. He is currently a Ph.D. student at the Institute of Acoustics, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing, China. His research interests include incremental/lifelong/continual learning, dialogue system, and text classification.

Ruiliu Fu received the BSc degree in Electronic Engineering from Tsinghua University, China, 2017. He is currently a Ph.D. student in Institute of Acoustics, Chinese Academy of Sciences, China. His research interests include question answering system, interpretability in natural language understanding, event detection, and dialogue system.

Chengzhang Li received the B.S. degree in Department of Electronic Engineering from Tsinghua University, Beijing, China in 2019. He is currently working toward the Ph.D. degree in Signal and information processing with the University of Chinese Academy of Sciences, Beijing, China. His research interests include Knowledge Distillation and Chinese Spelling Check.
Xuejun Zhang received the B.S. degree in electronic communication engineering from North China Institute of Science and Technology, Hebei, China, in 2012 and the M.S. degree in communication engineering from Beijing Institute of Technology, Beijing, China, in 2015. She is currently pursuing the Ph.D. degree in natural language processing from Institute of Acoustics, Chinese Academy of Sciences (CAS) in 2012, Beijing, China. From 2015 to 2018, she was a Research Assistant with the Institute of Acoustics, Chinese Academy of Sciences (CAS), Beijing, China. She used to study cross-lingual semantic representation and conversational text classification. Her current research interest lies in spoken language understanding, dialog state tracking, and dialogue decision.

Jun Zhou is an associate researcher at the Institute of Acoustics, the Chinese Academy of Sciences. His research direction: big data computing, the question answering system.

Xing Bai is a Postdoc at Institute of Acoustics, Chinese Academy of Sciences. He received Ph.D. in signal and information processing from the University of Chinese Academy of Sciences in 2021, MSc in mathematics from the China University of Petroleum (Beijing) in 2017, and BSc in mathematics from Beijing Institute of Technology in 2014. His research interests include semantic image segmentation, deepfake detection and talking face generation.

Yonghong Yan received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 1990, and the Ph.D. degree in computer science and engineering from the Oregon Graduate Institute of Science and Technology, Hillsboro, OR, USA, in 1995. He is currently a Professor with the Speech Acoustics and Content Understanding Laboratory, Chinese Academy of Sciences, Beijing, China. His research interests include speech processing and recognition, language/speaker recognition, and human computer interface.

Qingwei Zhao received the B.E. degree and M.S. degree from Harbin Engineering University in 1992 and 1995 respectively. He received the Ph.D. degree in Signal and Information Processing from Tsinghua University in 1999. He is currently a professor at Institute of Acoustics, Chinese Academy of Sciences. His research interests include speech recognition, man-machine speech interaction and speech analysis, etc. He has been in charge of ten government research projects and several collaboration projects with enterprises, such as Baidu, Alibaba, China Mobile, China Telecom, Sony, etc. He has published more than 90 papers at international or domestic journals and conferences, and he has obtained 3 international PCT patents and 12 domestic patents. He has obtained one outstanding achievement award of science and technology from Chinese Academy of Sciences. He has also obtained three Science and Technology awards from Beijing City. And he has obtained two Division Recognition Awards from Intel Corporation.
Affiliations

Han Wang¹,² · Ruiliu Fu¹,² · Chengzhang Li³,² · Xuejun Zhang¹,² · Jun Zhou¹,² · Xing Bai¹,² · Yonghong Yan¹,² · Qingwei Zhao¹,²

Han Wang
wanghan@hccl.ioa.ac.cn

Ruiliu Fu
furuiliu@hccl.ioa.ac.cn

Chengzhang Li
lichengzhang@hccl.ioa.ac.cn

Xuejun Zhang
zhangxuejun@hccl.ioa.ac.cn

Jun Zhou
zhoujun@hccl.ioa.ac.cn

Xing Bai
baixing@hccl.ioa.ac.cn

Yonghong Yan
yanyonghong@hccl.ioa.ac.cn

¹ Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China

² University of Chinese Academy of Sciences, Beijing, China