Applied Intelligence (2023) 53:5607–5619
https://doi.org/10.1007/s10489-022-03779-8

Knowledge distilled pre-training model for vision-language-navigation

Bo Huang[1] · Shuai Zhang[1] · Jitao Huang[2] · Yijun Yu[1] · Zhicai Shi[3] · Yujie Xiong[1]

Accepted: 17 May 2022 / Published online: 30 June 2022

Abstract

Vision-language-navigation(VLN) is a challenging task that requires a robot to autonomously move to a destination based on visual observation following a human's natural language instructions. To improve the performance and generalization ability, the pre-training model based on the transformer is used instead of the traditional methods. However, the pre-training model is not suitable for sustainable computing and practical application because of its complex computations and large amount of hardware occupation. Therefore, we propose a slight pre-training model through knowledge distillation. Through knowledge distillation, the plenty of knowledge encoded in a large "teacher" model can be well transferred to a small "student" model, which greatly reduces the model parameters and inference time while maintaining the original performance. In the experiments, the model size is reduced by 87%, and the average inference time is reduced by approximately 86%. It can be trained and run much faster. At the same time, 95% performance of the original model was maintained, which is still better than the traditional VLN models.

Keywords Natural language processing · Computer vision · Cross-modality · Deep learning

1 Introduction

Learning to navigate in a visual environment following natural language instruction, the model should be trained to fuse textual and visual information. The specific VLN process [1] is shown in Fig. 1, which shows the global trajectory of the instruction, the local visual scene and the top view. The agent must finish the navigation in a house according to the step by step instruction.

Most traditional methods build on a Seq2Seq architecture, which encodes and decodes all the information through LSTM. In this way, each instruction is understood in isolation. The model learns from scratch without a priori in-domain knowledge. Additionally, the instructions corresponding to each trajectory in the VLN task describe that trajectory from a partial perspective, so using priori in-domain knowledge is necessary.

Therefore, we adopt the PREVALENT [2] model, which is a transformer-based pre-training model, as the base model. It improves the success rate (SR) and the success rate weighted by path length (SPL) in both seen and unseen environments compared to traditional models. Instead of using a trajectory (a string of points) as a training sample, the trajectory is split into several points; each point is a training sample in the form of a "text-image-action" triad, i.e., each training sample is the corresponding instruction, visual state, and action for that point. This model enables the use of a priori knowledge and does not allow each instruction to be understood independently.

However, it is too expensive to train and run because of the tremendous number of model parameters. The PREVALENT model has over 130 million parameters. Such a large model needs many powerful GPUs to calculate and costs too much energy. Obviously, it is not appropriate for sustainable computing and practical application in real life.

Therefore, inspired by Hinton [3], we propose knowledge distillation [4] to reduce the model parameters and shorten the inference time while maintaining the most performance. Knowledge distillation aims to transfer the knowledge embedded in a large teacher model to a small student model. In this paper, PREVALENT is the large teacher model. The small student model is trained to follow the behaviours of the teacher model. To transfer the knowledge, four loss functions are designed to fit the teacher model's

✉ Bo Huang
huangbosues@sues.edu.cn

Extended author information available on the last page of the article.

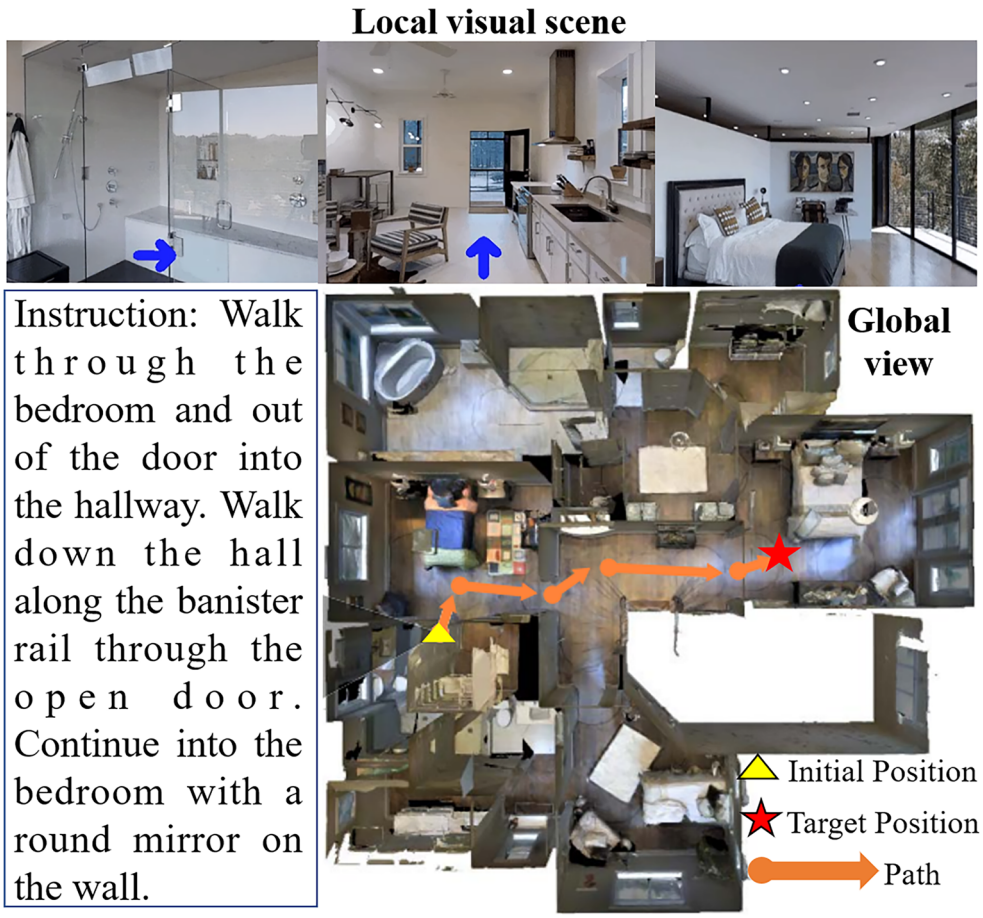


Fig. 1 VLN task demonstration. The left side is the instruction. Above and below are the local visual scene and global view

representation according to the the transformer core structure: 1) the embedding output, 2) the attention mechanism hidden state, 3) the feedforward layer hidden state, 4) the logits output of the last prediction layer. After the experiments, the results show that the student model transfers approximately 95% of the knowledge from the teacher model.The model size is reduced by 87%, and the average inference time is reduced by approximately 86%.It maintains a good cross-modality semantic representation ability and runs much faster.

2 Background

VLN tasks involve multimodal data, and one of its greatest challenges is to require the agent to take "appropriate" actions in an environment that has never been seen before. An agent determines a trajectory $T = (s_1, s_2 \ldots s_m)$ that consists of m viewpoints based on visual observation V, following the natural language instruction $X = (x_1, x_2 \ldots x_n)$, which consists of n words. At each step t, the agent obtains visual observation $v_t \in V$ and navigable viewpoint set $\{l_{t,k}\}_{k=1}^{N_t}$. The visual observation is a panorama $\{o_{t,i}\}_{i=1}^{36}$ concatenated by 36 RGB images. Each image $o_{t,i}$ represents a different orientation $(\theta_{t,i}, \phi_{t,i})$,where $\theta_{t,i} \in [-\pi, +\pi]$ is the heading pose, and $\phi_{t,i} \in \left[-\frac{\pi}{2}, +\frac{\pi}{2}\right]$ is the elevation pose. The navigable viewpoint set $\{l_{t,k}\}_{k=1}^{N_t}$ indicates that at step t there are N_t viewpoints that the agent can go next.$l_{t,k}$ contains the relative orientation $(\hat{\theta}_{t,i}, \hat{\phi}_{t,i})$ between the current point and the next navigable point. The agent needs to take an action a_t according to instruction X,observation V and historical actions $\{a_\tau\}_{\tau=1}^{t-1}$.

3 Related work

3.1 Vision-language-navigation

At present, there are generally three VLN tasks, R2R, CVDN [5] and HANNA [6]. The R2R task is the current mainstream research task with clear natural language instructions. The CVDN task allows robots to navigate autonomously in conversation. However, no specific intermediate process is specified in the HANNA task.

The current methods for VLN tasks are as follows. In Speaker-Follower [7], a panoramic image was used as the image state for the first time, and the dataset was augmented by backtranslation [8] technology. EnvDrop

[9] further uses dropouts in the environment to reach SOTA. RCM [10] processes attention in many ways, predicts the next action through LSTM, and uses imitation learning and reinforcement learning to improve accuracy. In R2R environments, the agent's ability to perceive real-world images and understand natural languages becomes even more crucial. PTA [11] designed a transformer-like architecture that combines action history and perception patterns and achieved good results in low-level VLN on R2R. MEPPR [12] uses BERT to evaluate tweet preprocessing to avoid noise and use hidden information. The most relevant work is PRESS [13], which uses a partial pre-training approach to encode the text directly using BERT. However, in this paper, we completely use a pre-training approach to pre-train the model for image and language information. In addition, most of the policy approaches tend to expose biases, and random actions can cause the intelligence to deviate from the correct path and invalidate the original instructions. To improve the success rate, both beam search and pre-exploration are widely used.

3.2 Vision language pre-training based on BERT

After the transformer-based pre-training model achieved great success in natural language processing, it was also been increasingly used in the cross-modality field. A batch of improved visual-language representation models based on BERT was proposed. The pre-training methods can improve not only performance but also generality. They can be applied to multiple different tasks with only fine-tuning,such as image captioning [14], visual question answering (VQA) and visual reasoning. For example, LXMERT [15] and ViLBERT [16] use single-stream and dual-stream methods to combine cross-modal attention to directly achieve the best performance on these tasks. VideoBERT [17] and VLBERT [18] fully demonstrate the powerful representation capabilities of the pre-trained model. In this paper, we use an approach similar to LXMERT method in Section 4, which is designed to significantly reduce memory usage, allowing the entire model to be trained on a single GPU without performance degradation.

3.3 Model compression

There are several model compression methods for reducing model size, training time and inference time, such as quantization [19], weight pruning [20], filter pruning [21] and knowledge distillation. ALBERT [22] reduced the model size by embedding factorization and parameter sharing. Since ALBERT does not reduce the hidden size or transformer block level, it still requires many computations. LFPC [23] learns and optimizes the validation loss of the pruning network obtained from the sampled criteria and can adaptively select the appropriate pruning criteria for different functional layers. Reformer [24] introduced the local sensitive hashing and RevNet's structure to reduce the attention complexity. The performance is comparable to that of the transformer model while being more memory efficient and faster on long sequences. mBERT [25] demonstrated the effectiveness of cross-linguistic transfer learning by using four different training strategies. ResNeXt [26] migrates the higher-level network knowledge to the lower-level network through self-distillation, which is more efficient for training than traditional knowledge distillation. AMTML-KD [27] can enable a student model to learn multiple knowledge levels from multiple teachers. DistillBert [28] refines knowledge through soft and hard logarithmic label loss with better language comprehension and generalization performance. We systematically investigate the mechanisms behind the above knowledge extraction and analyse how these effects help the training of student models. Our model differs from the above in two main ways. 1) The difference in application areas, for example, Zhang et al. used the ResNeXt model, and the task was image classification, while we applied the model to the VLN task. 2) The present model is more similar to DistillBert and TinyBERT [29] in that the knowledge is extracted from the model in the pre-training and fine-tuning phases through the transformer distillation method. However, we optimize the distillation performance by reducing the number of training steps.

4 Model

As Fig. 2 shows, The upper part is the teacher model that contains 3 single modality layers and 9 cross-modality layers. The lower part is the student model that contains 1 single modality layer and 3 cross-modality layers. By knowledge distillation, the large teacher model is compressed to the small student model with fewer layers. Figure 3 shows the detailed teacher model structure, which consists of the embedding, single modality encoder, and cross-modality encoder. The student model structure is the same as the teacher structure, but its number of encoder layers and its hidden state size are smaller.

4.1 Embedding

Embedding aims to turn the image and text input into a feature sequence. It consists of visual embedding and textual embedding.

4.1.1 Visual embedding

Different from PREVALENT, which directly uses CNN to extract features from the entire panoramic image, we extract

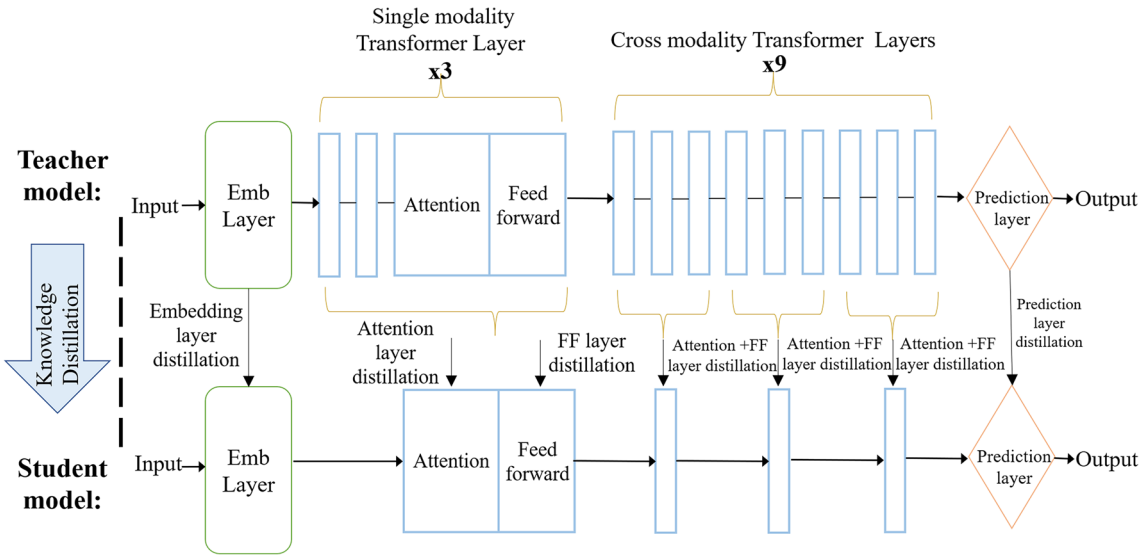


Fig. 2 Overview diagram of model distillation

object-level highlight features, which is called the region of interest feature (ROI). The Faster-RCNN is used to extract the objects in the image. The 2048-dimensional pooling feature before the final classification layer is used as the ROI feature. Each ROI feature's position embedding is the four-dimensional coordinates of its bounding box, namely,(x, y, h, w), x,y as the centre point coordinates, and h,w as height and width. Then, after passing through two fully connected layers (FCs), they are added to obtain the final image embedding. The specific formulas are as follows:

$$\hat{f}_j = \text{LayerNorm}\left(W_f f_j + b_f\right) \tag{1}$$

$$\hat{p}_j = \text{LayerNorm}\left(W_p p_j + b_p\right) \tag{2}$$

$$F_{\text{visualEmb}} = \left(\hat{f}_j + \hat{p}_j\right)/2 \tag{3}$$

where W_f, W_p and b_f, b_p are the weights and bias of a set of linear layers, respectively, and LayerNorm(LN) means layer normalization. $F_{\text{visualEmb}}$ is the final visual embedding.

4.1.2 Textual embedding

The textual embedding is similar to BERT. First, the input sentence is tokenized through WordPiece. Special tokens([CLS], [SEP]) are added at the beginning and end of the input embedding. Then, the one-hot word vector and its position are embedded separately. The final textual embedding is the sum of word embedding and position embedding. The word embedding represents the mapping from the vocabulary to the hidden state. The specific formulas are as follows:

$$\hat{x}_i = \text{WordEmb}\left(x_i\right) \tag{4}$$

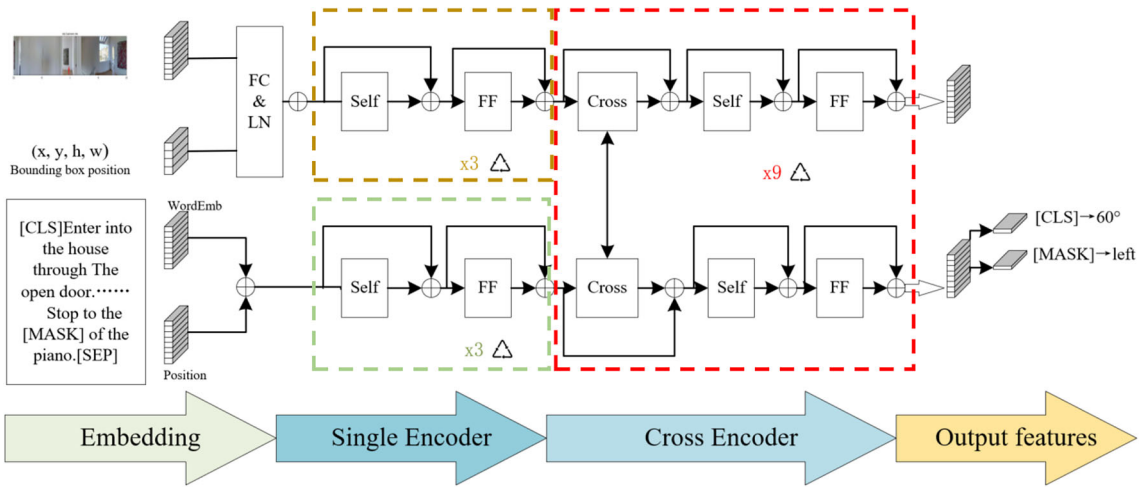


Fig. 3 The detailed model structure

$$\hat{u}_i = IdxEmb\left(u_i\right) \tag{5}$$

$$F_{\text{textEmb}} = \text{LayerNorm}\left(\hat{x}_i + \hat{u}_i\right) \tag{6}$$

4.2 Single modality encoder

Single modality encoders process visual information and textual information separately. The single modality encoder is a multilayer normal transformer encoder. As shown in Fig. 3, the transformer encoder consists of self-attention and a feedforward layer.

Self-attention aims to retrieve information from a set of key-value pairs related to a query vector and only needs to address its own information to update the training parameters without adding additional information. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values. The formula is as follows:

$$\text{Attention}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

The feedforward layer is a fully connected linear layer with ReLU activation. Moreover, the residual structure is applied to both the self-attention and feedforward layers.

4.3 Cross-modality encoder

As shown in Fig. 3, the cross-modality encoder is a modified multilayer transformer encoder. To fuse visual and textual information, cross-modality attention is added in front of the normal transformer encoder. In the process of cross attention, the queries that come from the outputs of single modality encoders from different modalities are exchanged with each other. The scaled dot products of visual queries with textual keys are computed to obtain textual features that contain visual information. In the same way, the scaled dot products of textual queries with visual keys are computed to obtain visual features that include textual information. The formulas are as follows:

$$\text{CrossAtt}_{L\to V}\left(Q_L, K_V, V_V\right) = \text{softmax}\left(\frac{Q_L K_V^T}{\sqrt{d_k}}\right)V_V \tag{8}$$

$$\text{CrossAtt}_{V\to L}\left(Q_V, K_L, V_L\right) = \text{softmax}\left(\frac{Q_V K_L^T}{\sqrt{d_k}}\right)V_L \tag{9}$$

where the subscripts V and L indicate vision and language, respectively.

4.4 Student model

As shown in Fig. 2, the student model structure is the same as the teacher model. However, it is much smaller than the teacher model. The student's layer number of the cross-modality encoder is 3, while that of the teachers is 9. The hidden state size(as well as embedding size) of the student model is 312 dimensions, while that of the teachers is 768. The student model size is only one-fifth of the teacher model. The detailed model size comparison is shown in Table 1. The detailed student model distillation process is presented in Section 4.

5 Knowledge distillation

Knowledge distillation (KD) aims to transfer knowledge from a well-learned large teacher model. After all the training and KD, only the small student model needs to be used, which is faster and as good as the large model. In this section, four KD parts are designed for the student model to fit the teacher model's behaviours.

5.1 Embedding layer distillation

According to the structure of the teacher model, and because of the student model's smaller embedding size, the following mean square error loss function(MSE) forces the student model to obtain embedding representation transference. The loss is as follows:

$$l_{emb} = \frac{1}{M}\sum_{i=1}^{M}\left(E^s W_e - E^t\right)^2 \tag{10}$$

where $E^s \in \mathbb{R}^{l\times d'}$ and $E^t \in \mathbb{R}^{l\times d}$ are the embedding matrix sizes of the student and teacher models, respectively.l is the input sequence length.d is the teacher model size 768. d' is the student model size 312. Because of the different model sizes,$W_e \in \mathbb{R}^{d'\times d}$ is the learnable scaling matrix.

5.2 Attention layer distillation

Inspired by Ganeshz [30] and Guarasci [31], which found that the attention weights learned by BERT capture rich linguistic knowledge and that knowledge can be embedded in the structural layer of the BERT model, we propose attention distillation to force the fused cross-modality semantic knowledge to be transferred from the teacher model to the student model. Specifically, the student learns to fit the matrix of multihead attention in the teacher network, and the loss(MSE) is defined as:

$$l_{att} = \frac{1}{h}\frac{1}{M}\sum_{j=1}^{h}\sum_{i=1}^{M}\left(A_j^s - A_j^t\right)^2 \tag{11}$$

where h is the attention head number and $A_j^S \in \mathbb{R}^{l\times l}$ is the i-th head attention matrix of the student.

5.3 Feedforward layer distillation

The feedforward layer is another part of the transformer beyond the attention mechanism. The feedforward layer size of the student model is also smaller than that of the teacher. Therefore, we distil the output (the hidden state) of this layer for full knowledge transference. The loss(MSE) is defined as:

$$l_{ff} = \frac{1}{M}\sum_{i=1}^{M}\left(H^s W_h - H^t\right)^2 \tag{12}$$

where $H^s \in \mathbb{R}^{l\times d'}$, $H^t \in \mathbb{R}^{l\times d}$ and $W_h \in \mathbb{R}^{d'\times d}$ are similar to the embedding distillation.

5.4 Prediction layer distillation

To fit the output distribution of the teacher model, prediction layer distillation is designed as in Hinton's approach. Specifically, the soft cross-entropy loss is used:

$$l_{pre} = -\,\text{Softmax}\left(z^t\right)\cdot\log\left(\text{Softmax}\left(\frac{z^s}{T}\right)\right) \tag{13}$$

where z^t and z^s are the logits (final prediction output) of the teacher and student models, respectively, and T is the temperature value of log-likelihood loss. Here T is 1 in our setting.

Compared with TinyBERT, the distillation method in this section has a certain optimization. We merge the redundant steps into one step. By simply optimizing all losses at once, as described in the equation. This simplification not only reduces training time but also optimizes performance. Finally, the loss of the whole knowledge distillation is the sum of the weights above four parts distillation losses. The weights of each part are hyperparameters.

$$L = l_{emb} + l_{att} + l_{ff} + l_{pre} \tag{14}$$

6 Training

First, we train the teacher model following Sections 6.1 and 6.2. Then, the student model is trained through knowledge distillation (Section 5) to imitate the fine-tuned large teacher model. The pre-training process uses the public website's Room-to-Room dataset, which we trained ourselves from scratch.

6.1 Pre-training

The pre-training stage contains two tasks: image-attended masked language modelling (MLM) and action prediction (AP).

6.1.1 Masked language modelling

For text input, 15% of words are randomly replaced. For these 15% replaced words, 80% of the words were replaced with the special token [MASK], 10% of the words were replaced with other random words, and the remaining 10% of the words remained unchanged. The goal is to predict these masked words x_i based on the remaining words X_i and visual information v by minimizing the negative log-likelihood:

$$l_{MLM} = -\mathbb{E}\log\left(p\left(x_i \mid \boldsymbol{X}_i, v\right)\right) \tag{15}$$

The reasons for this masking strategy are as follows: replacing the original word with [MASK] can integrate true bidirectional semantics without revealing the label; replacing words randomly can force the model to learn the global semantics; keeping 10% of the words unchanged can give the model a certain degree of bias.

6.1.2 Action prediction

This task aims to predict the action a′. The special token([CLS]) is further processed. This token is the fused representation of two modalities' information. The output feature of this token is passed through an activated linear layer and normalized layer. Then, it is passed through another fully connected layer to obtain the turning angle value. This turning angle represents the specific angle that should be turned to reach the next navigable point according to visual observation and language instruction. This task uses the mean square error loss function, as follows:

$$l_{AP} = \mathbb{E}\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{a} - p\left(a' \mid x_{[\text{CLS}]}, v\right)\right)^2 \tag{16}$$

Finally, the full pre-training loss function is:

$$\text{Loss} = l_{MLM} + l_{AP} \tag{17}$$

Table 1 The model size comparison between the teacher and student model

Model	Layers	Hidden State Size	Feedforward Size	Model Size
Teacher	12	768	3,072	130 M
Student	4	312	1,200	17.3 M (87%↓)

6.2 Fine-tuning

After pre-training, the model should be fine-tuned further. The models based on the pre-training method can be applied to different downstream tasks via fine-tuning. PREVALENT has proven that the model can be used for 3 different tasks: VLN, CVDN, and HANNA. This paper only studies the most representative VLN task. To use the pre-trained model for fine-tuning in the VLN task, the output features that are attended and contextualized are fed into the EnvDrop model.

7 Experimental results & analysis

7.1 Dataset

The Room-to-Room(R2R) dataset for the VLN task is based on photorealistic home environments. There are 90 different scenes that contain 7,189 trajectories, and each trajectory contains approximately 5 viewpoints. Each trajectory corresponds to 3 different natural language instructions. The dataset is split into a training set, a seen validation set, an unseen validation set and a test set. Among them, the original dataset is in the form of three instructions for each path (several viewpoints). In this article, the dataset is separated into the form of each viewpoint corresponding to one instruction and the turning angle from the current point to the next point is calculated as an action. Finally, it forms a viewpoint-instruction-action triplet to apply to our model, for a total of 104,000 triplets.

7.2 Evaluation

The evaluation indicators used in this article are success rate(SR), navigation error(NE) and success weighted by path length (SPL).

- SR: The percentage of the agent's final location that is less than 3 metres away from the target location.
- NE: The mean distant error between the ground-truth viewpoint and the point to which the agent moves. The lower the result is, the better.
- SPL: A higher score represents more navigation efficiency.

Among these indicators, SPL is the recommended main indicator, and other indicators are considered auxiliary indicators.

7.3 Setting

The batch size is 96. The optimizer is AdamW. The learning rate is 5×10^{-5}. The word vocabulary is WordPiece(30255). We use $g(m) = 3 \times m$ as the layer mapping function, so that the student model learns every 3 layers for the teacher model. For the teacher model, a 3-layer single modality encoder for text and image, and a 9-layer cross-modality encoder are used. For the student model, a 1-layer single modality encoder for text and image, and a 3-layer cross-modality encoder are used. The hyperparameter temperature T is fixed to 1, $\alpha = 0.2$, and the learning weight λ is set to 1 for each layer, which works well for model learning.

7.4 Baseline models

We compare our approach with six recently published models.

a) Random:An agent that randomly chooses a direction and moves five steps in that direction.
b) Seq2Seq: The model based on the sequence-to-sequence model using a limited discrete action space.
c) RCM: The model combines model-free and model-based reinforcement learning.
d) EnvDrop: An agent is trained with environment-dropout, which can generate more environments base-don the limited seen environments.
e) Speaker-Follower: The model is trained with data augmentation from a speaker model on the panoramic action space.
f) PRESS: The model directly uses BERT for word embedding and random sampling of agents to train them to generalize well in unseen environments.

7.5 Results comparison with baselines

The results of our model in every part of the dataset have overall improvement compared to other baseline models due to adopting the transformer-based pre-training model. As shown in Table 2 and Fig. 4, the SPL is increased by 3.1 and 2.4, respectively, on Val Unseen and Test compared to the best baseline model. The SR also increases by 2% in the test set, which is an unseen environment. This means that our pre-training method has a better generalization ability. The higher SPL of our model shows that it can better understand the instructions and visual information.

7.6 Efficiency comparison

This part compares the efficiency of the teacher model (PREVALENT) and the student model (ours). The knowledge distillation improvement makes the model much more efficient. As shown in Table 3, Figs. 5 and 6, the model parameters of the student model(ours) are reduced by approximately 87% compared with the teacher model, which is also the PREVALENT model. Additionally, the

Table 2 The performance comparison with baselines

Model	Val Seen			Val Unseen			Test		
	NE(m)	SR(%)	SPL	NE(m)	SR(%)	SPL	NE(m)	SR(%)	SPL
Random	9.45	16	–	9.23	16	–	9.77	13	12
Seq2Seq	6.01	39	–	7.81	22	–	7.85	20	18
RCM	3.53	67	–	6.09	43	–	6.12	43	38
EnvDrop	3.99	62	59	5.22	52	48	5.23	50	47
Speaker-Follower	3.36	66	–	6.62	35	–	6.62	35	28
PRESS	4.39	58	55	5.28	49	45	5.49	49	45
Ours	3.86	69	64	4.98	44.7	51.1	5.53	52	49.4

inference time is greatly reduced by 86%, so our model is more suitable for practical applications and meets the need for sustainable computing. Although the model is increasingly smaller and faster, most knowledge of the teacher model is maintained. Only approximately 5% of the average performance is lost.

7.7 Effect of knowledge distillation

In this part, the effects and impact of each layer distillation are experimented.The student models were tested for their ability to navigate autonomously without embedding layer distillation (no embedding), attention layer distillation (no attention), feedforward layer distillation (no FF) or prediction layer distillation (no prediction), respectively. Table 4 and Fig. 7 show that all 4 parts of distillation are useful. Compared to the complete distillation (Ours) model, all metrics showed varying degrees of degradation. The attention layer distillation performance drops the most significantly and lost the most knowledge, which means that it is the key to student learning. This is followed by feedforward layer distillation, while attention layer

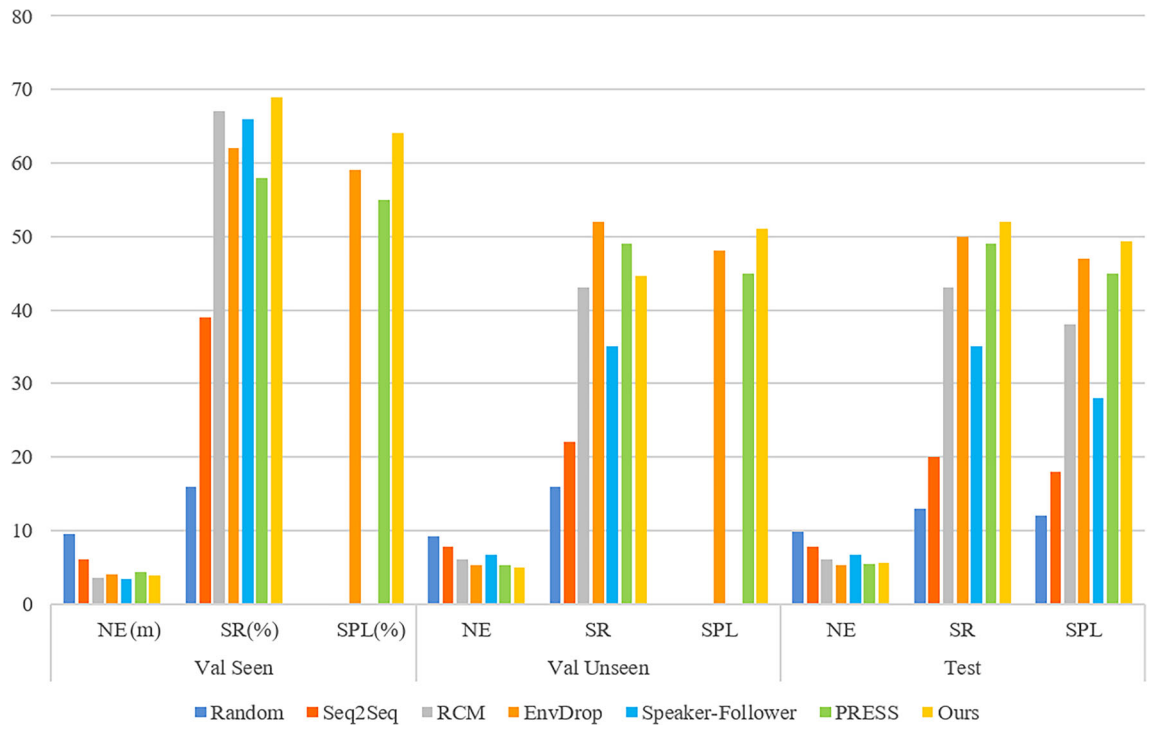


Fig. 4 The performance comparison with baselines

Table 3 The efficiency comparison

Model	Val Seen			Val Unseen			Test			Model Size	Inference time(s)
	NE	SR	SPL	NE	SR	SPL	NE	SR	SPL		
Teacher (PREVALENT)	3.67	69	65	4.71	58	53	5.30	54	51	130 M	7.9
Student (Ours)	3.86	69	64	4.98	44.7	51.1	5.53	52	49.4	17.3 M (87%↓)	1.1 (86.01%↓)

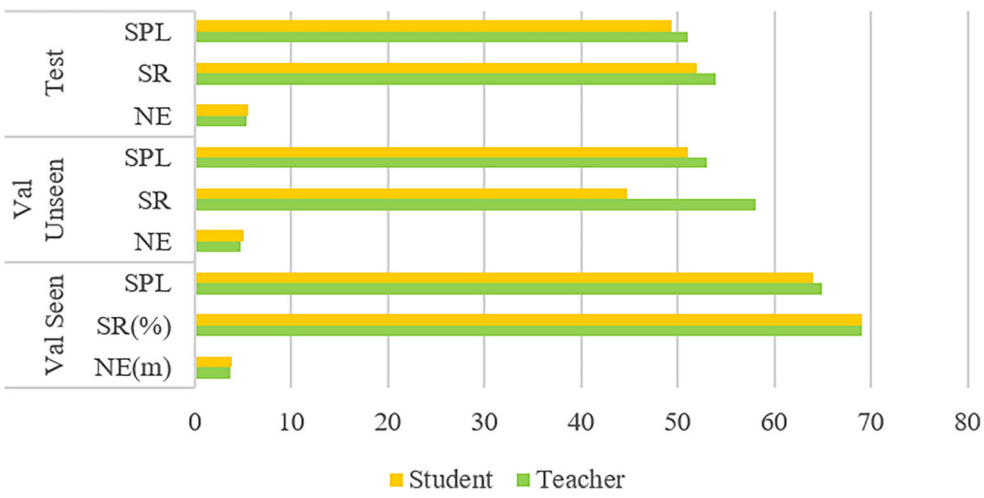


Fig. 5 Performance comparison between the teacher model and student model

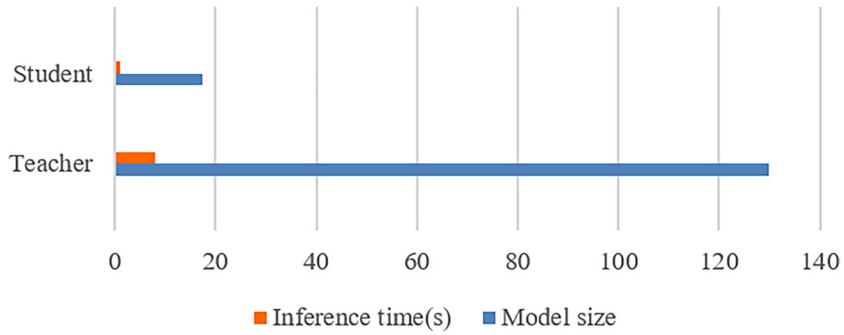


Fig. 6 Efficiency comparison between the teacher model and student model

Table 4 The effect of each part of knowledge distillation

Model	Val Seen			Val Unseen			Test		
	NE	SR	SPL	NE	SR	SPL	NE	SR	SPL
Ours	3.86	69	64	4.98	44.7	51.1	5.53	52	49.4
No Embedding	3.92	69	63	5.06	43.9	50.4	5.62	51.1	48.6
No Attention	4.53	62	55	5.98	36	40	6.11	40.1	38
No FF	3.99	65	59	5.14	40.1	45.2	5.72	45.3	41.3
No Prediction	3.95	67	62	5.01	43.63	52.8	5.66	50.8	48.2

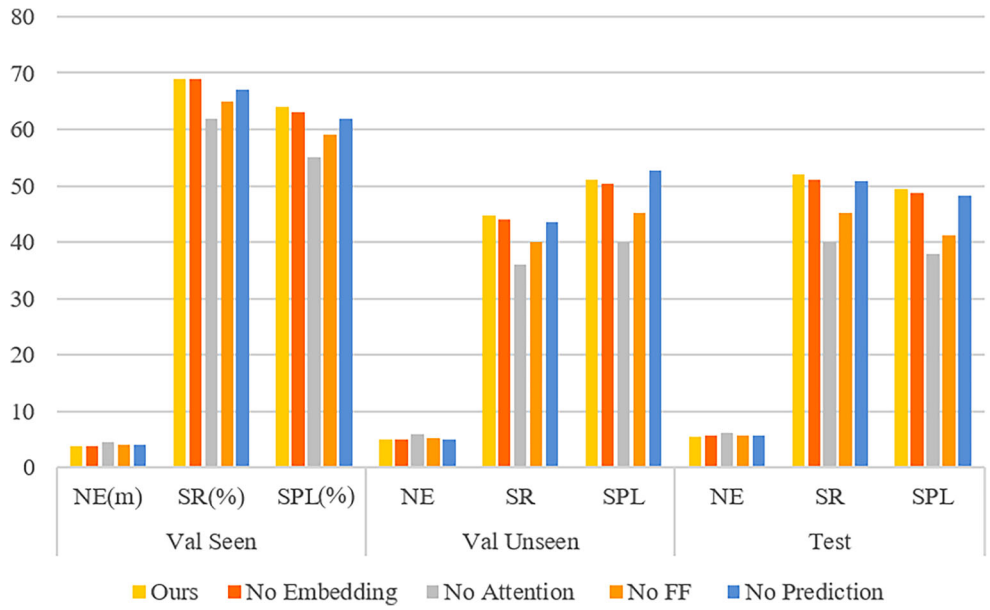


Fig. 7 The effect of each part of knowledge distillation

distillation and feedforward layer knowledge distillation are complementary to each other. The effect of embedding layer distillation is the smallest. To further examine the performance of the proposed knowledge distillation method. Compared with TinyBERT, its performance is better than TinyBERT under the same configuration. It can save about 21%-31% of training time. This is very important in the training of large datasets, this improvement not only reduces the training time, but also further improves the performance, which allows the student model to learn the prediction and intermediate layers more efficiently. In addition, Compared with other compression models, such as quantization, pruning and other methods, because the number of model parameters is different. So this is an unfair comparison and we will investigate this part further in future work. We will show the processing of the model in Appendix.

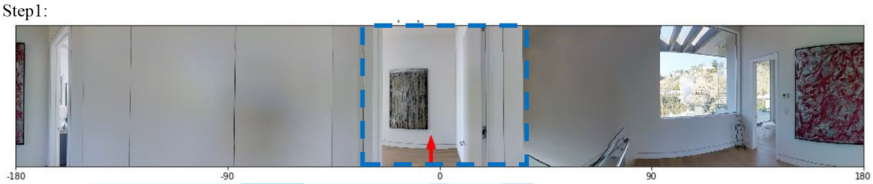
8 Conclusion

In this paper, a more efficient pre-training model for the VLN task is introduced. In contrast to the traditional LSTM-based autonomous navigation models, this paper uses a structure similar to the BERT language pre-training model. For the autonomous navigation task, object-level RoI features are embedded in the visual embedding phase, whereas in general autonomous navigation models only use Resnet-152 for overall-level visual embedding. The model stacks multiple layers using the transformer for cross-modal attention as an encoder for the fusion and characterisation of cross-modal features. The training method used is simpler and more suitable for autonomous navigation tasks. After complete training, the knowledge from the original large model is transferred to the small model in four ways using knowledge distillation. Experiments have proven to be more efficient and run faster while maintaining almost most of the performance.Our future work will focus on three areas: 1) how to improve the model's ability to generalise in the face of unknown environments, such as street navigation and navigation in continuous environments. 2) Currently our model is only used for VLN, we believe it has great potential for solving other tasks that require sequential interaction/decision making, such as verbal and visual dialogue, conversational navigation, etc. 3) We will further delve into how to effectively enable the model to learn from a broader and deeper teacher model, improve SR, and consider ways to improve the stability of the KD algorithm through methods such as self-supervised.

Appendix A: Visualization

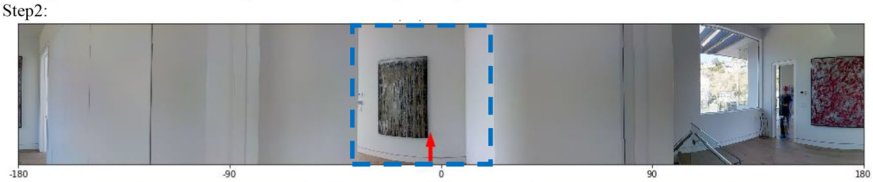
This part shows the process of our model step by step, especially the attention mechanism. The blue dotted box in the panorama denotes the visual information that needs to be given the most attention according to the textual information and historical trajectory. In addition, the red arrow is the specific orientation (the next point). The blurry part in the image needs to be ignored according to the attention mechanism. The coloured words (such as coloured words) in the instruction represent the degree of attention. The darker the colour, the more attention required.

Step1:



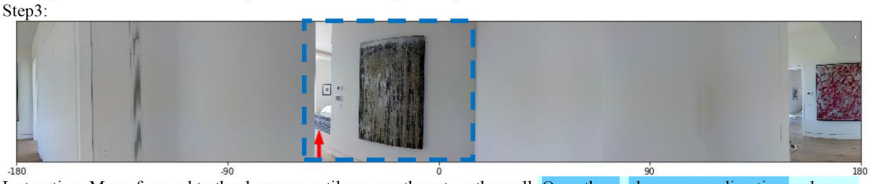
Instruction: Move forward to the doorway until you see the art on the wall. Once there, change your direction and move onward towards the left. Continue your forward progress through the bedroom. Past the footstool on your left. Continue moving forward out on the balcony and come to a complete stop behind the couch.

Step2:



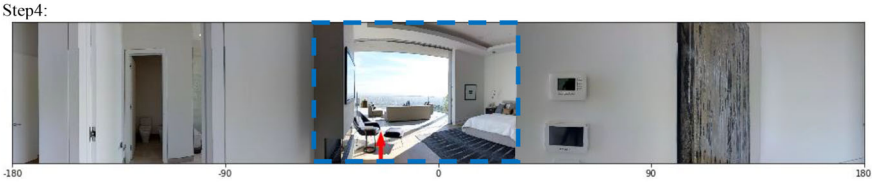
Instruction: Move forward to the doorway until you see the art on the wall. Once there, change your direction and move onward towards the left. Continue your forward progress through the bedroom. Past the footstool on your left. Continue moving forward out on the balcony and come to a complete stop behind the couch.

Step3:



Instruction: Move forward to the doorway until you see the art on the wall. Once there, change your direction and move onward towards the left. Continue your forward progress through the bedroom. Past the footstool on your left. Continue moving forward out on the balcony and come to a complete stop behind the couch.

Step4:



Instruction: Move forward to the doorway until you see the art on the wall. Once there, change your direction and move onward towards the left. Continue your forward progress through the bedroom. Past the footstool on your left. Continue moving forward out on the balcony and come to a complete stop behind the couch.

Acknowledgements This work is sponsored by the Scientific and Technological Innovation 2030 - Major Project of New Generation Artificial Intelligence (No. 2020AAA0109300), the Shanghai Science and Technology Young Talents Sailing Program (No. 19YF1418400), the National Natural Science Foundation of China (No. 62006150), the Shanghai Science and Technology Innovation Action Plan (22S31903700, 21S31904200), the Songjiang District Science and Technology Research Project (No.19SJKJGG83), and Shanghai Local Capacity Enhancement Project (No. 21010501500).

Declarations

Competing interests The authors declare that they have no conflicts of interest in this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

1. Anderson P, Qi W, Teney D, Bruce J, Johnson M, Sünderhauf N, Reid I, Gould S, Van Den Hengel A (2018) Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3674–3683
2. Hao W, Li C, Li X, Carin L, Gao J (2020) Towards learning a generic agent for vision-and-language navigation via pre-training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 13137–13146
3. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv:1503.02531
4. Wu MC, Chiu CT (2020) Multi-teacher knowledge distillation for compressed video action recognition based on deep learning. J Syst Archit 103:101695
5. Zhu Y, Zhu F, Zhan Z, Lin B, Jiao J, Chang X, Liang X (2020) Vision-dialog navigation by exploring cross-modal memory. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10730–10739
6. Nguyen K, Daumé III H (2019) Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. arXiv:1909.01871
7. Fried D, Hu R, Cirik V, Rohrbach A, Andreas J, Morency LP, Berg-Kirkpatrick T, Saenko K, Klein D, Trevor D (2018) Speaker-follower models for vision-and-language navigation. arXiv:1806.02724
8. Vapnik V, Izmailov R (2015) Learning using privileged information: similarity control and knowledge transfer. J Mach Learn Res 16(1):2023–2049
9. Tan H, Yu L, Bansal M (2019) Learning to navigate unseen environments:, Back translation with environmental dropout. arXiv:1904.04195

10. Wang X, Huang Q, Celikyilmaz A, Gao J, Shen D, Wang YF, Wang WY, Zhang L (2019) Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6629–6638
11. Landi F, Baraldi L, Cornia M, Corsini M, Cucchiara R (2021) Multimodal attention networks for low-level vision-and-language navigation. Comput Vis Image Underst 210:103255
12. Pota M, Ventura M, Fujita H, Esposito M (2021) Multilingual evaluation of pre-processing for bert-based sentiment analysis of tweets. Expert Syst Appl 181:115119
13. Li X, Li C, Xia Q, Bisk Y, Celikyilmaz A, Gao J, Smith N, Choi Y (2019) Robust navigation with language pretraining and stochastic sampling. arXiv:1909.02244
14. Kinghorn P, Li Z, Shao L (2018) A region-based image caption generator with refined descriptions. Neurocomputing 272:416–424
15. Cao J, Gan Z, Cheng Y, Yu L, Chen YC, Liu J (2020) Behind the scene: Revealing the secrets of pre-trained vision-and-language models. European Conference on Computer Vision
16. Lu J, Batra D, Parikh D, Lee S (2019) Vilbert:, Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv:1908.02265
17. Sun C, Myers A, Vondrick C, Murphy K, Schmid C (2019) Videobert: a joint model for video and language representation learning. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7464–7473
18. Liu G, Liao Y, Wang F, Zhang B, Zhang L, Liang X, Wan X, Li S, Li Z, Zhang S et al (2021) Medical-vlbert: Medical visual language bert for covid-19 ct report generation with alternate learning. IEEE Trans Neural Netw Learn Syst 32(9):3786–3797
19. Hubara I, Courbariaux M, Soudry D, El-Yaniv R, Bengio Y (2017) Quantized neural networks: Training neural networks with low precision weights and activations. J Mach Learn Res 18(1):6869–6898
20. Yeom SK, Seegerer P, Lapuschkin S, Binder A, Wiedemann S, Müller KR, Samek W (2021) Pruning by explaining: a novel criterion for deep neural network pruning. Pattern Recogn 115:107899
21. Wang X, Zheng Z, He Y, Yan F, Zeng Z, Yang Y (2021) Soft person reidentification network pruning via blockwise adjacent filter decaying. IEEE Transactions on Cybernetics
22. Chi PH, Chung PH, Wu TH, Hsieh CC, Chen YH, Li SW, Lee HY (2021) Audio albert: A lite bert for self-supervised learning of audio representation. 2021 IEEE Spoken Language Technology Workshop (SLT)
23. He Y, Ding Y, Liu P, Zhu L, Zhang H, Yi Y (2020) Learning filter pruning criteria for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2009–2018
24. Riaz N, Latif S, Latif R (2021) From transformers to reformers. 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)
25. Catelli R, Gargiulo F, Casola V, De Pietro G, Fujita H, Esposito M (2020) Crosslingual named entity recognition for clinical de-identification applied to a covid-19 italian data set. Appl Soft Comput 97:106779
26. Zhang L, Song J, Gao A, Chen J, Bao C, Ma K (2019) Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In: Proceedings of the IEEE/CVF international conference on computer Vision, pp 3713–3722
27. Liu Y, Zhang W, Wang J (2020) Adaptive multi-teacher multi-level knowledge distillation. Neurocomputing 415:106–113
28. Sanh V, Debut L, Chaumond J, Wolf T (2019) Distilbert, a distilled version of bert:, smaller, faster, cheaper and lighter. arXiv:1910.01108
29. Liu W, Zhao X, Zhao Z, Qi J, Yang X, Lu W (2021) An empirical study on adaptive inference for pretrained language model IEEE Transactions on Neural Networks and Learning Systems
30. Ganesh P, Chen Y, Lou X, Khan MA, Yang Y, Sajjad H, Nakov P, Chen D, Winslett M (2021) Compressing large-scale transformer-based models: a case study on bert. Trans Assoc Comput Linguist 9:1061–1080
31. Guarasci R, Silvestri S, De Pietro G, Fujita H, Esposito M (2021) Assessing bert's ability to learn italian syntax: a study on null-subject and agreement phenomena. Journal of Ambient Intelligence and Humanized Computing

Bo Huang is currently an associate professor in the School of Electronic and Electrical Engineering, Shanghai University of Engineering Science. His main research interests include software engineering, artificial intelligence, big data, and natural language processing.

Shuai Zhang is a postgraduate student of the School of Electrical and Electronic Engineering, Shanghai University of Engineering and Technology. His main research interests are natural language processing, sentiment analysis, and text classification.

Jitao Huang is an algorithm engineer from Chinatelecom, who received the master degree in control engineering from the School of Electrical and Electronic Engineering, Shanghai University of Engineering and Technology. His research interests include natural language processing, cross modality and computer vision.

Affiliations

Bo Huang[1] · Shuai Zhang[1] · Jitao Huang[2] · Yijun Yu[1] · Zhicai Shi[3] · Yujie Xiong[1]

Shuai Zhang
854400656@qq.com

Jitao Huang
421024976@qq.com

Yijun Yu
1210947362@qq.com

Zhicai Shi
szc1964@163.com

Yujie Xiong
xiong@sues.edu.cn

[1] School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai, China

[2] China Telecom Corporation Limited Shanghai Branch, Shanghai, China

[3] Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai, China