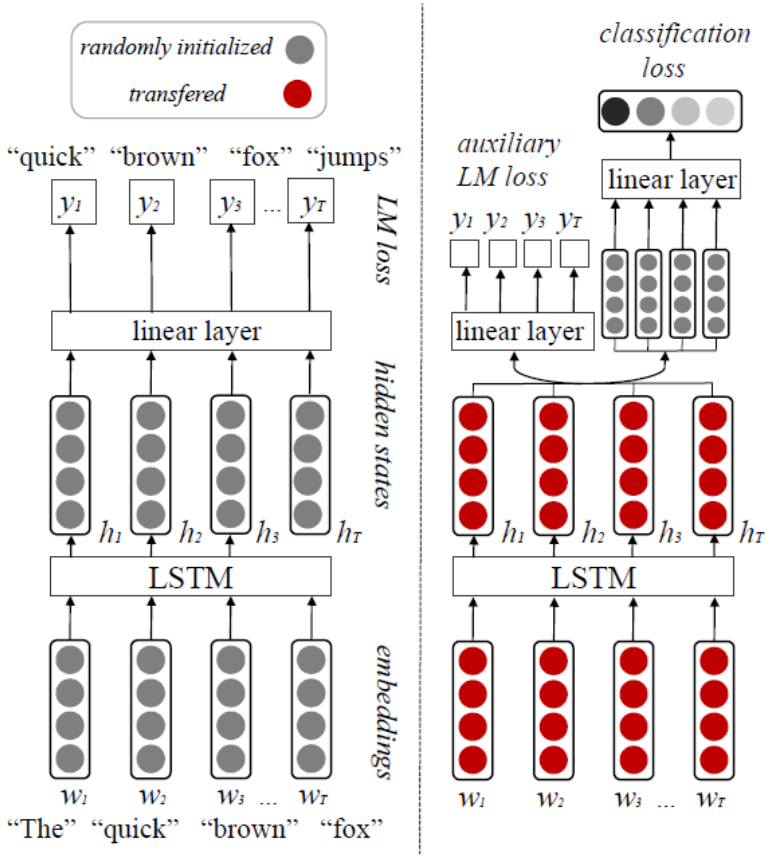
Transfer & Dialog

An Embarrassingly Simple Approach for Transfer Learning from Pretrained Language Models

Alexandra Chronopoulou[1], Christos Baziotis[1], Alexandros Potamianos[1,2]

[1]School of ECE, National Technical University of Athens, Athens, Greece
[2] Signal Analysis and Interpretation Laboratory (SAIL), USC, Los Angeles, USA



NAACL-2019 Short Paper

LM Pretraining.

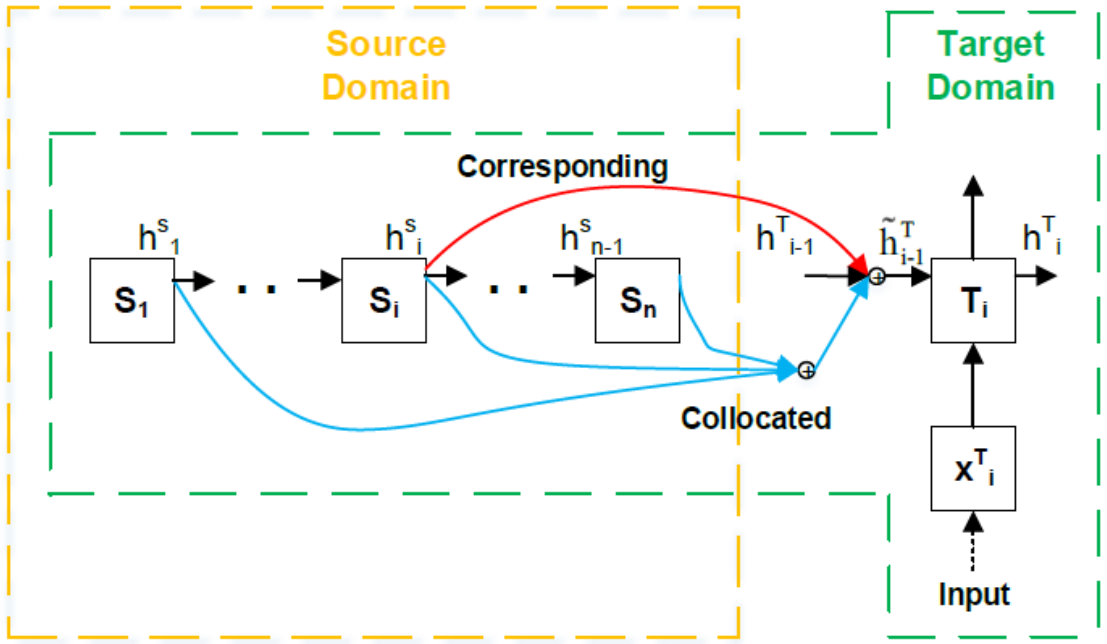
Transfer & auxiliary loss.

Exponential decay of γ.

Sequential Unfreezing.

TRANSFER LEARNING FOR SEQUENCES VIA LEARNING TO COLLOCATE

Wanyun Cui[§] **Guangyu Zheng**[‡] **Zhiqiang Shen**[¶] **Sihang Jiang**[‡] **Wei Wang**[‡]
cui.wanyun@sufe.edu.cn, {simonzheng96, zhiqiangshen0214, tedjiangfdu}@gmail.com
weiwang1@fudan.edu.cn
[§]Shanghai University of Finance and Economics
[‡]Shanghai Key Laboratory of Data Science, Fudan University
[¶]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University



ICLR-2019 Paper

- 之前的transfer往往是句子层级的，将source netword的句子表示迁移过来。
- 本文在词级别做迁移
- 并且增加attetion结构，捕获长距离依赖

Aligned Recurrent Transfer (ART)

$h_i^S = RNN(h_{i-1}^S, x_i^S; \theta_S)$ 基础的RNN

$h_i^T = RNN(\widetilde{h_{i-1}^T}, x_i^T; \theta_T)$ Hidden state改为考虑Source Domain

$\widetilde{h_{i-1}^T} = f(h_{i-1}^T, \psi_i | \theta_f)$ Didden state的计算方式

$\psi_i = (1 - u_i) \circ \pi_i + u_i \circ h_i^S$ 如何计算对Source Domain的Att

$u_i = \delta(W_u h_i^S + C_u \pi_i)$ 权重，Gate计算

$\pi_i = \sum_{j=1}^{n} \alpha_{ij} h_j^S$ 当前State对左右Source Hidden的Att

$\alpha_{ij} = \frac{\exp(a(h_{i-1}^T, h_j^S))}{\sum_{j'=1}^{n} \exp(a(h_{i-1}^T, h_{j'}^S))}$ Att中权重的计算方式

$a(h_i^T, h_j^S) = v_a^\top \tanh(W_a h_i^T + U_a h_j^S)$ 该相似度计算能降低复杂度

$f(h_i^T, \psi_i) = (1 - z_i) \circ h_{i-1}^T + z_i \circ \widetilde{\psi_i}$

如何整合Target的Hidden和Source Att

$\widetilde{\psi_i} = \tanh(W_\psi x_i + U_\psi [r_i \circ h_{i-1}^T] + C_\psi \psi_i)$

$z_i = \delta(W_z x_i + U_z h_{i-1}^T + C_z \psi_i)$

$r_i = \delta(W_r x_i + U_r h_{i-1}^T + C_r \psi_i)$

上述整合的权重计算，考虑

ART over LSTM

$$\begin{bmatrix} \widetilde{c_i^S} \\ o_t^S \\ i_t^S \\ f_t^S \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} T_{A,b}^S \begin{bmatrix} x_t^S \\ h_{t-1}^S \end{bmatrix}$$

$$c_t^S = \widetilde{c_t^S} \circ i_t^S + c_{t-1}^S \circ f_t^S$$

$$h_t^S = o_t^S \circ \tanh(c_t^S)$$

$$\begin{bmatrix} \widetilde{c_i^T} \\ o_t^T \\ i_t^T \\ f_t^T \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} T_{A,b}^T \begin{bmatrix} x_t^T \\ f(h_{i-1}^T, \psi_{hi} | \theta_{fh}) \end{bmatrix}$$

$$c_t^T = \widetilde{c_t^T} \circ i_t^T + f(c_{i-1}^T, \psi_{ci} | \theta_{fc}) \circ f_t^T$$

$$h_t^T = o_t^T \circ \tanh(c_t^T)$$

Results

Table 2: Classification accuracy on the Amazon review dataset.

Source	Target	LSTM	LSTM-u	LSTM-s	CCT	LWT	DANN	DAmSDA	AMN	HATN	ART
Books	DVD	0.695	0.770	0.718	0.730	0.784	0.725	0.755	0.818	0.813	**0.870**
Books	Elec.	0.733	0.805	0.678	0.768	0.763	0.690	0.760	0.820	0.790	**0.848**
Books	Kitchen	0.798	0.845	0.678	0.818	0.790	0.770	0.760	0.810	0.738	**0.863**
DVD	Books	0.745	0.788	0.730	0.800	0.778	0.745	0.775	0.825	0.798	**0.855**
DVD	Elec.	0.733	0.788	0.663	0.775	0.785	0.745	0.800	0.810	0.805	**0.845**
DVD	Kitchen	0.798	0.823	0.708	0.815	0.785	0.780	0.775	0.830	0.765	**0.853**
Elec.	Books	0.745	0.740	0.648	0.773	0.735	0.655	0.725	0.785	0.763	**0.868**
Elec.	DVD	0.695	0.753	0.648	0.768	0.723	0.720	0.695	0.780	0.788	**0.855**
Elec.	Kitchen	0.798	0.863	0.785	0.823	0.793	0.823	0.838	**0.893**	0.808	0.890
Kitchen	Books	0.745	0.760	0.653	0.803	0.755	0.645	0.755	0.798	0.740	**0.845**
Kitchen	DVD	0.695	0.758	0.678	0.750	0.748	0.715	0.775	0.805	0.738	**0.858**
Kitchen	Elec.	0.733	0.815	0.758	0.810	0.805	0.810	**0.870**	0.833	0.850	0.853
Average		0.763	0.792	0.695	0.803	0.774	0.735	0.774	0.817	0.783	**0.858**

Table 5: Performance over POS tagging and NER.

Task	Source	Target	HRN	FLORS	LSTM	CCT	ART
POS Tagging	PTB	Twitter/0.1	0.837	0.763	0.798	0.852	**0.859**
POS Tagging	PTB	Twitter/0.01	0.647		0.573	0.653	**0.658**
NER	CoNLL	Twitter/0.1	0.432	-	0.210	0.434	**0.450**
NER	Twitter	CoNLL/0.01	-	-	0.576	0.675	**0.707**

Analysis

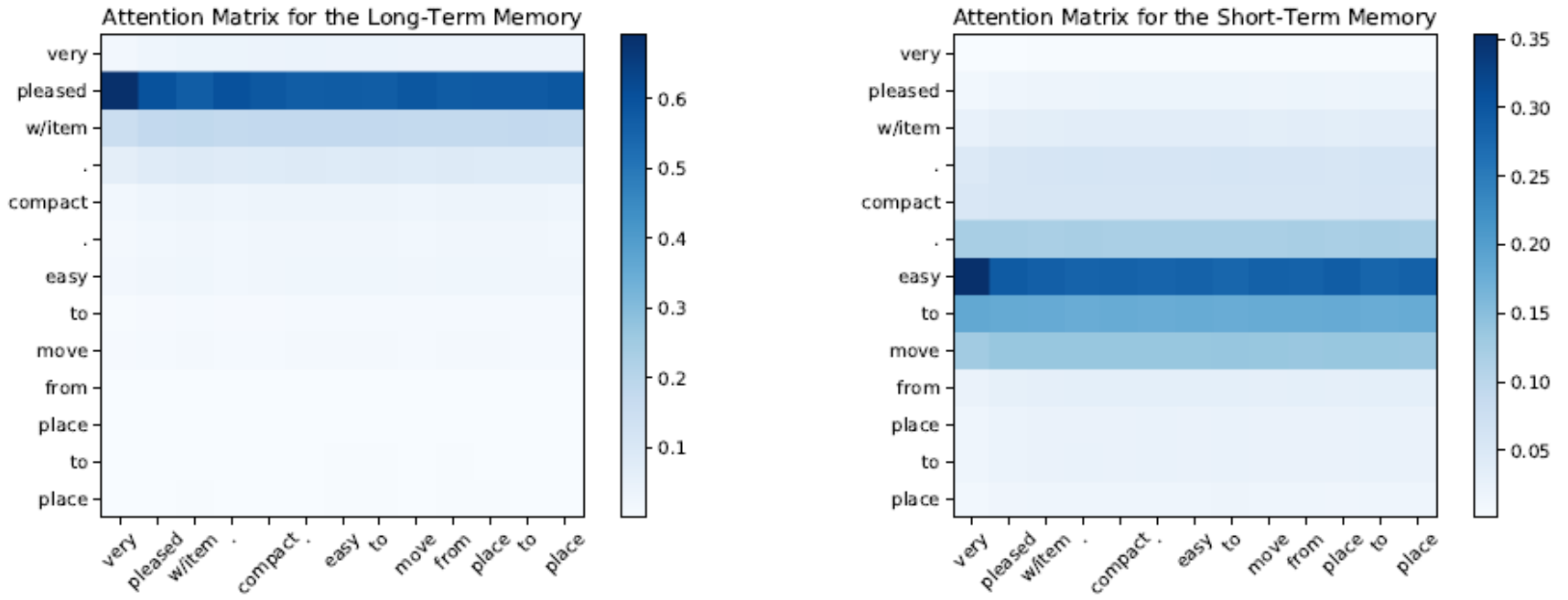


Figure 3: Attention matrix visualization. The x-axis and the y-axis denote positions in the target domain and source domain, respectively. Figure (a) shows the attention matrix for the long-term memory in the forward neural network. Figure (b) shows the attention matrix for the short-term memory in the forward neural network.

Questions?

Re-evaluating ADEM: A Deeper Look at Scoring Dialogue Responses

Ananya B. Sai[*†§], **Mithun Das Gupta**[‡], **Mitesh M. Khapra**[*†], **Mukundhan Srinivasan**[§]
[*]Department of Computer Science and Engineering, Indian Institute of Technology, Madras
[†]Robert Bosch Center for Data Sciences and AI (RBC-DSAI), Indian Institute of Technology, Madras
[‡]Microsoft, India
[§]NVIDIA, India

AAAI-2019 Paper

Recall: ADEM

$$score(c, r, \hat{r}) = (c^T M \hat{r} + r^T N \hat{r} - \alpha)/\beta$$

$$\mathcal{L} = \sum_{i=1:K} [score(c_i, r_i, \hat{r}_i) - human_i]^2 + \gamma||\theta||_2$$

Conicity

$$\text{ATM}(v, V) = v^T \bar{V}$$

$$\text{Conicity}(V) = \frac{1}{|V|} \sum_{x \in V} \text{ATM}(x^T \bar{V})$$

Mean: 2.75 Standard deviation: 0.34
Conicity value: 0.6

Conicity

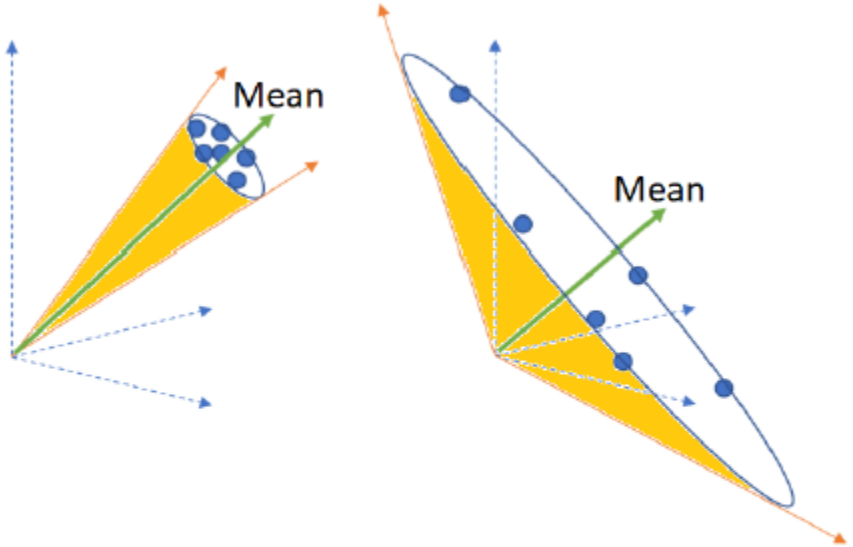


Figure 1: Conicity for a set of vectors. Left: high conicity with a small vector spread obtained from multiplicative systems. Right: low conicity with a large vector spread obtained from additive systems.

Deeper Look

Response to be evaluated	ADEM mean	ideal score
ground-truth response	2.75	5
context repeated as response	3.03	1
machine generated response*	2.64	1
swapping reference response and machine response	2.6	5

Table 3: ADEM scores on simple test cases (*Machine generated responses are obtained by training a GAN based neural dialogue generation model (Li et al. 2017))

批判

Response to be evaluated	mean	SD	%1 SD
Reference response	2.75	0.34	71.65
Punctuation removed	2.85	0.31	71.65
NLTK stopwords removed	2.69	0.33	70.60
25 common stopwords removed	2.80	0.24	69.08
[pro]nouns and verbs only	2.80	0.36	68.96
Named entities removed	2.74	0.35	70.60
Replace words with synonyms	2.83	0.32	70.36
Jumble words in the sentence	2.73	0.33	72
Reverse the response	2.75	0.33	68.84
Retain only nouns	2.73	0.39	68.26
Repeat words in the response	2.70	0.36	71.41
Generic and Irrelevant responses:			
I'm sorry, can you repeat?	2.65	0.34	69.43
I will do	2.69	0.34	70
fantastic! how are you?	3.18	0.4	69.4

Table 4: ADEM scores on simple dataset variants. The last column indicates the percentage of scores within one standard deviation of the mean score.

Variant	Pearson	Spearman	Better score
Punctuation removed	0.55	0.5	64.41%
NLTK stopwords removed	0.78	0.76	37.92%
25 common stopwords removed	0.6	0.57	60.33%
[pro]nouns and verbs	0.52	0.49	56.71%
Named entities removed	0.98	0.97	11.2%
Replace words with synonyms	0.79	0.75	68.03%
Jumble words in the sentence	0.68	0.64	47.02%
Reverse the response	0.52	0.49	48.66%
Retain only nouns	0.29	0.26	50.64%
Repeat words in the response	0.91	0.90	37.57%
"fantastic! how are you?"	0.34	0.32	86.93%

Table 5: Correlation of ADEM scores on different variants of the response with the ADEM scores on original (reference) response. p-values in all these cases are < 0.001. The last column indicates the percentage of times the concerned variant received a better score than original

Whitebox Attack on ADEM

- Guided Backpropagation

Questions?