EMNLP 2020 Follow-up

Chiyu Zhang

SentiLARE: Sentiment-Aware Language Representation Learning with Linguistic Knowledge

Pei Ke[*], **Haozhe Ji**[*], **Siyang Liu**, **Xiaoyan Zhu**, **Minlie Huang**[†]
Department of Computer Science and Technology, Institute for Artificial Intelligence,
State Key Lab of Intelligent Technology and Systems,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China
kepei1106@outlook.com,{jhz20,siyang-l18}@mails.tsinghua.edu.cn
{zxy-dcs,aihuang}@tsinghua.edu.cn

SentiLARE

Goal: Introduce linguistic knowledge into pre-trained language representation model.

In this paper, they propose a novel pre-trained language representation model called SentiLARE to deal with **two** challenges:

1) Knowledge acquisition across different contexts.

2) Knowledge integration into pre-trained models.

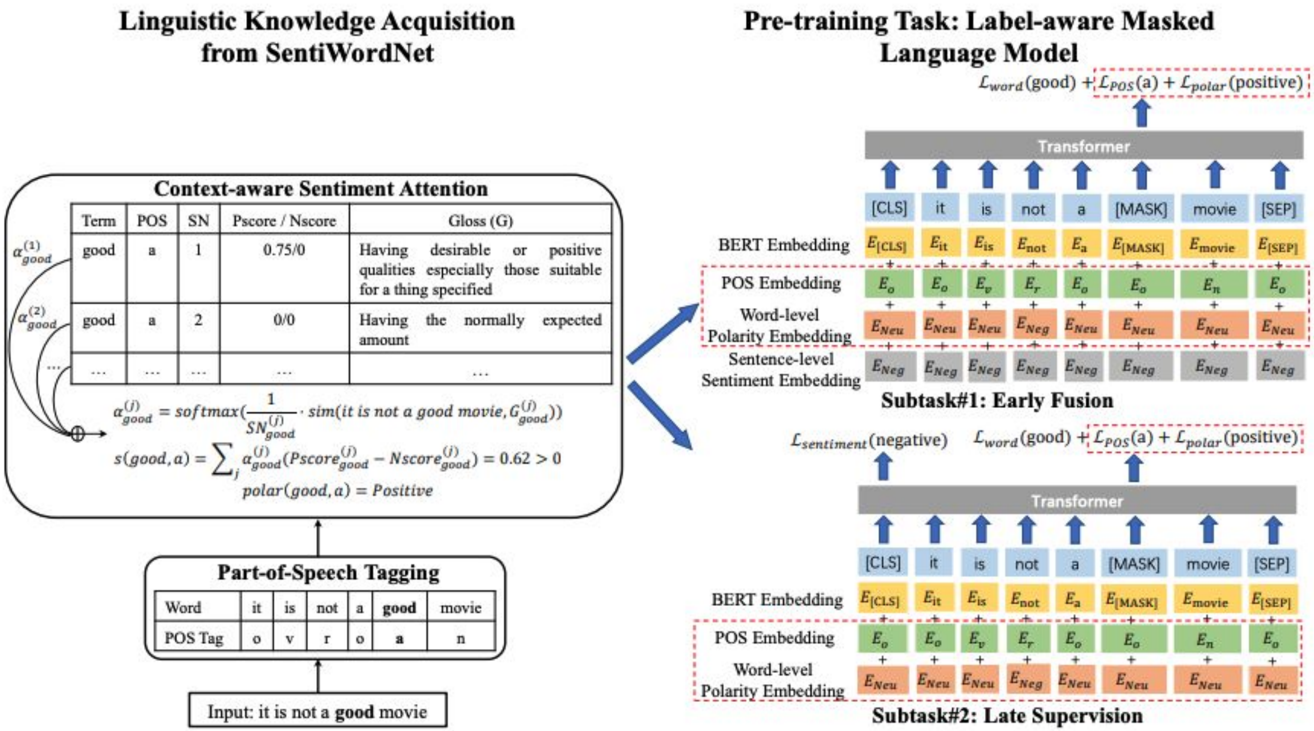


Figure 1: Overview of SentiLARE. This model first labels each word with its part-of-speech tag, and then uses the word and tag to match the corresponding senses in SentiWordNet. The sentiment polarity of each word is obtained by weighting the matched senses with context-aware sentiment attention. During pre-training, the model is trained based on label-aware masked language model including *early fusion* and *late supervision*. Red dotted boxes denote that the linguistic knowledge is used in input embedding or pre-training loss function.

Linguistic Knowledge Acquisition

- Overview of knowledge acquisition module
  - Input: a sequence of words $X = \{x_1, \dots, x_n\}$
  - Output: a sequence of words, POS tags, and sentiment polarities

$$X_k = \{(x_i, pos_i, polar_i)_{i=1}^{n}\}$$

- POS tagging
  - Stanford log-linear part-of-speech tagger
  - Five POS labels: verb (v), noun (n), adjective (a), adverb (r), others (o)

Slides from EMNLP20 presentation, https://slideslive.com/38938931/sentilare-sentimentaware-language-representation-learning-with-linguistic-knowledge

Sentiment polarity acquisition

- Find the m senses for (x_i, pos_i):

$$\{(SN_i^{(j)}, Pscore_i^{(j)}, Nscore_i^{(j)}, G_i^{(j)})_{j=1}^{m}\}$$

- Context-aware sentiment attention:

$$\alpha_i^{(j)} = softmax\left(\frac{1}{SN_i^{(j)}} \cdot sim(X, G_i^{(j)})\right)$$

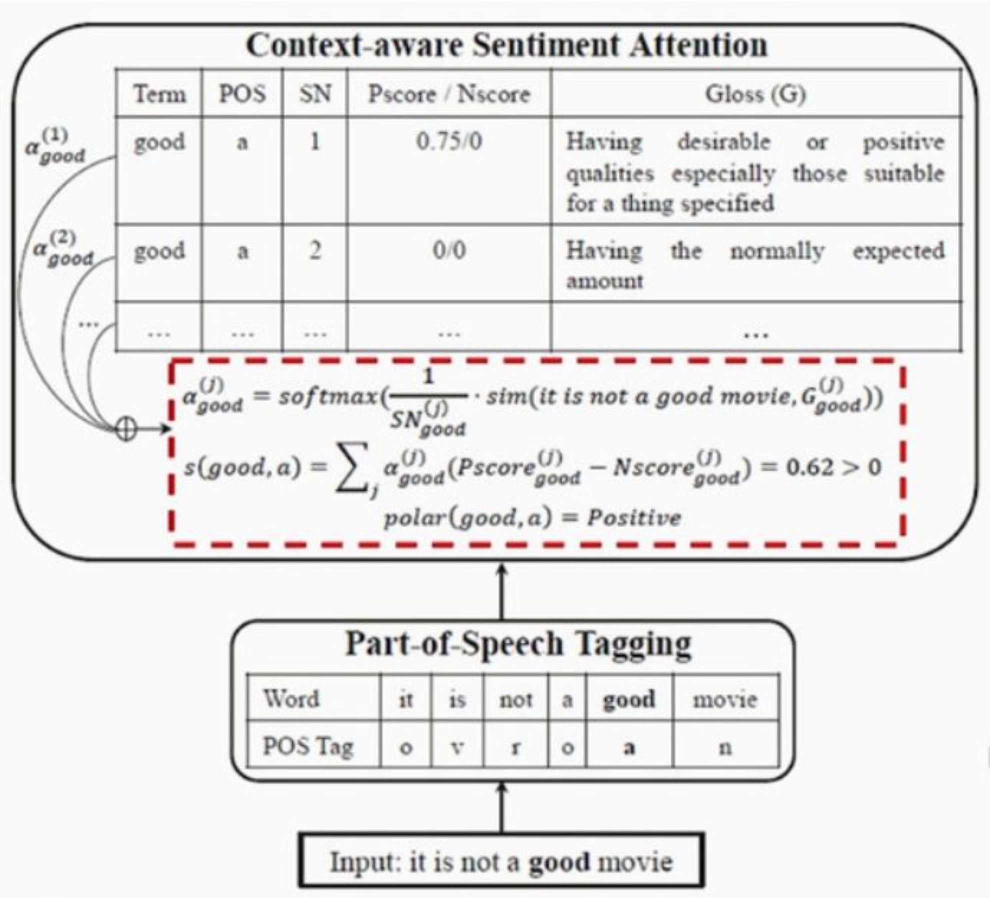
$$sim\left(X, G_i^{(j)}\right) = \cos(\mathrm{SBERT}(X), \mathrm{SBERT}(G_i^{(j)}))$$

- Sentiment score and polarity:

$$s(x_i, pos_i) = \sum_{j=1}^{m} \alpha_i^{(j)} \cdot (Pscore_i^{(j)} - Nscore_i^{(j)})$$

$$polar_i = \begin{cases} Positive, & s(x_i, pos_i) > 0 \\ Negative, & s(x_i, pos_i) < 0 \\ Neutral, & s(x_i, pos_i) = 0 \end{cases}$$

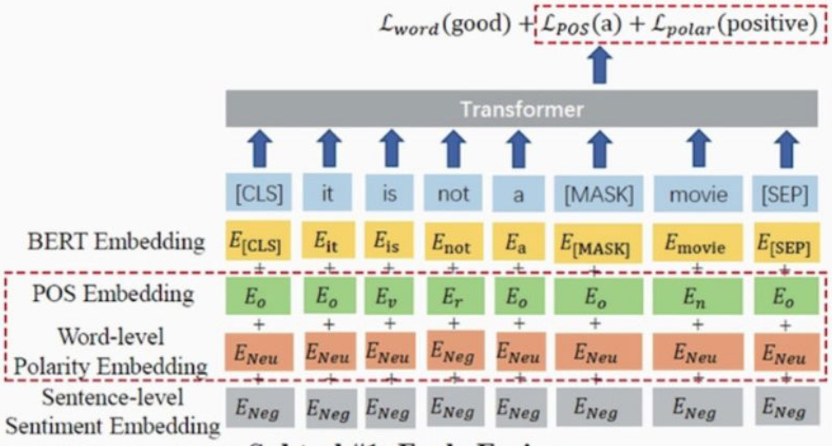
Slides from EMNLP20 presentation, https://slideslive.com/38938931/sentilare-sentimentaware-language-representation-learning-with-linguistic-knowledge



Slides from EMNLP20 presentation, https://slideslive.com/38938931/sentilare-sentimentaware-language-representation-learning-with-linguistic-knowledge

Knowledge Integration

- Early fusion

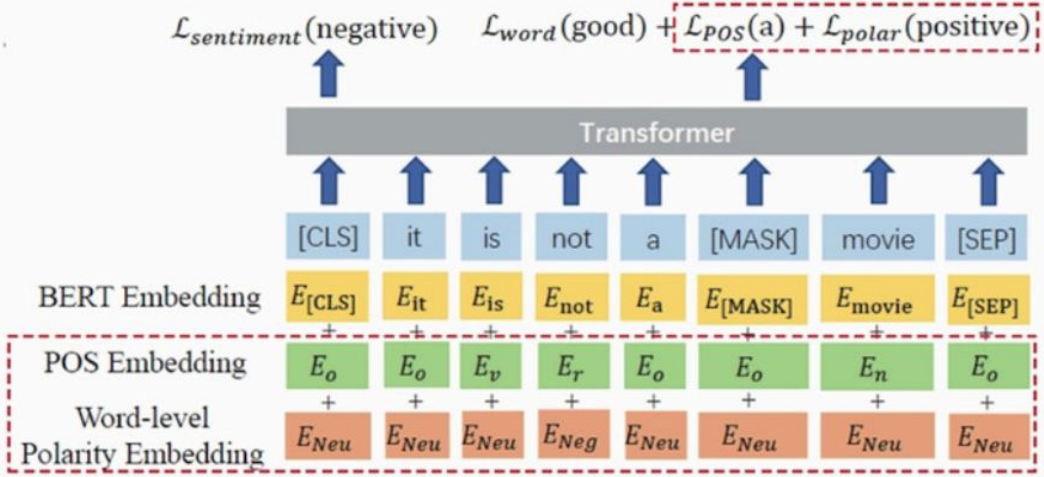


Subtask#1: Early Fusion

- The embedding of the sentence-level sentiment label is **early** added to the input embeddings.
- Loss function:

$$\mathcal{L}_{EF} = -\sum_{t=1}^{n} m_t \cdot [\log P(x_t|\hat{X}_k, l) + \log P(pos_t|\hat{X}_k, l) + \log P(polar_t|\hat{X}_k, l)]$$

Late supervision



Subtask#2: Late Supervision

- The sentence-level sentiment label is used as the **late** supervision signal.
- Loss function:

$$\mathcal{L}_{LS} = -\log P(l|\hat{X}_k) - \sum_{t=1}^{n} m_t \cdot [\log P(x_t|\hat{X}_k) + \log P(pos_t|\hat{X}_k) + \log P(polar_t|\hat{X}_k)]$$

Experiments

- Pre-training settings
  - Dataset: Yelp Dataset Challenge 2019 (6.68 million reviews with review-level sentiment labels)
  - Base model: RoBERTa-base
- Fine-tuning settings

Task	Input Format	Output Hidden States
Sentence-level Sentiment Classification	$\texttt{[CLS]}\, x_1, \cdots x_n \,\texttt{[SEP]}$	$h_{\texttt{[CLS]}}$
Aspect Term Extraction	$\texttt{[CLS]}\, x_1 \cdots x_n \,\texttt{[SEP]}$	$h_1, h_2, \cdots, h_n$
Aspect Term Sentiment Classification	$\texttt{[CLS]}\, a_1 \cdots a_l \,\texttt{[SEP]}\, x_1 \cdots x_n \,\texttt{[SEP]}$	$h_{\texttt{[CLS]}}$
Aspect Category Detection	$\texttt{[CLS]}\, x_1 \cdots x_n \,\texttt{[SEP]}$	$h_{\texttt{[CLS]}}$
Aspect Category Sentiment Classification	$\texttt{[CLS]}\, a_1 \cdots a_l \,\texttt{[SEP]}\, x_1 \cdots x_n \,\texttt{[SEP]}$	$h_{\texttt{[CLS]}}$
Text Matching	$\texttt{[CLS]}\, x_1 \cdots x_n \,\texttt{[SEP]}\, y_1 \cdots y_m \,\texttt{[SEP]}$	$h_{\texttt{[CLS]}}$

- Baseline
  - General pre-trained models: BERT, XLNet, RoBERTa
  - Task-specific pre-trained models: BERT-PT, TransBERT, SentiBERT
  - Task-specific models without pre-training

Results

Model	SST	MR	IMDB	Yelp-2	Yelp-5
SOTA-NPT	55.20♯	82.50♯	93.57†	97.27‡	69.15‡
BERT	53.37	87.52	93.87	97.74	70.16
XLNet	56.33	89.45	95.27	97.41	70.23
RoBERTa	54.89	89.41	94.68	97.98	70.12
BERT-PT	53.24	87.30	93.99	97.77	69.90
TransBERT	55.56	88.69	94.79	96.73	69.53
SentiBERT	56.87	88.59	94.04	97.66	69.94
SentiLARE	**58.59****	**90.82****	**95.71****	**98.22****	**71.57****

Table 3: Accuracy on sentence-level sentiment classification (SSC) benchmarks (%). SOTA-NPT means the state-of-the-art performance from the baselines without pre-training, where the results marked with ♯, † and ‡ are re-printed from Chen et al. (2019), Sachan et al. (2019) and Wang (2018), respectively. ** indicates that our model significantly outperforms the best pre-trained baselines on the corresponding dataset (t-test, *p-value*$<$ 0.01).

Task	ATE		ATSC				ACD		ACSC			
Dataset	Lap14	Res14	Lap14		Res14		Res14	Res16	Res14		Res16	
Model	F1	F1	Acc.	MF1.	Acc.	MF1.	F1	F1	Acc.	MF1.	Acc.	MF1.
SOTA-NPT	$81.59^{\sharp}$	-	$77.19^{\dagger}$	$72.99^{\dagger}$	$82.30^{\dagger}$	$74.02^{\dagger}$	$90.61^{\ddagger}$	$78.38^{\ddagger}$	$85.00^{\flat}$	$73.53^{\flat}$	-	-
BERT	83.22	87.68	78.18	73.11	83.77	76.06	90.48	72.59	88.35	80.40	86.55	71.19
XLNet	86.02	89.41	80.00	75.88	84.93	76.70	91.35	73.00	91.63	84.79	87.46	73.06
RoBERTa	87.25	89.55	81.03	77.16	86.07	79.21	91.69	77.89	90.67	83.81	88.38	76.04
BERT-PT	85.99	89.40	78.46	73.82	85.86	77.99	91.89	75.42	91.57	85.08	90.20	77.09
TransBERT	83.62	87.88	80.06	75.43	86.38	78.95	91.50	76.27	91.43	85.03	90.41	78.56
SentiBERT	82.63	88.67	76.87	71.74	83.71	75.42	91.67	73.13	89.68	82.90	87.08	72.10
SentiLARE	**88.22***	**91.15****	**82.16***	**78.70***	**88.32****	**81.63****	**92.22**	**80.71****	**92.97****	**87.30****	**91.29**	**80.00**

Table 5: F1, accuracy (Acc.) and Macro-F1 (MF1.) on four aspect-level sentiment analysis tasks including aspect term extraction (ATE), aspect term sentiment classification (ATSC), aspect category detection (ACD) and aspect category sentiment classification (ACSC) (%). SOTA-NPT means the state-of-the-art performance from the baselines without pre-training, where the results marked with $\sharp$, $\dagger$, $\ddagger$ and $\flat$ are re-printed from Xu et al. (2018), Sun et al. (2019a), Movahedi et al. (2019) and Wang et al. (2019b), respectively. - means that the results are not reported in the references. * indicates that our model significantly outperforms the best pre-trained baselines on the corresponding dataset (t-test, *p-value*<0.05), while ** means *p-value*<0.01.

Task	SSC	ATE	ATSC		ACD	ACSC	
Dataset	SST	Res14	Res14		Res16	Res14	
Model	Acc.	F1	Acc.	MF1.	F1	Acc.	MF1.
RoBERTa	54.89	89.55	86.07	79.21	77.89	90.67	83.81
SentiLARE	**58.59**	**91.15**	**88.32**	**81.63**	**80.71**	**92.97**	**87.30**
- EF	58.44	90.82	87.70	81.11	80.42	92.70	86.42
- LS	57.33	90.88	87.21	80.46	79.74	92.44	86.14
- EF - LS	56.91	90.74	86.95	79.71	78.92	91.32	84.73
- POS	58.15	90.94	87.98	81.38	80.27	92.51	86.61
- POL	57.95	90.63	87.64	81.34	79.40	92.46	86.30
- POS - POL	57.31	90.35	87.59	81.20	79.21	92.21	85.68

Table 6: Ablation test on sentiment analysis tasks. EF / LS / POS / POL denotes early fusion / late supervision / part-of-speech tag / word-level polarity, respectively.

Train No Evil: Selective Masking for Task-Guided Pre-Training

Yuxian Gu[1,2,3], **Zhengyan Zhang**[1,2,3], **Xiaozhi Wang**[1,2,3], **Zhiyuan Liu**[1,2,3†], **Maosong Sun**[1,2,3]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Institute for Artificial Intelligence, Tsinghua University, Beijing, China
[3]State Key Lab on Intelligent Technology and Systems, Tsinghua University, Beijing, China
{gu-yx17, zy-z19, wangxz20}@mails.tsinghua.edu.cn

To better capture domain-specific and task specific patterns, we propose a three-stage framework by adding a task-guided pre-training stage with selective masking between the general pretraining and fine-tuning.

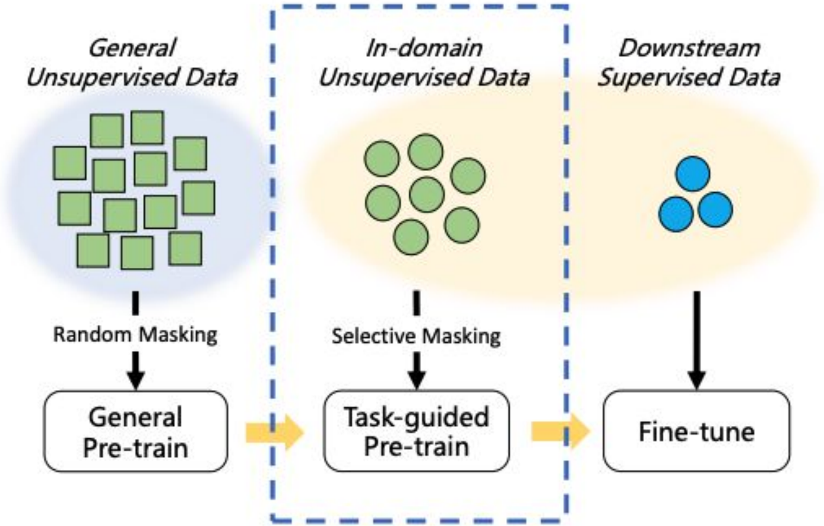


Figure 1: The overall three-stage framework. We add task-guided pre-training between general pre-training and fine-tuning to efficiently and effectively learn the domain-specific and task-specific language patterns.

Selective Masking

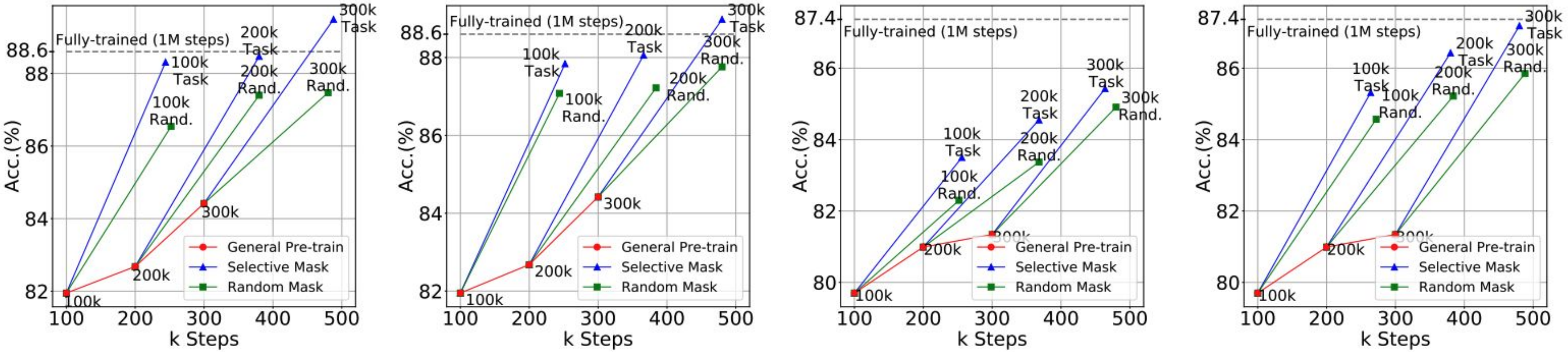
- Downstream Mask
  - Train a model to calculate classification scores of each sentence.
  - Calculate the score $S(*)$ of each token.
  - Judge the important of the token by its score.

$$\text{The food tastes good here .} \longrightarrow \boxed{\text{BERT}} \longrightarrow P(\textit{positive} \mid \text{The food tastes good here . })$$

$$S(\text{good}) = P(\textit{positive} \mid \text{The food tastes good here . }) - P(\textit{positive} \mid \text{The food tastes good})$$

$S(\text{good}) < \delta \longrightarrow$ Important, mask "good" in the sentence

Slides from EMNLP20 presentation, https://slideslive.com/38938884/train-no-evil-selective-masking-for-taskguided-pretraining



(a) Sem14-Rest + Yelp (b) Sem14-Rest + Amazon (c) MR + Yelp (d) MR + Amazon

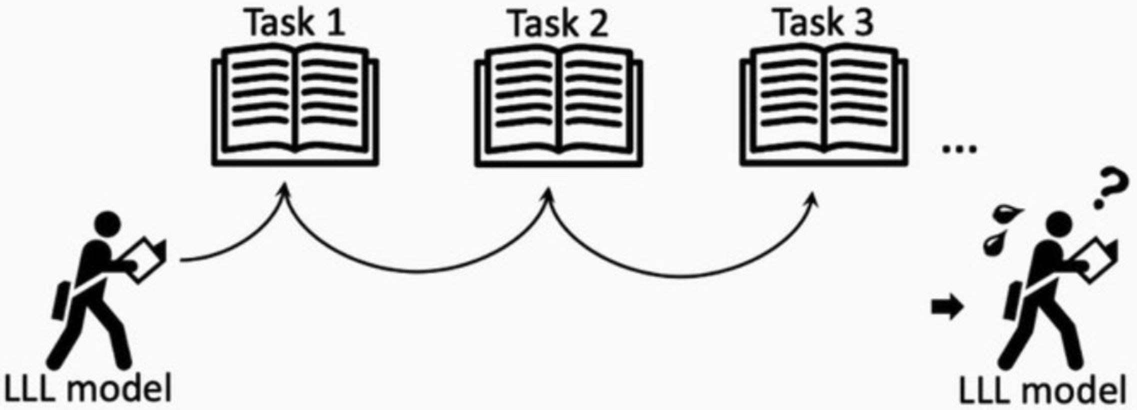
		MR	Sem14-Rest
w/o Task-guided pre-training		87.37	88.60
Amazon	Random	88.35	90.40
	Selective	**89.51****	**91.56****
Yelp	Random	87.20	90.70
	Selective	**88.15****	**91.87***

Table 1: Test accuracies of models trained with different methods (without task-guided pre-training or task-guided pre-training with different masking strategies) after full general pre-training (1M steps). $*$ and $**$ indicate statistically significant ($p < .05$ and $p < .001$).

Lifelong Language Knowledge Distillation

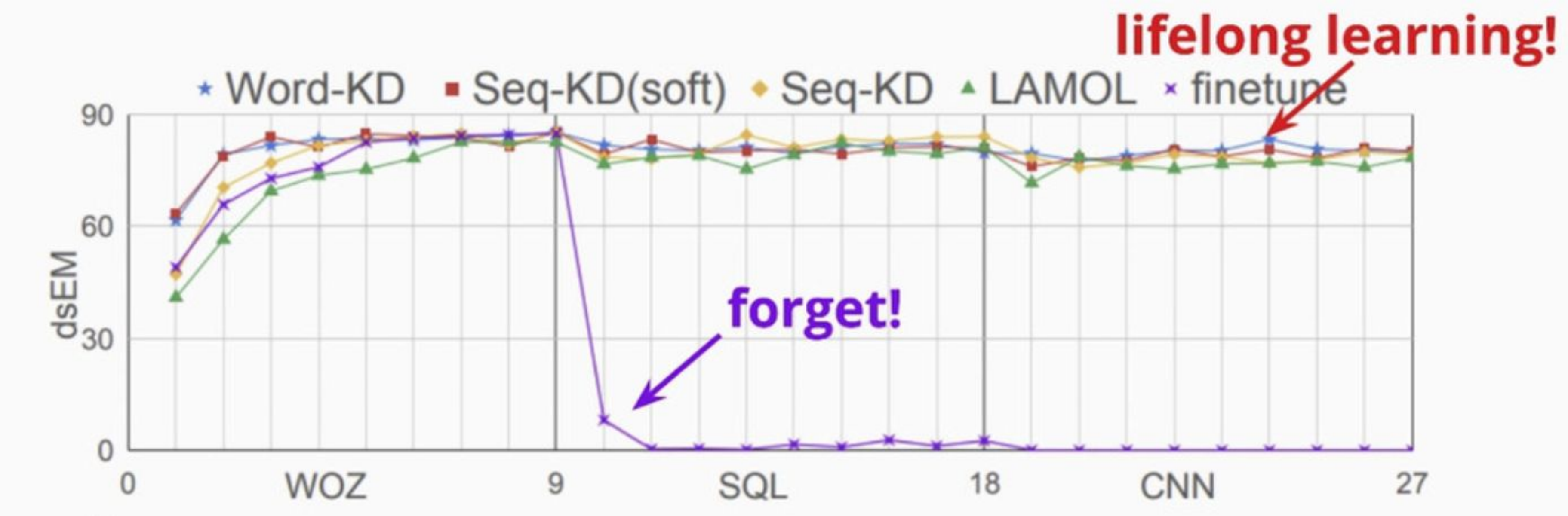
Yung-Sung Chuang Shang-Yu Su Yun-Nung Chen
National Taiwan University, Taipei, Taiwan
{b05901033,f05921117}@ntu.edu.tw y.v.chen@ieee.org

- Facing <u>catastrophic forgetting</u> problem



(a) Normal Lifelong Language Learning.

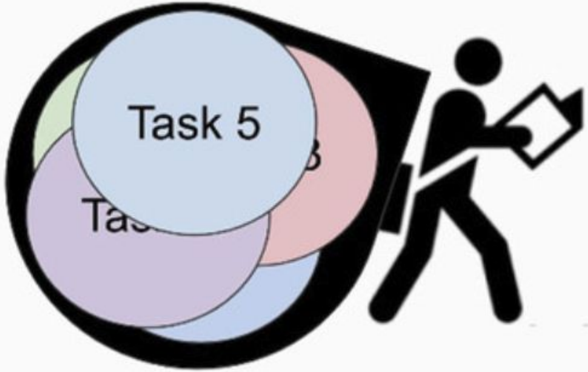
Slides from EMNLP20 presentation, https://slideslive.com/38938863/lifelong-language-knowledge-distillation

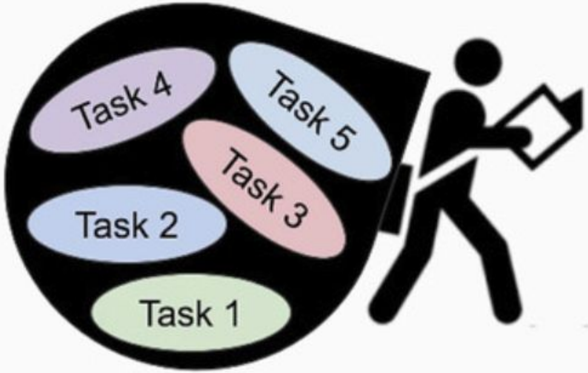


lifelong learning!
Word-KD
Seq-KD(soft)
Seq-KD
LAMOL
finetune
dsEM
90
60
30
0
forget!
0
WOZ
9
SQL
18
CNN
27

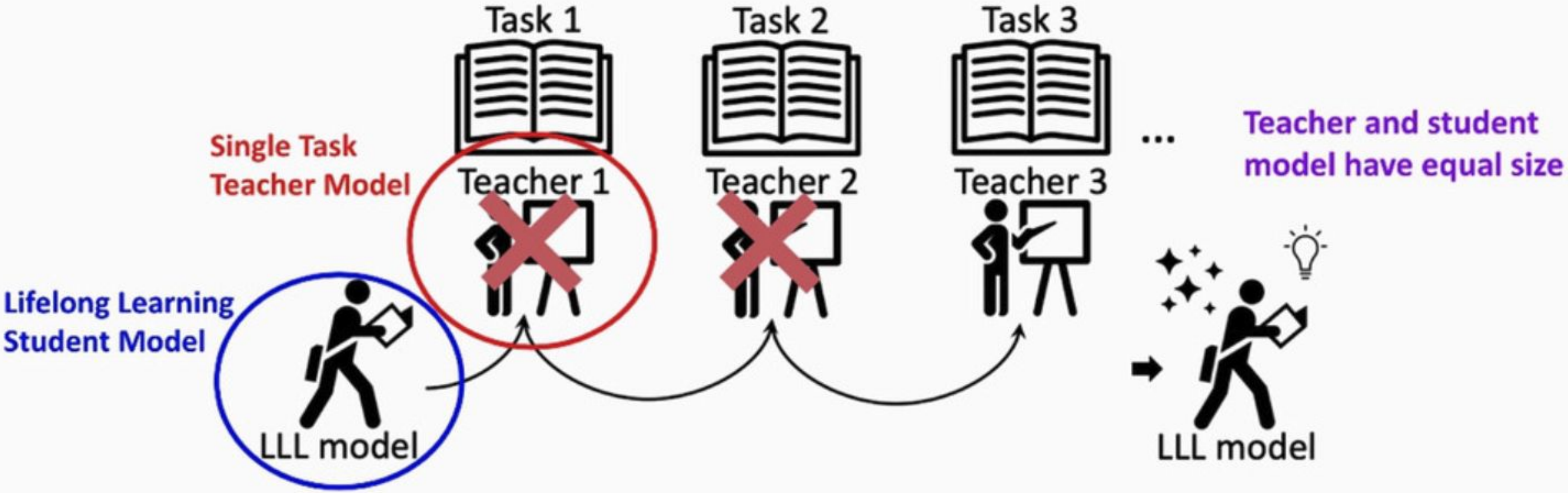
Motivation

- Model has limited capacity.
- Previously learned knowledge is affected by new knowledge.
- Compress the knowledge of incoming tasks.





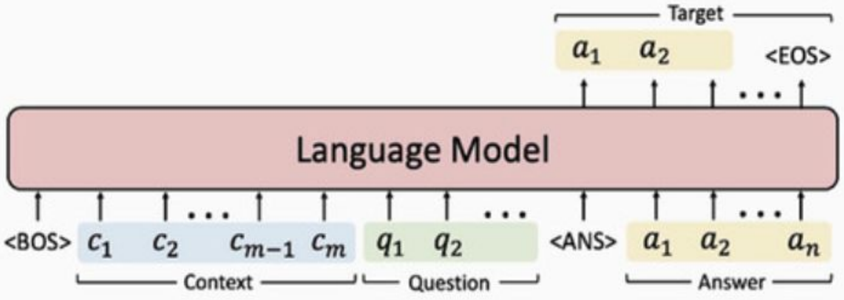
- Mitigating catastrophic forgetting by knowledge distillation



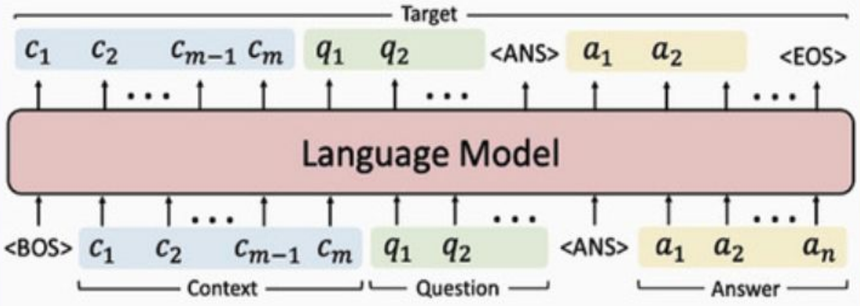
Base Model: LAMOL

LAnguage MOdeling for Lifelong Language Learning, ICLR 2020

- generate pseudo training samples for the previous tasks
- train on both data from the new task and the generated pseudo-data



(a) Learning to solve target tasks (QA).



(b) Learning to generate pseudo-data (LM).

Knowledge Distillation

1. Word-Level (Word-KD): ***soft target***

$$\mathcal{L}_{\text{Word-KD}}(x;\theta_S;\theta_T) =$$

$$\sum_{t=t_0}^{T} \sum_{k=1}^{|\mathcal{V}|} -\underbrace{\boxed{P(\mathcal{V}_k \mid x_{<t};\theta_T)}}_{\text{teacher model output distribution}} \log \underbrace{\boxed{P(\mathcal{V}_k \mid x_{<t};\theta_S)}}_{\text{lifelong model output distribution}},$$

Knowledge Distillation

2. Sequence-Level (Seq-KD): ***hard target***

$$\mathcal{L}_{\text{Seq-KD}}(\hat{x};\theta_S) = \sum_{t=t_0}^{T} -\log P(\underbrace{\boxed{\hat{x}_t}}_{\text{greedy decode output from teacher model}} \mid \hat{x}_{<t};\theta_S).$$

Knowledge Distillation

3. Soft Sequence-Level (Seq-KD$_{\text{soft}}$): ***soft target***

greedy decode output from teacher model

$$\mathcal{L}_{\text{Seq-KD}_{\text{soft}}}(\hat{x};\theta_S;\theta_T) = \sum_{t=t_0}^{T}\sum_{k=1}^{|\mathcal{V}|} -\underbrace{\boxed{P(\mathcal{V}_k \mid \hat{x}_{<t};\theta_T)}}_{\text{teacher model output distribution}} \log \underbrace{\boxed{P(\mathcal{V}_k \mid \hat{x}_{<t};\theta_S)}}_{\text{lifelong model output distribution}}.$$

Loss Function

$$\mathcal{L}_{\text{new}}(X_i^m;\theta_S;\theta_T^m) = \mathcal{L}_{\text{new}}^{\text{QA}} + \mathcal{L}_{\text{new}}^{\text{LM}}$$
$$\mathcal{L}_{\text{new}}^{\text{QA}} = \mathcal{L}_{\text{Word-KD}}(X_i^m;\theta_S;\theta_T^m;t_0 = a_1)$$
$$\mathcal{L}_{\text{new}}^{\text{LM}} = \mathcal{L}_{\text{Word-KD}}(X_i^m;\theta_S;\theta_T^m;t_0 = 0),$$

$$\mathcal{L}_{\text{prev}}(X_i^{\text{prev}};\theta_S) = \mathcal{L}_{\text{prev}}^{\text{QA}} + \mathcal{L}_{\text{prev}}^{\text{LM}}$$
$$\mathcal{L}_{\text{prev}}^{\text{QA}} = \mathcal{L}_{\text{NLL}}(X_i^{\text{prev}};\theta_S;t_0 = a_1)$$
$$\mathcal{L}_{\text{prev}}^{\text{LM}} = \mathcal{L}_{\text{NLL}}(X_i^{\text{prev}};\theta_S;t_0 = 0).$$

$$\theta_S^* = \arg\min_{\theta_S}\Big(\sum_{X_i^m \in D_m} \mathcal{L}_{\text{new}} + \sum_{X_i^{\text{prev}} \in D_{\text{prev}}} \mathcal{L}_{\text{prev}}\Big)$$

Dataset	Metric	# Train	# Test
Sequence Generation for Different Tasks			
WikiSQL	lfEM	6,525	15,878
CNN/DailyMail	ROUGE	6,604	2,250
MultiWOZ	dsEM	2,536	1,646
Sequence Generation for Different Domains			
E2E NLG		6,000	2,000
RNNLG (rest.)		6,228	1,039
RNNLG (hotel)	ROUGE	6,446	1,075
RNNLG (tv)		8,442	1,407
RNNLG (laptop)		7,944	2,649
Text Classification for Different Tasks			
AGNews		115,000	7,600
Yelp		115,000	7,600
Amazon	Exact Match	115,000	7,600
DBPedia		115,000	7,600
Yahoo		115,000	7,600

Table 1: Dataset sizes and the evaluation metrics.

	Method	WOZ	CNN	SQL	Avg	WOZ	CNN	SQL	Avg	WOZ	CNN	SQL	Avg
		WOZ → CNN → SQL				CNN → SQL → WOZ				SQL → WOZ → CNN			
(a)	Finetune	0.0	26.3	64.3	30.2	84.6	6.8	2.1	31.2	0.1	26.0	0.0	8.7
(b)	LAMOL	67.6	27.3	62.5	52.4	83.0	27.8	60.8	57.2	76.1	26.0	55.0	52.4
(c)	(b) + Word-KD	**82.4**	27.6	**65.0**	**58.3**	**86.1**	27.5	63.2	**59.0**	79.5	26.2	**59.6**	55.1
(d)	(b) + Seq-KD$_{soft}$	81.0	26.9	64.7	57.5	84.1	27.6	**63.4**	58.4	**81.7**	25.9	58.4	**55.3**
(e)	(b) + Seq-KD	76.4	**28.0**	63.7	56.1	83.0	**28.3**	61.5	57.6	81.0	**27.5**	57.3	**55.3**
		WOZ → SQL → CNN				CNN → WOZ → SQL				SQL → CNN → WOZ			
(a)	Finetune	0.0	25.8	0.0	8.6	3.6	24.5	64.0	30.7	85.0	7.3	0.0	30.8
(b)	LAMOL	76.1	26.3	59.3	53.9	79.8	27.3	64.1	57.0	84.0	27.2	58.7	**56.6**
(c)	(b) + Word-KD	**81.4**	26.7	59.6	**55.9**	83.5	27.8	**65.0**	58.8	78.7	26.4	**59.0**	54.7
(d)	(b) + Seq-KD$_{soft}$	80.4	26.1	**59.9**	55.5	**83.7**	28.6	64.8	**59.0**	84.7	26.2	58.8	**56.6**
(e)	(b) + Seq-KD	77.2	**27.0**	59.5	54.5	82.8	**29.5**	64.4	58.9	**84.9**	**27.8**	57.3	**56.6**

Table 2: Detailed experimental results on MultiWOZ (WOZ), CNN/DailyMail (CNN), WikiSQL (SQL), with six different lifelong learning orders.

Averaged Results

- KD improve LAMOL by 2.2
- KD improve Multitask by 0.2

➔ Lifelong learning model has more space to improve

- KD declines the STD by 1.4

➔ Lifelong learning model can be more "order-robust" with KD

Non-Lifelong Methods		**WOZ**	**CNN**	**SQL**	**Avg**
(1)	Single QA	84.8	25.5	63.1	**57.8**
(2)	Single QA+LM	82.2	25.9	**63.7**	57.3
(3)	$\text{Multi}_{\text{same}}$ QA	66.2	25.6	53.0	48.3
(4)	$\text{Multi}_{\text{same}}$ QA+LM	59.0	26.3	53.6	46.3
(5)	$\text{Multi}_{\text{long}}$ QA	82.7	26.1	61.1	56.6
(6)	$\text{Multi}_{\text{long}}$ QA+LM	**85.4**	**26.7**	61.3	**57.8**
(7)	(6) + Seq-KD	84.4	27.6	61.8	58.0
Lifelong Methods (averaged over six orders)					
(a)	Finetune	28.9	19.5	21.7	23.4
(b)	LAMOL	77.7	27.0	60.0	54.9
(c)	(b) + Word-KD	81.9	27.0	**61.9**	57.0
(d)	(b) + $\text{Seq-KD}_{\text{soft}}$	**82.6**	26.9	61.7	**57.1**
(e)	(b) + Seq-KD	80.9	**28.0**	60.6	56.5
STD of Lifelong Methods					
(f)	Finetune	43.3	9.6	32.9	28.6
(g)	LAMOL	6.0	**0.7**	3.2	3.3
(h)	(g) + Word-KD	2.7	**0.7**	**2.8**	2.1
(i)	(g) + $\text{Seq-KD}_{\text{soft}}$	**1.8**	1.0	3.0	**1.9**
(j)	(g) + Seq-KD	3.4	0.9	3.1	2.5

upper bounds

+0.2

KD improves little on multitask

without KD

+2.2

with KD

-1.4

KD improves order-robustness

Method	amazon	yelp	yahoo	ag	dbpedia	Avg
$\text{Single}_{(QA)}$	55.9	63.3	70.6	93.6	99.0	76.5
$\text{Single}_{(QA+LM)}$	56.9	**64.5**	70.1	93.7	**99.1**	76.9
$\text{Multi}_{(QA)}$	56.6	63.3	69.2	93.7	99.0	76.4
$\text{Multi}_{(QA+LM)}$	**57.8**	64.4	**70.9**	**94.0**	**99.1**	**77.2**
Left-to-right (amazon → yelp → yahoo → ag → dbpedia)						
LAMOL	52.7	61.6	70.3	93.6	99.1	75.5
+ Word-KD	**57.5**	**63.6**	**71.3**	**93.9**	**99.2**	**77.1**
+ Seq-KD_{soft}	55.7	62.0	**71.3**	**93.9**	**99.2**	76.4
+ Seq-KD	56.8	62.3	71.1	93.4	99.1	76.6
Right-to-left (dbpedia → ag → yahoo → yelp → amazon)						
LAMOL	57.9	63.5	70.7	91.7	98.3	76.4
+ Word-KD	57.0	64.1	**73.2**	**92.7**	**98.8**	**77.1**
+ Seq-KD_{soft}	57.0	64.1	71.9	92.4	**98.8**	76.8
+ Seq-KD	**58.4**	**64.4**	71.7	91.5	**98.8**	76.9

Table 5: Experimental results on five text classification datasets.