

# Distilling Task-Specific Knowledge from BERT into Simple Neural Networks

- 저자:
  - Raphael Tang\*, Yao Lu\*, Linqing Liu\*, Lili Mou, Olga Vechtomova, and Jimmy Lin  
(University of Waterloo)
- 발표:
  - Presenter: 윤주성
  - Date: 191211

# Who is an Author?

- ICASSP를 들고 있는 NLP 하던 분인 듯
- 보통은 문서분류쪽 많이 한듯



Raphael Tang

University of Waterloo  
uwaterloo.ca의 이메일 확인됨

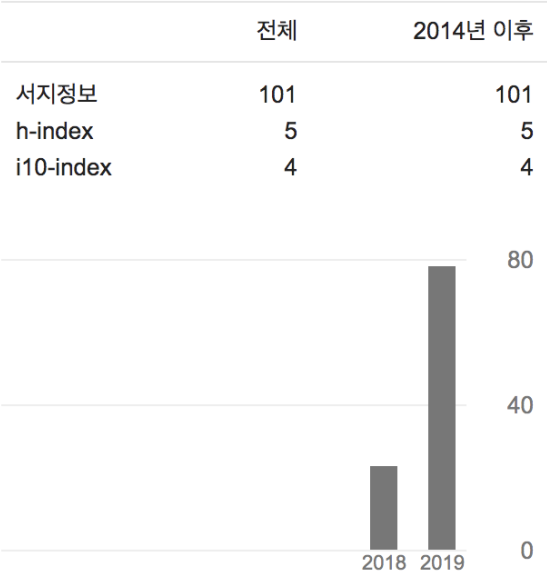
Deep learning   natural language processing   speech recognition

팔로우


내 프로필 만들기

제목	인용	연도
Deep Residual Learning for Small-Footprint Keyword Spotting R Tang, J Lin 2018 IEEE International Conference on Acoustics, Speech and Signal ...	38	2018
An Experimental Analysis of the Power Consumption of Convolutional Neural Networks for Keyword Spotting R Tang, W Wang, Z Tu, J Lin 2018 IEEE International Conference on Acoustics, Speech and Signal ...	14	2018
DocBERT: BERT for Document Classification A Adhikari, A Ram, R Tang, J Lin arXiv preprint arXiv:1904.08398	13	2019
Distilling Task-Specific Knowledge from BERT into Simple Neural Networks R Tang*, Y Lu*, L Liu*, L Mou, O Vechtomova, J Lin arXiv preprint arXiv:1903.12136	13	2019
Honk: A PyTorch Reimplementation of Convolutional Neural Networks for Keyword Spotting R Tang, J Lin arXiv preprint arXiv:1710.06554	9	2017
Rethinking Complex Neural Network Architectures for Document Classification A Adhikari*, A Ram*, R Tang, J Lin Proceedings of the 2019 Conference of the North American Chapter of the ...	5	2019


## 인용



## 공동 저자

- 

Jimmy Lin  
University of Waterloo

>
- 

Achyudh Ram  
University of Waterloo

>

## 느낀점

- 아이디어는 간단함
- Data Augmentation을 넣은건 좋았음
- 그러나 성능이 좋아진게 Distillation 때문인지 Data Augmentation 때문인지를 정확히 다루지 않아서.. 이 부분이 이 논문의 최대 예러임

## Abstract

- 요즘엔 BERT, ELMo, and GPT 같은 deep language representation model이 대세임
- 이러한 발전은 예전에 쓰던 shallower neural networks 를 안쓰게 만듦
- 본 논문에서는 lightweight neural network도 구조 변경, 추가 데이터, 추가 feature 없이도 아직 쓸만하다는걸 보여주려고함
- BERT에서 Knowledge distillation 해서 성능을 높여보려고함
- sentence classification, sentence-pair task 등으로 테스트 할 것임
- ELMo보다 100배 적은 파라미터, 15배 빠른 추론속도로 비슷한 성능을 얻음

# 1. Introduction

- BERT등이 등장하면서 "first-generation" neural network가 잘 안쓰이게됨
- 본 논문에서는 간단하지만 효과적인 transfer 기법을 제안하고자함 (task-specific knowledge from BERT)
- single-layer BiLSTM을 사용할거고, BERT로부터 배울 것임
- 효율적인 knowledge transfer 를 위해선 데이터가 많이 필요하니, unlabeled dataset으로부터 teacher output을 만들어서 student로 학습하게 할 것임

## 2. Related Work

- NLP에서 CNN, RNN 등이 발달됨
- 최근엔, ELMo(6가지 task SOTA 찍었음), BERT등이 등장함
- Model Compression:
  - local error-based method for pruning unimportant weights (LeCun et al. (1990))
  - Han et al. (2015) propose a simple compression pipeline, achieving 40 times reduction in model size without hurting accuracy. Unfortunately, these techniques induce irregular weight sparsity, which precludes highly optimized computation routines
  - quantizing neural networks (Wu et al., 2018); in the extreme, Courbariaux et al. (2016) propose binarized networks with both binary weights and binary activations
- 위에서 소개된 Model Compression과는 다르게 본 논문에서는 knowledge distillation approach (Ba and Caruana, 2014; Hinton et al., 2015) 를 사용하고자함
- NLP에서 이미 이것에 대해 적용된 연구들이 있음(In the NLP literature, it has previously been used in neural machine translation (Kim and Rush, 2016) and language model- ing (Yu et al., 2018))

### 3. Our Approach

- First, teacher model과 student model을 선택 후 학습
- Second, 저자의 distillation procedure로 학습
  - logits-regression objective 사용
  - transfer dataset 구축

## 3.1 Model Architecture

- Teacher network : pretrained, fine-tuned BERT
  - feature vector  $\mathbf{h} \in \mathbb{R}^d$  위에 우리가 사용할 classifier를 task에 맞게 추가해서 쓸것임
  - For single-sentence classification
    - 다음과 같이 softmax 추가해서 쓸 것임( $k$  is the number of label)  $\mathbf{y}^{(B)} = \text{softmax}(W\mathbf{h})$ , where  $W \in \mathbb{R}^{k \times d}$
  - For sentence-pair task
    - 두 문장에 대한 BERT features를 concat후 softmax layer에 넣는 방식으로 함
  - 학습시에는 BERT와 classifier에 대한 param을 둘다 업데이트하고 cross-entropy loss 사용함



### 3.1 Model Architecture

- Student model: single-layer BiLSTM with a non-linear classifier
  - For classification
    - last step의 값을 concat 후 fc layer with ReLU 에 feed해서 softmax layer 로 분류함
  - For Sentence-pair tasks
    - BiLSTM encoder weights를 share해서 siamese architecture로 사용 sentence vectors  $\mathbf{h}_{s1}$  and  $\mathbf{h}_{s2}$  를 만들어냄
    - $f(\mathbf{h}_{s1}, \mathbf{h}_{s2}) = [\mathbf{h}_{s1}, \mathbf{h}_{s2}, \mathbf{h}_{s1} \odot \mathbf{h}_{s2}, |\mathbf{h}_{s1} - \mathbf{h}_{s2}|]$ ,
    - where  $\odot$  denotes elementwise multiplication
    - attention이나, layer norm 같은 스킴은 최대한 제외하고 BiLSTM의 representation Power에만 국한하는 설계를 함

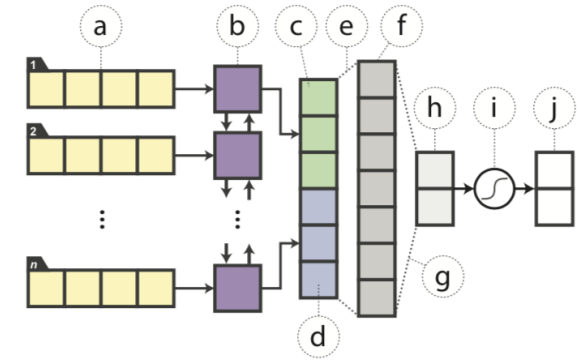


Figure 1: The BiLSTM model for single-sentence classification. The labels are (a) input embeddings, (b) BiLSTM, (c, d) backward and forward hidden states, respectively, (e, g) fully-connected layer; (e) with ReLU, (f) hidden representation, (h) logit outputs, (i) softmax activation, and (j) final probabilities.

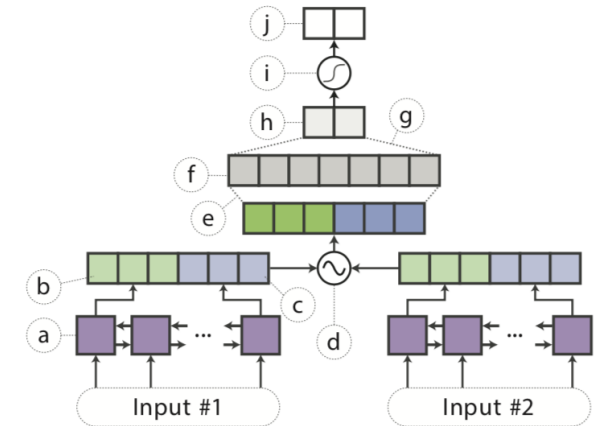


Figure 2: The siamese BiLSTM model for sentence matching, with shared encoder weights for both sentences. The labels are (a) BiLSTM, (b, c) final backward and forward hidden states, respectively, (d) concatenate-compare unit, (e, g) fully connected layer; (e) with ReLU, (f) hidden representation, (h) logit outputs, (i) softmax activation, and (j) final probabilities.

## 3.2 Distillation Objective

- In addition to a one-hot predicted label, the teacher's predicted probability is also important. In binary sentiment classification, for example, some sentences have a strong sentiment polarity, whereas others appear neutral.
- If we use only the teacher's predicted one-hot label to train the student, we may lose valuable information about the prediction uncertainty.

$$\tilde{y}_i = \text{softmax}(\mathbf{z}) = \frac{\exp\{\mathbf{w}_i^\top \mathbf{h}\}}{\sum_j \exp\{\mathbf{w}_j^\top \mathbf{h}\}}$$

where  $w_i$  denotes the  $i^{\text{th}}$  row of softmax weight  $W$ , and  $z$  is equivalent to  $\mathbf{w}^\top \mathbf{h}$ .

- Training on logits makes learning easier for the student model since the relationship learned by the teacher model across all of the targets are equally emphasized (Ba and Caruana, 2014).
- student network's logits과 teacher's logits의 MSE로 distillation objective를 만듦 (Cross entropy등도 사용가능하나, 저자의 실험에서 MSE가 좀 더 결과가 좋았다고함)

$$\mathcal{L}_{\text{distill}} = \left\| \mathbf{z}^{(B)} - \mathbf{z}^{(S)} \right\|_2^2$$

- 최종적인 Loss는 기존의 one-hot에 대한 CE loss와 distill Loss를 weighted sum해서 사용함 ( $t$ 는 one-hot label)

$$\begin{aligned} \mathcal{L} &= \alpha \cdot \mathcal{L}_{\text{CE}} + (1 - \alpha) \cdot \mathcal{L}_{\text{distill}} \\ &= -\alpha \sum_i t_i \log y_i^{(S)} - (1 - \alpha) \left\| \mathbf{z}^{(B)} - \mathbf{z}^{(S)} \right\|_2^2 \end{aligned}$$

- unlabeled data의 경우엔 teacher가 예측한걸 기준으로 사용함 i.e.,  $t_i = 1$  if  $i = \arg \max y^{(B)}$  and 0 otherwise.

### 3.3 Data Augmentation for Distillation

- 데이터 셋 작으면 티쳐한테 배울때 효과가 별로 없음
- 그렇기 때문에 데이터셋 키우기로함 (with pseudo-labels provided by the teacher)
- 하지만.. NLP에서의 Data augmentation은 Computer vision에 비해 어려움
  - CV는 비슷한 이미지들이 매우 많음 (CIFAR-10 is a subset of the 80 million tiny images dataset)
  - CV는 이미지 회전이나 노이즈 추가등.. 방법이 많음 (Second, it is possible to synthesize a near-natural image by rotating, adding noise, and other distortions)
    - NLP에서 이 방법 쓰면 not be fluent되기 때문에 쓸 수 없음  $\pi$

### 3.3 Data Augmentation for Distillation

- 본 논문에서는 약간의 휴리스틱으로 task-agnostic data augmentation을 하려고함 (image distortion과 같진 않고, 비슷하다고 생각하면됨)
  - Masking:
    - $p_{mask}$  의 확률로 랜덤하게 단어를 [MASK]로 바꿈
    - Intuitively, this rule helps to clarify **the contribution of each word toward the label** (각 단어의 contribution을 파악하는데 도움이 된다고 주장)
    - e.g., the teacher network produces less confident logits for "I [MASK] the comedy" than for "I loved the comedy."
  - POS-guided word replacement:
    - $p_{pos}$  의 확률로 단어를 같은 pos 태그를 갖는 다른 단어로 교체함 (~~히히 요상한 방법일세~~)
    - 이러한 룰은 semantic을 방해하기도 함 (This rule perturbs the semantics of each example, e.g., "What do pigs eat?" is different from "How do pigs eat?")
  - n-gram sampling:
    - $p_{ng}$  의 확률로 예시 문장에서 n-gram을 샘플링함 (n is randomly selected from {1,2,...,5})
    - n-gram 외의 단어는 드랍해버리는것과 비슷한 효과고 마스킹보다 더 공격적인 방법임 (This rule is conceptually equivalent to dropping out all other words in the example, which is a more ag- gressive form of masking)

### 3.3 Data Augmentation for Distillation

- Data augmentation procedure:
  - training example  $\{w_1, \dots, w_n\}$
  - 단어에 대해서 iteration 하면서 각 단어에 대해서 유니폼 분포로 확률을 계산함  $X_i \sim \text{UNIFORM}[0, 1]$  for each  $w_i$
  - if  $X_i < p_{\text{mask}}$  , we apply masking to  $w_i$
  - if  $p_{\text{mask}} < X_i < p_{\text{mask}} + p_{\text{pos}}$  , we apply POS-guided word replace to  $w_i$
  - masking과 POS-guided swapping은 mutually exclusive하게 진행해서 한개가 적용되면 나머지는 적용안함
  - iteration이 끝나면,  $p_{ng}$ 의 확률로 n-gram sampling을 synthetic example(위에서 만든 문장)에 적용하면 final synthetic example이 완성됨
  - 이러한 전체 프로세스를 한 문장당  $n_{\text{iter}}$  \$번 적용해서 총  $n_{\text{iter}}$  \$개의 문장을 만듦 (중복은 제거)
  - For sentence-pair datasets, we cycle through augmenting the first sentence only (holding the second fixed) , the second sentence only (holding the first fixed) , and both sentences .

## 4. Experimental Setup

- Teacher Network으로써의 BERT는 large 버전 사용
  - BERT fine-tuning 할 땐 Adam opt with lr  $\{2,3,4,5\} \times 10^{-5}$  적용
  - val set 기준 best model 선택
  - 여기선 data augmentation 안씀
- Student model 학습할 땐 data augmentation 사용
  - soft logit target을 사용한 모델을  $BiLSTM_{SOFT}$  로 표기하겠음
  - 3.2 세션에서 weighted sum으로 기존 CE와 distillation Loss를 추가해서 만들었는데  $\alpha = 0$ 으로 셋팅해서 distillation objective 만 사용한게 젤 잘나왔음

## 4. Experimental Setup

### 4.1 Datasets

- GLUE에서 3개 뽑아서 씬
  - SST-2: movie reviews for binary sentiment classification (positive vs. negative)
  - MNLI: to predict the relationship between a pair of sentences as one of entailment, neutrality, or contradiction
  - QQP: binary label of each question pair indicates redundancy

### 4.2 Hyperparameters

- Student Model
  - BiLSTM hidden: 150 or 300
  - ReLU activated hidden: 200 or 400
  - Optim: AdaDelta (lr: 1.0  $\rho$ : 0.95)
  - Batch size: 50 (SST2), 256 (MNLI, QQP)
- Data Augmentation
  - $p_{mask} = p_{pos} = 0.1$  and  $p_{ng} = 0.25$
  - $n_{iter} = 20$  (SST),  $n_{iter} = 10$  (MNLI, QQP)

### 4.3 Baseline Models

- BERT
- OpenAI GPT

## 5. Results and Discussion

### 5.1 Model Quality

- our distillation approach of matching logits using the augmented training dataset, and achieve an absolute improvement of 1.9– 4.5 points against our base BiLSTM.
- ~~data augmentation 없이 distillation한 것도 보여줘야.. 설득력이 더 있을텐데 음..~~

#	Model	SST-2	QQP	MNLI-m	MNLI-mm
		Acc	F <sub>1</sub> /Acc	Acc	Acc
1	BERT <sub>LARGE</sub> (Devlin et al., 2018)	94.9	72.1/89.3	86.7	85.9
2	BERT <sub>BASE</sub> (Devlin et al., 2018)	93.5	71.2/89.2	84.6	83.4
3	OpenAI GPT (Radford et al., 2018)	91.3	70.3/88.5	82.1	81.4
4	BERT ELMo baseline (Devlin et al., 2018)	90.4	64.8/84.7	76.4	76.1
5	GLUE ELMo baseline (Wang et al., 2018)	90.4	63.1/84.3	74.1	74.5
6	Distilled BiLSTM <sub>SOFT</sub>	<b>90.7</b>	<b>68.2/88.1</b>	<b>73.0</b>	<b>72.6</b>
7	BiLSTM (our implementation)	86.7	63.7/86.2	68.7	68.3
8	BiLSTM (reported by GLUE)	85.9	61.4/81.7	70.3	70.8
9	BiLSTM (reported by other papers)	87.6 <sup>†</sup>	– /82.6 <sup>‡</sup>	66.9 <sup>*</sup>	66.9 <sup>*</sup>

Table 1: Test results on different datasets. The BiLSTM results reported by other papers are drawn from Zhou et al. (2016),<sup>†</sup> Wang et al. (2017),<sup>‡</sup> and Williams et al. (2017).<sup>\*</sup> All of our test results are obtained from the GLUE benchmark website.



## 5. Results and Discussion

### 5.2 Inference Efficiency

- On a single NVIDIA V100 GPU
- with a batch size of 512 on all 67350 sentences of the SST-2 training set
- our single-sentence model uses 98 and 349 times fewer parameters than ELMo and BERT<sub>LARGE</sub>, respectively
- **15 and 434 times faster**

	# of Par.	Inference Time
BERT <sub>LARGE</sub>	335 (349×)	1060 (434×)
ELMo	93.6 (98×)	36.71 (15×)
BiLSTM <sub>SOFT</sub>	0.96 (1×)	2.44 (1×)

Table 2: **Single-sentence model size and inference speed on SST-2. # of Par. denotes number of millions of parameters, and inference time is in seconds.**

## 6. Conclusion and Future Work

- Explore distilling the knowledge from BERT into a simple BiLSTM-based model
- The distilled model achieves comparable results with ELMo, while using much fewer parameters and less inference time
- Future work로는 더 단순한 모델로 KD하거나 더 복잡한 모델로 KD하거나..~~(이거 넘 당연한 발상 아닌가..)~~

### Code

ref: [https://github.com/qiangsiwei/bert\\_distill](https://github.com/qiangsiwei/bert_distill)