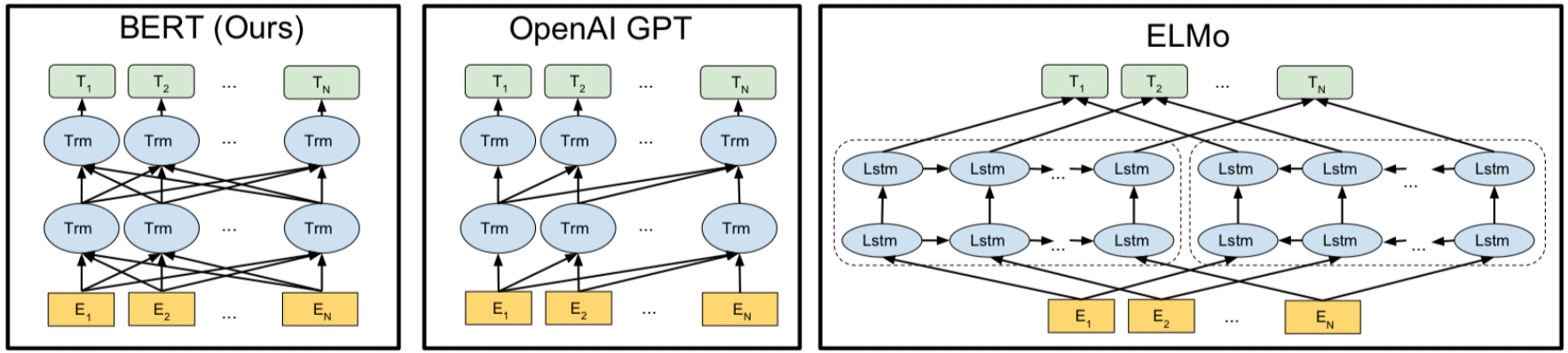
EXTREME LANGUAGE MODEL COMPRESSION WITH OPTIMAL SUBWORDS AND SHARED PROJECTIONS (ICLR 2020 under review)

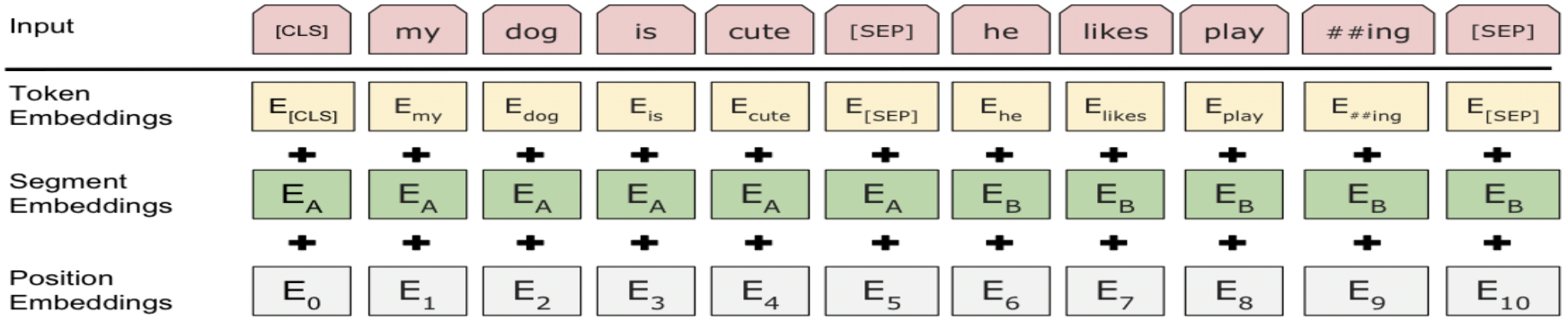
서상우

ABSTRACT

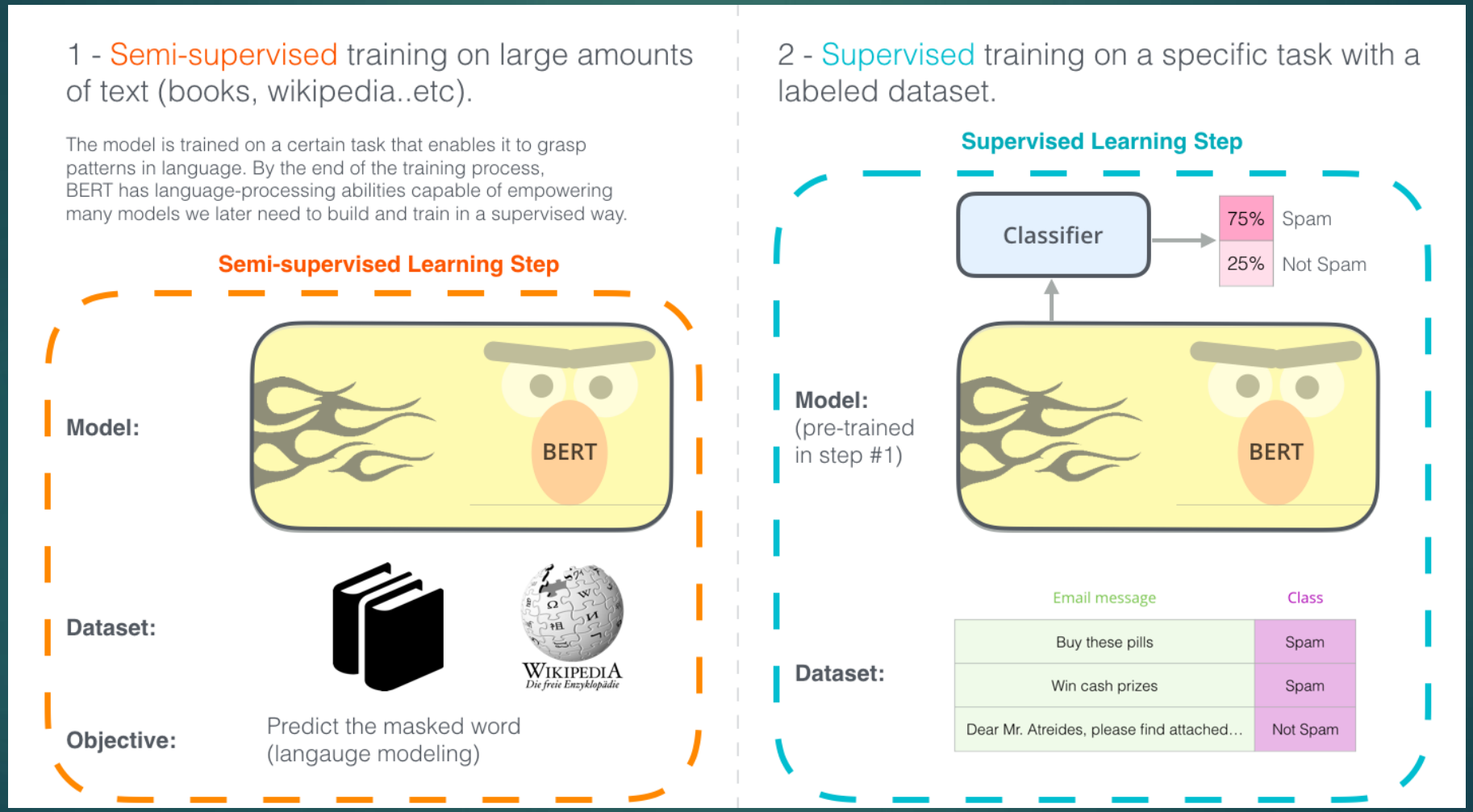
- Pretrained deep neural network language model은 다양한 language understanding tasks에서 좋은 성과
  - ELMo, GPT, BERT and XLNet etc.
- 하지만 mobile이나 edge devices에서 사용하기에 모델들의 크기가 문제
  - 특히 embedding matrix (단어 개수 * embedding dimension)가 메모리의 대부분을 차지
- Knowledge distillation techniques과 같은 network 압축 기술의 필요성

Background: language modeling





Background: language modeling

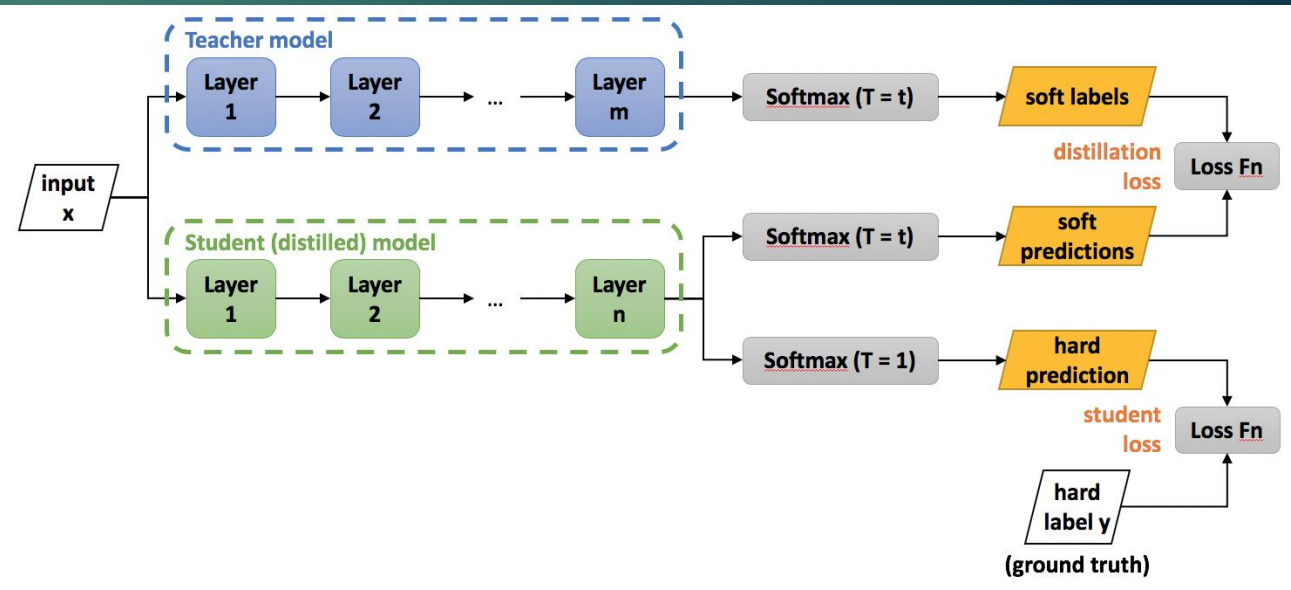


http://jalammar.github.io/illustrated-bert/

Background: Knowledge distillation

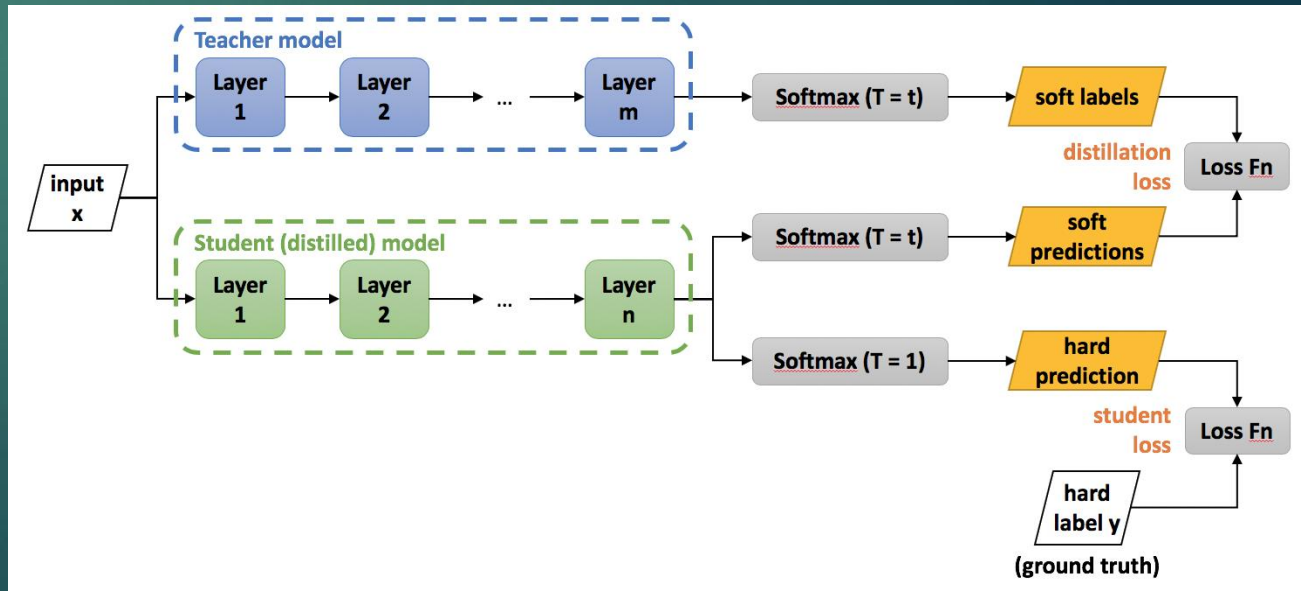
- Knowledge distillation은 Pretrain 된 큰 모델 (혹은 앙상블)을 모방하도록 train하는 모델 압축 방법론 (generalized by Hinton et al., 2015)

- Teacher model의 logits을 softmax한 확률 분포를 모방
  - 그러나 보통의 경우에는 softmax를 취한 값이 정답 레이블을 제외하면 0에 매우 가까움
  - Temperature 개념을 도입하여 해결

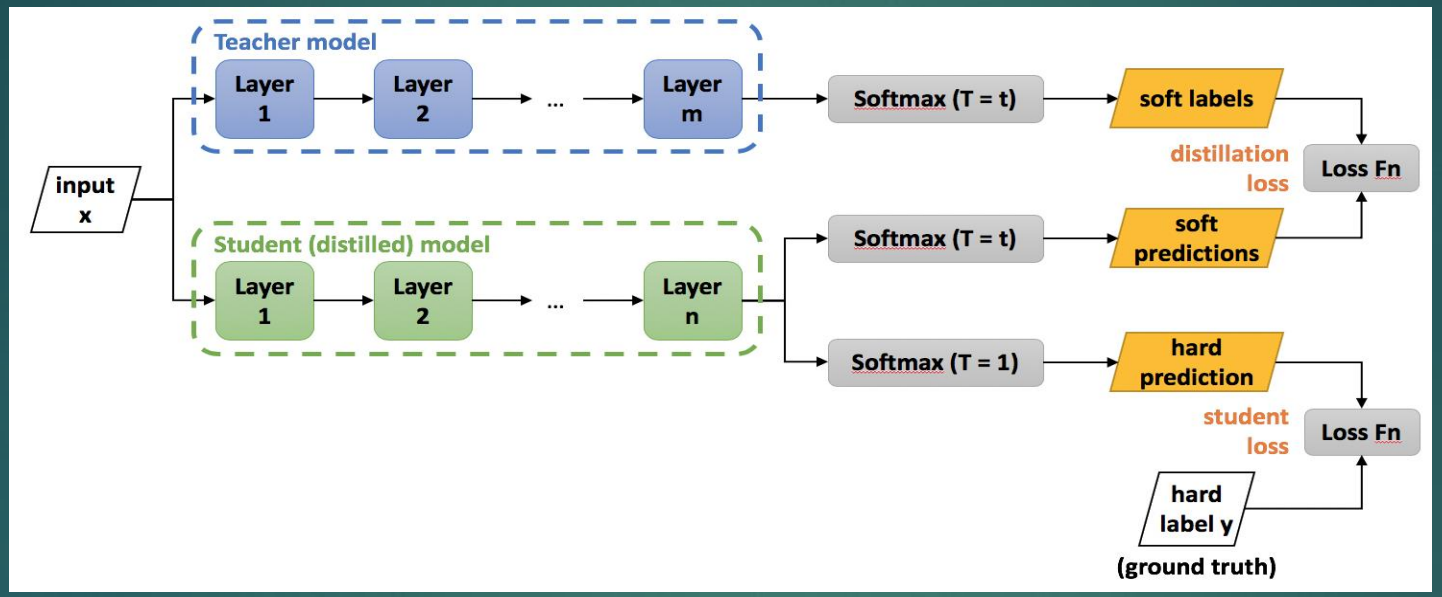


Background: Knowledge distillation

- $p_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}$
- T=1 softmax function
- T>1 probability distribution generated by the softmax function becomes softer



Background: Knowledge distillation



$$\mathcal{L}(x;W) = \alpha * \mathcal{H}(y, \sigma(z_s; T=1)) + \beta * \mathcal{H}(\sigma(z_t; T=\tau), \sigma(z_s, T=\tau))$$

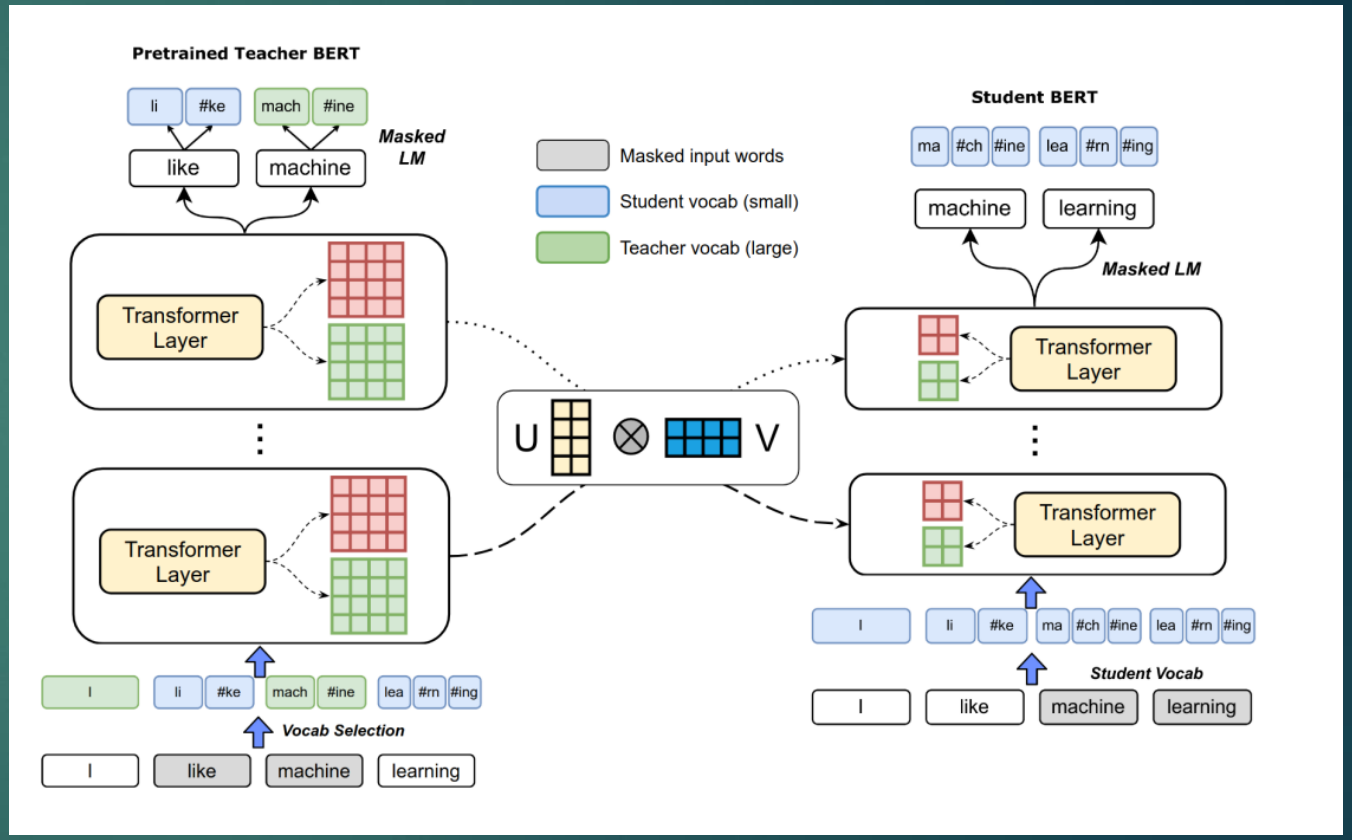
where x is the input, W are the student model parameters, y is the ground truth label, $\mathcal{H}$ is the cross-entropy loss function, σ is the softmax function parameterized by the temperature T, and α and β are coefficients. z_s and z_t are the logits of the student and teacher respectively.

ABSTRACT

- Knowledge distillation techniques은 Original teacher model과 vocabularies가 다르기 때문에 student model을 생산하는데 있어서 비효율적이다.
  - student vocabulary에 최적화된 word embedding을 얻기 위해 dual-training mechanism을 차용하여 teacher 와 student 모델을 동시에 학습
  - 이 approach와 learning shared projection matrices를 결합하여 layer-wise knowledge를 transfer
  - 우리의 방법은 BERT_base 와 비교하여 downstream한 task에서 조금의 성능 하락은 있지만 60배 빠르며 LM의 결과는 7MB이하의 size

METHODOLOGY

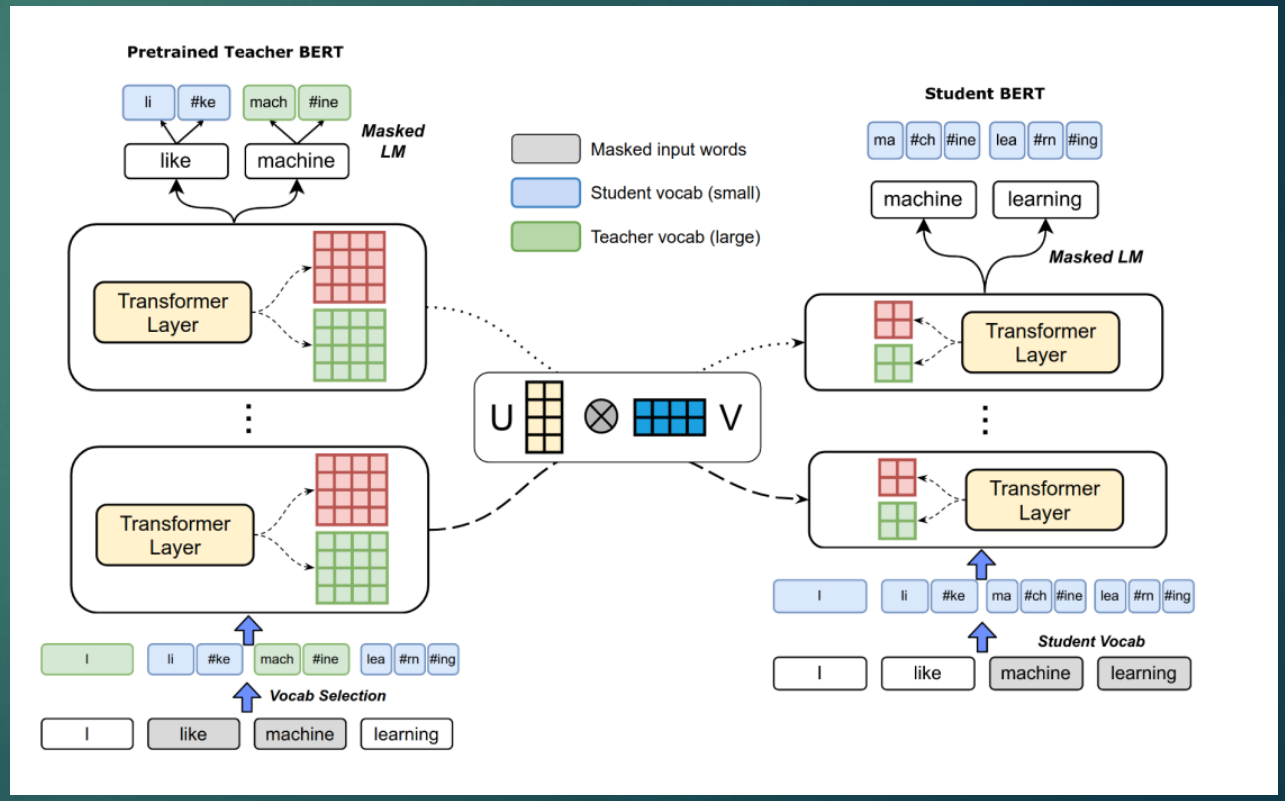
- 우리의 knowledge distillation은 WordPiece token 개수를 줄이는 데 중점
- Teacher BERT
  - 12-layer uncased BERT_base
  - 30552 WordPiece tokens
  - 768d embedding and hidden
  - Parameter denoted by θ_t
- Student BERT
  - 12-layer BERT
  - 4928 WordPiece tokens
  - 48d embedding and hidden
  - Parameter denoted by θ_s



- sub-word units obtained by applying a greedy segmentation algorithm to the training corpus
  - a desired number (say, D)

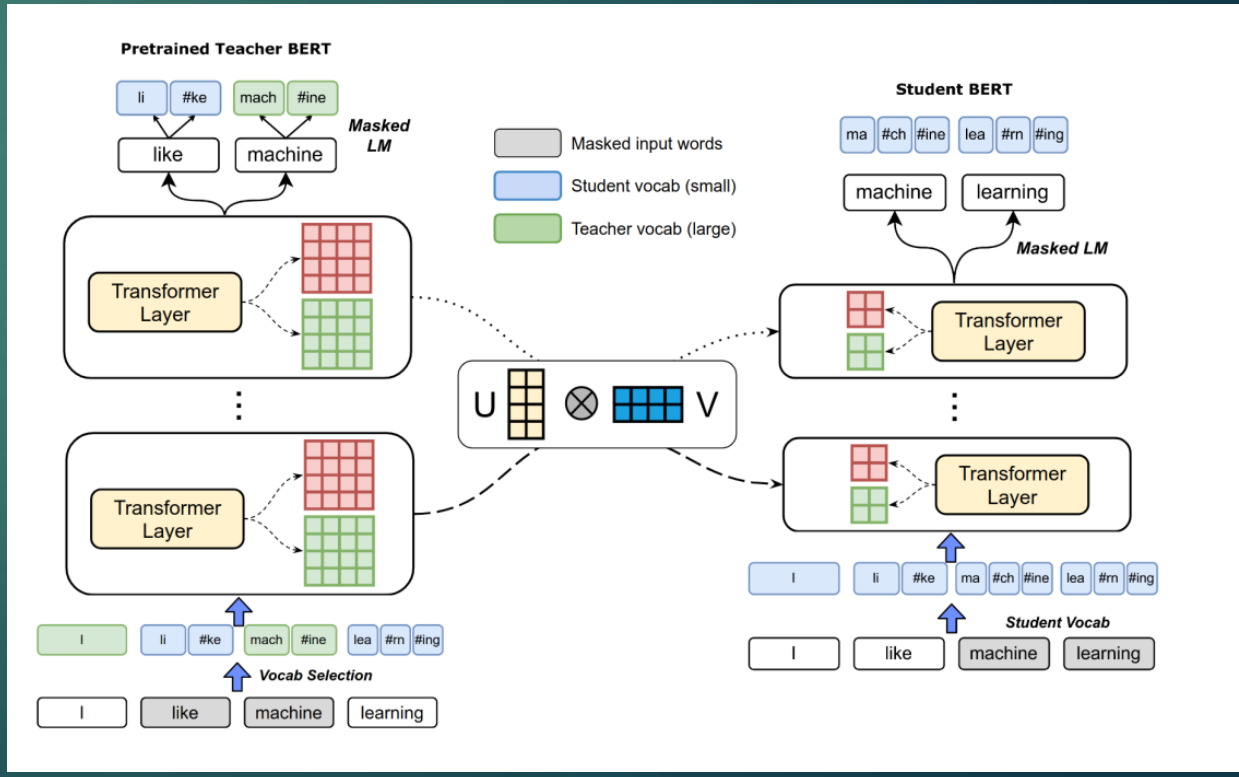
METHODOLOGY

- original training objective를 학습
  - masked language modeling and next sentence prediction
- 그러나, student vocabulary는 teacher vocabular의 완전한 subset이 아님
  - 같은 단어를 다르게 토큰화
  - 결국 최종적인 output이 서로 align X
  - 제안하는 두 가지 기술들이 필요



METHODOLOGY: DUAL TRAINING

- Teacher model의 input word의 몇몇 부분을 student vocabulary로 대체
  - mix the teacher and student vocabularies by randomly selecting
  - words ‘I’ and ‘machine’ are segmented using the teacher vocabulary (in green)
  - while ‘like’ and ‘learning’ are segmented using the student vocabulary (in blue)

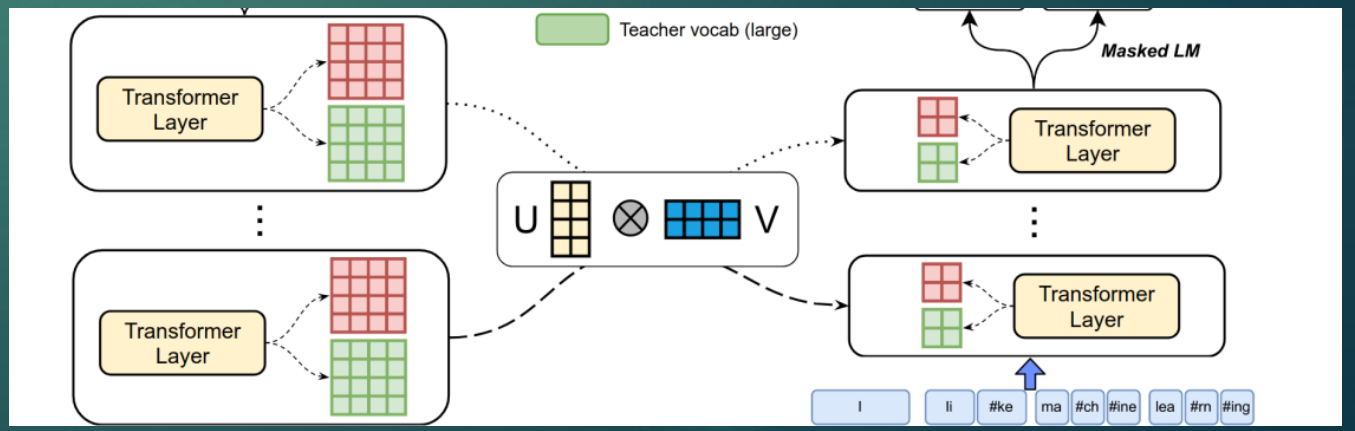


METHODOLOGY: SHARED PROJECTIONS

- Teacher 모델의 output만을 이용해 학습하는 것은 부족
  - layer 중간중간의 prediction을 사용하여 학습이 필요
- θ_t의 각 trainable variable를 상응하는 θ_s로 projection
  - θ_t 를 Projection을 시킨 뒤 L2 loss를 적용하여 θ_s와 비슷하도록 U, V를 학습
  - Ex, θ_t 768*768 matrix => U d*768 matrix V 768*d matrix (d는 student 의 hidden size)
  - U, V는 distillation 과정에서만 필요한 파라미터이고 이후에 fine-tuning 할 때는 사용 X

$$L_p^{\downarrow} = \sum_{\theta'_t, \theta'_s \subset \theta_t, \theta_s} \left\| \mathbf{U}\theta'_t\mathbf{V} - \theta'_s \right\|^2$$

$$L_p^{\uparrow} = \sum_{\theta'_t, \theta'_s \subset \theta_t, \theta_s} \left\| \theta'_t - \mathbf{V}\theta'_s\mathbf{U} \right\|^2$$



OPTIMIZATION OBJECTIVE

- projection loss + two masked language modeling cross-entropy losses
  - masked language modeling cross-entropy losses for the student as well as the teacher models
  - teacher model is trained with dual-vocabulary inputs and is not static.

$$L_{ce} = \sum_i \left(\sum_{c \subset C} \left[\mathbb{1}_{[y_i]=c} \log P(y_i = c|\theta_s) \right] + \sum_{c \subset C} \left[\mathbb{1}_{[y_i]=c} \log P(y_i = c|\theta_t) \right] \right)$$

$$L_{final} = L_p + \epsilon \times L_{ce}$$

EXPERIMENTS & RESULTS

- Student BERT 모델의 학습을 위해 원래 BERT 모델의 학습과 동일한 학습을 진행
  - BooksCorpus, English Wikipedia data 학습
  - 다음 문장 예측을 위한 loss가 성능에 약간의 영향을 미치기 때문에 최종적인 loss를 계산하기 위해 Masked language modeling만 사용
  - 32 TPU 코어로 학습
  - Student BERT 모델 차원에 따라 2-4 일이 소요

EXPERIMENTS & RESULTS

Model Type	Vocab Size	Hidden Dim	# Params	Model Size (MB)	FLOPS ratio
$\text{BERT}_{\text{DISTILLED}}$	4928	48	1,775,910	6.8	1.3%
		96	5,665,926	22	1.32%
		192	19,169,094	73	4.49%
$\text{BERT}_{\text{BASE}}$	30522	768	110,106,428	420	100%

Table 1: A summary of our student models' sizes compared to $\text{BERT}_{\text{BASE}}$. #Params indicates the number of parameters in the student model, model size is measured in megabytes, and FLOPS ratio measures the relative ratio of floating point operations required for inference on the model.

EXPERIMENTS & RESULTS

Model ↓ Hidden dimension →	**48**	**96**	**192**
NoKD Baseline	28.83	33.70	40.01
DualTrain	29.51	34.59	40.58
DualTrain + SharedProjDown	29.55	34.71	40.59
DualTrain + SharedProjUp	**29.89**	**34.74**	**40.64**

Table 2: Masked language modeling task accuracy for the distilled student models and a fine-tune-from-scratch baseline. We observe consistently better performance for our proposed approaches.

EXPERIMENTS & RESULTS

Model	Hidden Dim	Vocab Size	Compress Factor	MRPC (F1/Acc)	MNLI-m (Acc)	MNLI-mm (Acc)	SST-2 (Acc)
Teacher $BERT_{BASE}$	768	30522	1x	88.5/84.3	84.0	82.8	93.5
PKD, 6 layers (Sun et al., 2019)	768	30522	1.64x	85.0/79.9	81.5	81.0	92.0
PKD, 3 layers (Sun et al., 2019)			2.40x	80.7/72.5	76.7	76.3	87.5
NoKD Baseline	192	4928	5.74x	82.6/74.1	77.4	76.5	87.1
DualTrain				82.5/76.6	78.1	77.3	**88.4**
DualTrain + SharedProjDown				83.6/76.9	**78.2**	**77.7**	**88.4**
DualTrain + SharedProjUp				**84.9/78.5**	77.5	76.7	88.0
NoKD Baseline	96	4928	19.41x	84.6/77.3	76.2	75.1	85.4
DualTrain				**86.1/80.5**	76.1	74.7	85.4
DualTrain + SharedProjDown				83.7/77.5	**76.5**	**75.2**	**85.6**
DualTrain + SharedProjUp				84.9/78.1	76.4	**75.2**	84.7
NoKD Baseline	48	4928	61.94x	76.3/66.1	70.9	70.2	79.5
DualTrain				77.5/66.8	70.6	69.9	79.8
DualTrain + SharedProjDown				78.0/68.2	**71.3**	70.4	80.0
DualTrain + SharedProjUp				**79.3/68.6**	71.0	**70.8**	**82.2**

Table 3: Results of the distilled models, the teacher model and baselines on the downstream language understanding task test sets, obtained from the GLUE server, along with the size parameters and compression ratios of the respective models compared to the teacher $BERT_{BASE}$. MNLI-m and MNLI-mm refer to the genre-matched and genre-mismatched test sets for MNLI.