Multi-Task Deep Neural Networks for Natural Language Understanding

모두연 NLP Bootcamp
김 성 운

Contents

Introduction

- Text의 vector space representation을 배우는 두 가지 방법

1. Multi-Task learning

   - 하나의 모델을 활용하여 다양한 task를 지도 학습
   - DNN을 활용한 개별적인 지도 학습은 데이터 양이 많이 상당히 많이 필요하여 항상 적용하기 어려우나, MTL은 관련된 task의 다양한 데이터를 활용하여 효율적인 학습이 가능
   - 모델이 특정 task에 over-fitting 되지 않도록 하는 regularization 효과

2. Language model pre-training

   - 다수의 데이터를 활용한 비지도 학습(ELMO, GPT, BERT)
   - MT-DNN은 masked word prediction, next sentence prediction를 이용해 pre-training 되어 있는 BERT에 기반 (MTL를 적용한 fine-tunning)

Task

1. Single Sentence Classification

- 하나의 문장이 주어졌을 때 문장의 class를 분류하는 Task
- CoLA: 영어 문장이 문법적으로 이상 없는지 분류
- SST-2: 영화 Review 문장의 감정 분류

0	*	the box contained the ball from the tree .
0	*	the tube was escaped by gas .
1		water bubbled up out of the kettle .
1		the tub leaked water .

CoLA dataset examples

Task

2. Text Similarity

- 문장 pair가 주어졌을 때, 유사도를 예측하는 Regression Task
- STS-B: 문장 간의 유사도 scoring

0.600	A man is crying.	A woman is dancing.
2.600	The lady cracked an egg into a bowl.	The man is cracking eggs into a bowl.
5.000	A band is performing on a stage.	A band is playing onstage.
4.600	Elephants are walking down a trail.	A herd of elephants are walking along a trail.
5.000	A man is playing a guitar.	A man plays the guitar.

STS-B dataset examples

Task

3. Pairwise Text Classification

- 문장 pair가 주어졌을 때, 문장의 관계를 분류하는 Task

(RTE, MNLI, QQP, MRPC)

Met my first girlfriend that way.	FACE-TO-FACE **contradiction** C C N C	I didn't meet my first girlfriend until later.
8 million in relief in the form of emergency housing.	GOVERNMENT **neutral** N N N N	The 8 million dollars for emergency housing was still not enough to solve the problem.
Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community.	LETTERS **neutral** N N N N	All of the children love working in their gardens.
At 8:34, the Boston Center controller received a third transmission from American 11	9/11 **entailment** E E E E	The Boston Center controller got a third transmission from American 11.
I am a lacto-vegetarian.	SLATE **neutral** N N E N	I enjoy eating cheese too much to abstain from dairy.
someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny	TELEPHONE **contradiction** C C C C	No one noticed and it wasn't funny at all.

Table 1: Randomly chosen examples from the development set of our new corpus, shown with their genre labels, their selected gold labels, and the validation labels (abbreviated E, N, C) assigned by individual annotators.

MNLI dataset examples

Task

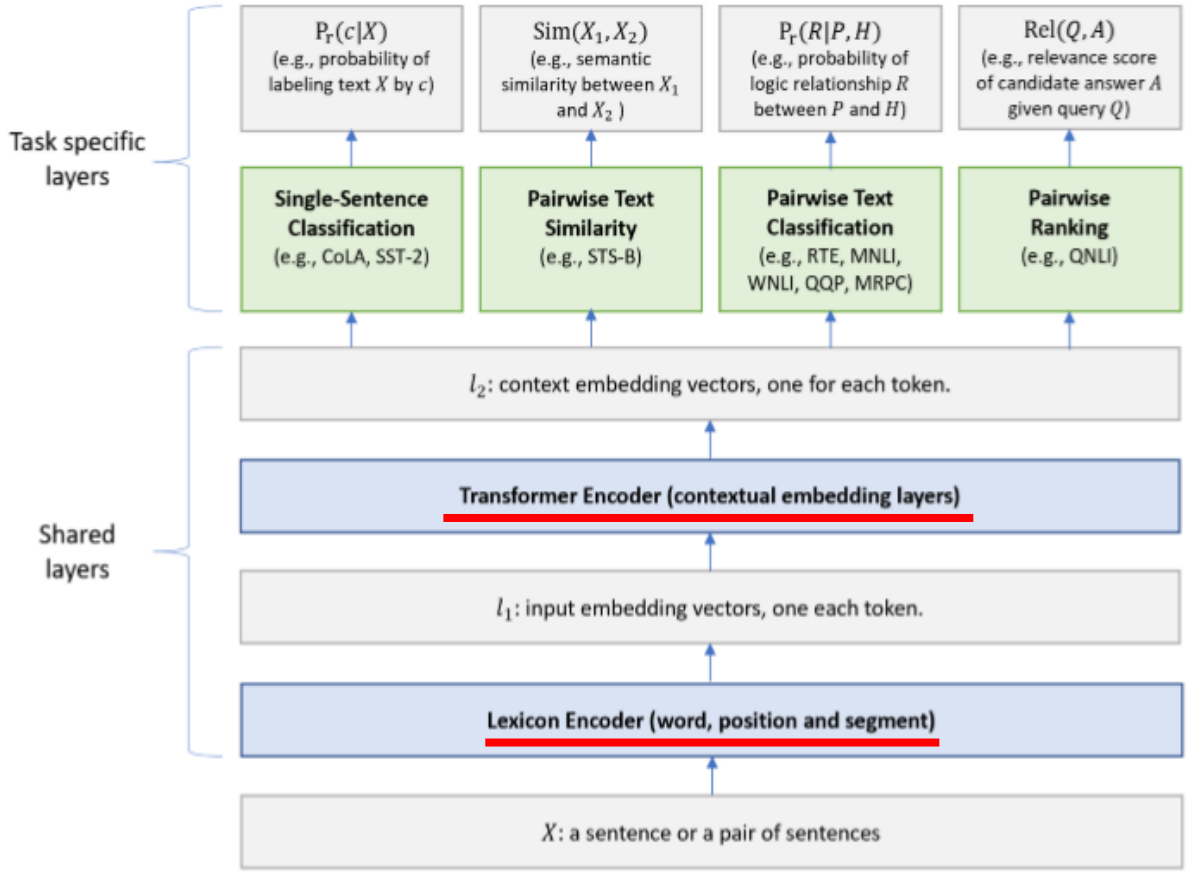
4. Relevance Ranking

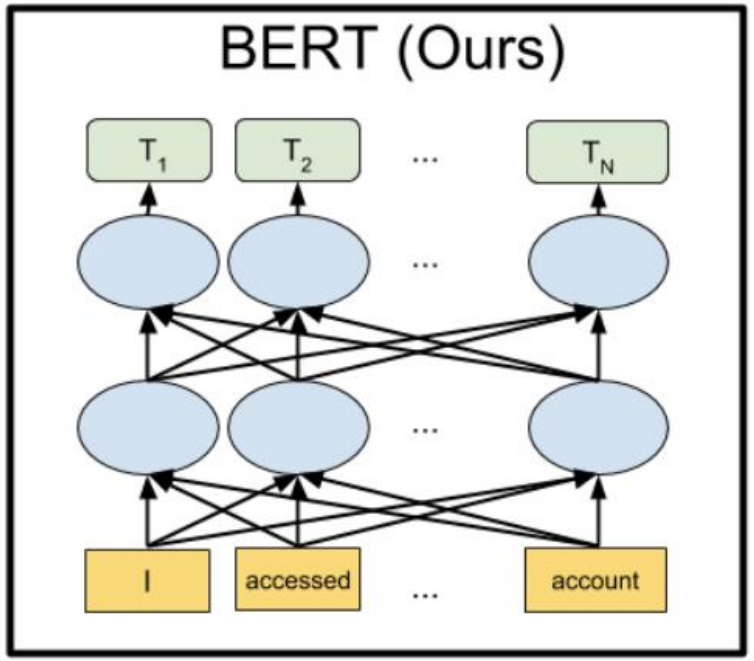
- 질문 문장과 지문이 주어졌을 때, 지문 중 정답이 있는 문장을 Ranking을 통해 찾는 Task
- QNLI: 질문과 해당 지문 중 한 문장이 쌍으로 주어졌을 때 해당 지문 문장에 질문의 답이 있는지 여부를 분류
- MT-DNN에서는 지문 candidate set을 구성하여 정답이 있을 가능성을 Scoring 한 후 가장 높은 점수에 해당하는 지문을 선택

What came into force after the new constitution was herald?
As of that day, the new constitution heralding the Second Republic came into force. (entailment)
What is the first major city in the stream of the Rhine?
The most important tributaries in this area are the Ill below of Strasbourg, the Neckar in Mannheim and the Main across from Mainz. (not entailment)

QNLI dataset examples

Model





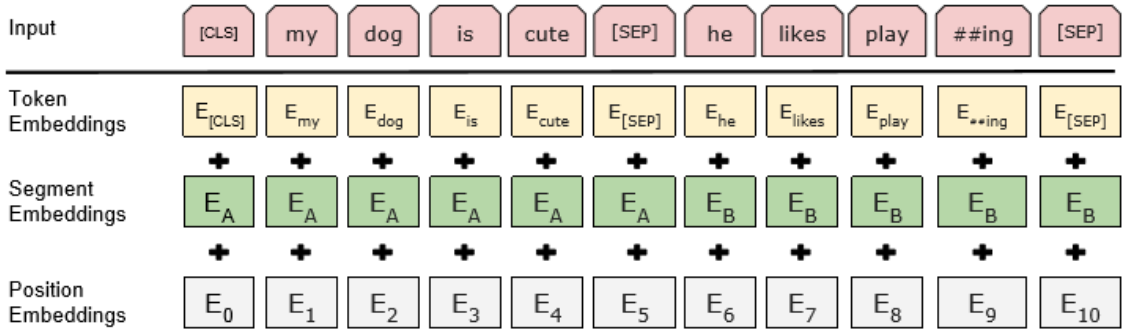


Figure 1: Architecture of the MT-DNN model for representation learning. The lower layers are shared across all tasks while the top layers are task-specific. The input X (either a sentence or a pair of sentences) is first represented as a sequence of embedding vectors, one for each word, in l_1. Then the Transformer encoder captures the contextual information for each word and generates the shared contextual embedding vectors in l_2. Finally, for each task, additional task-specific layers generate task-specific representations, followed by operations necessary for classification, similarity scoring, or relevance ranking.

Model

1. Single-Sentence Classification Output

   The probability that X is labeled as class c (i.e., the sentiment) is predicted by a logistic regression with softmax:

$$P_r(c|X) = \mathbf{softmax}(\mathbf{W}_{SST}^{\top} \cdot \mathbf{x})$$

2. Text Similarity Output

$$\mathbf{Sim}(X_1, X_2) = \mathbf{w}_{STS}^{\top} \cdot \mathbf{x}$$

Model

3. Pairwise Text Classification Output

The design of the output module follows the answer module of the **stochastic answer network (SAN)** (Liu et al., 2018a), a state-of-the-art neural NLI model. SAN's answer module uses multi-step reasoning. Rather than directly predicting the entailment given the input, it maintains a state and iteratively refines its predictions.

premise: *"If you need this book, it is probably too late unless you are about to take an SAT or GRE.",*
hypothesis: *"It's never too late, unless you're about to take a test."*

To predict the correct relation between these two sentences, the model needs to first infer that "SAT or GRE" is a "test", and then pick the correct relation, e.g., contradiction.

Model

3. Pairwise Text Classification Output

- stochastic answer network (SAN)을 이용한 multi-step reasoning

The SAN answer module works as follows. We first construct the working memory of premise P by concatenating the contextual embeddings of the words in P, which are the output of the transformer encoder, denoted as $\mathbf{M}^p \in \mathbb{R}^{d \times m}$, and similarly the working memory of hypothesis H, denoted as $\mathbf{M}^h \in \mathbb{R}^{d \times n}$. Then, we perform K-step reasoning on the memory to output the relation label, where K is a hyperparameter. At the beginning, the initial state $\mathbf{s}^0$ is the summary of $\mathbf{M}^h$: $\mathbf{s}^0 = \sum_j \alpha_j \mathbf{M}_j^h$, where $\alpha_j = \frac{\exp(\mathbf{w}_1^\top \cdot \mathbf{M}_j^h)}{\sum_i \exp(\mathbf{w}_1^\top \cdot \mathbf{M}_i^h)}$. At time step k in the range of $\{1, 2, , K-1\}$, the state is defined by $\mathbf{s}^k = \text{GRU}(\mathbf{s}^{k-1}, \mathbf{x}^k)$. Here, $\mathbf{x}^k$ is computed from the previous state $\mathbf{s}^{k-1}$ and memory $\mathbf{M}^p$: $\mathbf{x}^k = \sum_j \beta_j \mathbf{M}_j^p$ and $\beta_j = \text{softmax}(\mathbf{s}^{k-1} \mathbf{W}_2^\top \mathbf{M}^p)$. A one-layer classifier is used to determine the relation at each step k:

$$P_r^k = \text{softmax}(\mathbf{W}_3^\top [\mathbf{s}^k; \mathbf{x}^k; |\mathbf{s}^k - \mathbf{x}^k|; \mathbf{s}^k \cdot \mathbf{x}^k]). \quad (3)$$

At last, we utilize all of the K outputs by averaging the scores:

$$P_r = \text{avg}([P_r^0, P_r^1, ..., P_r^{K-1}]). \quad (4)$$

Model

4. Relevance Ranking Output

For a given Q, we rank all of its candidate answers based on their relevance scores computed using Equation 5.

$$\mathrm{Rel}(Q, A) = g(\mathbf{w}_{QNLI}^{\top} \cdot \mathbf{x}), \qquad (5)$$

Model

Training

Algorithm 1: Training a MT-DNN model.

Initialize model parameters Θ randomly.
Pre-train the shared layers (i.e., the lexicon encoder and the transformer encoder).
Set the max number of epoch: $epoch_{max}$.
//Prepare the data for T tasks.
for t *in* $1, 2, ..., T$ **do**
    Pack the dataset t into mini-batch: D_t.
end
for $epoch$ *in* $1, 2, ..., epoch_{max}$ **do**
    1. Merge all the datasets:
    $D = D_1 \cup D_2 ... \cup D_T$
    2. Shuffle D
    for b_t *in* D **do**
        //b_t is a mini-batch of task t.
        3. Compute loss : $L(\Theta)$
        $L(\Theta)$ = Eq. 6 for classification
        $L(\Theta)$ = Eq. 7 for regression
        $L(\Theta)$ = Eq. 8 for ranking
        4. Compute gradient: $\nabla(\Theta)$
        5. Update model: $\Theta = \Theta - \epsilon\nabla(\Theta)$
    end
end

Result

- Summary of the three benchmarks

Corpus	Task	#Train	#Dev	#Test	#Label	Metrics
Single-Sentence Classification (GLUE)						
CoLA	Acceptability	8.5k	1k	1k	2	Matthews corr
SST-2	Sentiment	67k	872	1.8k	2	Accuracy
Pairwise Text Classification (GLUE)						
MNLI	NLI	393k	20k	20k	3	Accuracy
RTE	NLI	2.5k	276	3k	2	Accuracy
WNLI	NLI	634	71	146	2	Accuracy
QQP	Paraphrase	364k	40k	391k	2	Accuracy/F1
MRPC	Paraphrase	3.7k	408	1.7k	2	Accuracy/F1
Text Similarity (GLUE)						
STS-B	Similarity	7k	1.5k	1.4k	1	Pearson/Spearman corr
Relevance Ranking (GLUE)						
QNLI	QA/NLI	108k	5.7k	5.7k	2	Accuracy
Pairwise Text Classification						
SNLI	NLI	549k	9.8k	9.8k	3	Accuracy
SciTail	NLI	23.5k	1.3k	2.1k	2	Accuracy

Result

Model	CoLA 8.5k	SST-2 67k	MRPC 3.7k	STS-B 7k	QQP 364k	MNLI-m/mm 393k	QNLI 108k	RTE 2.5k	WNLI 634	AX	Score
BiLSTM+ELMo+Attn [1]	36.0	90.4	84.9/77.9	75.1/73.3	64.8/84.7	76.4/76.1	-	56.8	65.1	26.5	70.5
Singletask Pretrain Transformer [2]	45.4	91.3	82.3/75.7	82.0/80.0	70.3/88.5	82.1/81.4	-	56.0	53.4	29.8	72.8
GPT on STILTs [3]	47.2	93.1	87.7/83.7	85.3/84.8	70.1/88.1	80.8/80.6	-	69.1	65.1	29.4	76.9
$\text{BERT}^{4}_{\text{LARGE}}$	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7/85.9	92.7	70.1	65.1	39.6	80.5
$\text{MT-DNN}_{\text{no-fine-tune}}$	58.9	94.6	**90.1/86.4**	89.5/88.8	**72.7/89.6**	86.5/85.8	**93.1**	79.1	65.1	39.4	81.7
MT-DNN	**62.5**	**95.6**	**91.1/88.2**	**89.5/88.8**	**72.7/89.6**	**86.7/86.0**	**93.1**	**81.4**	65.1	**40.3**	**82.7**
Human Performance	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0/92.8	91.2	93.6	95.9	-	87.1

Table 2: GLUE test set results scored using the GLUE evaluation server. The number below each task denotes the number of training examples. The state-of-the-art results are in **bold**, and the results on par with or pass human performance are in **bold**. MT-DNN uses $\text{BERT}_{\text{LARGE}}$ to initialize its shared layers. All the results are obtained from https://gluebenchmark.com/leaderboard on February 25, 2019. Model references: [1]:(Wang et al., 2018) ; [2]:(Radford et al., 2018); [3]: (Phang et al., 2018); [4]:(Devlin et al., 2018).

- 다양한 task에서 Human performance를 능가
- 82.7% score를 달성하며, BERT에 비해 2.2% 향상
- Dataset이 적은 task의 경우 상대적으로 높은 성능 향상

Result

Model	MNLI-m/mm	QQP	RTE	QNLI (v1/v2)	MRPC	CoLa	SST-2	STS-B
$\text{BERT}_{\text{LARGE}}$	86.3/86.2	91.1/88.0	71.1	90.5/92.4	89.5/85.8	61.8	93.5	89.6/89.3
ST-DNN	86.6/86.3	91.3/88.4	72.0	96.1/-	89.7/86.4	-	-	-
MT-DNN	**87.1/86.7**	**91.9/89.2**	**83.4**	**97.4/92.9**	**91.0/87.5**	**63.5**	**94.3**	**90.7/90.6**

Table 3: GLUE dev set results. The best result on each task is in **bold**. The Single-Task DNN (ST-DNN) uses the same model architecture as MT-DNN. But its shared layers are the pre-trainedBERT model without being refined via MTL. We fine-tuned ST-DNN for each GLUE task using task-specific data. There have been two versions of the QNLI dataset. V1 is expired on January 30, 2019. The current version is v2. MT-DNN use $\text{BERT}_{\text{LARGE}}$ as their initial shared layers.

- Task Specific Layer만 바꾸어 수행한 결과 SAN를 적용한 task들은 성능 개선이 미미
- Relevance Ranking Output을 적용한 QNLI는 높은 성능 향상

Result

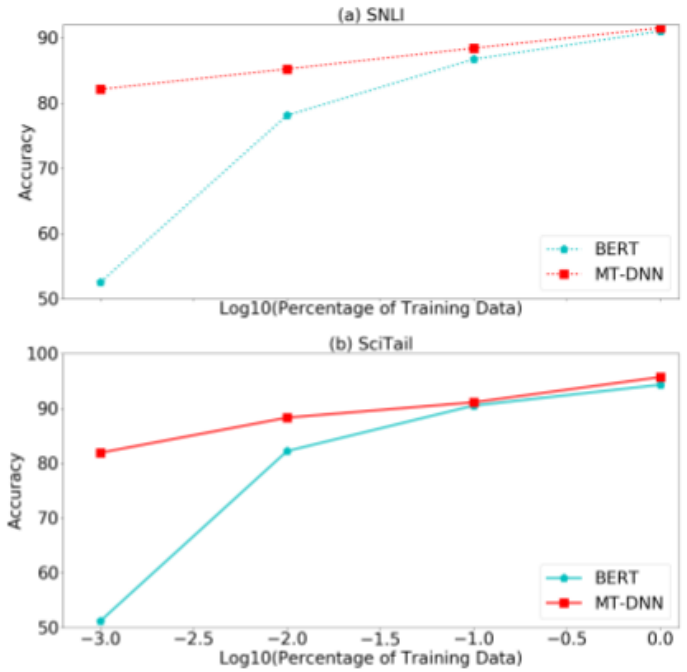


Figure 2: Domain adaption results on SNLI and SciTail development datasets using the shared embeddings generated by MT-DNN and BERT, respectively. Both MT-DNN and BERT are fine-tuned based on the pre-trained $\text{BERT}_{\text{BASE}}$. The X-axis indicates the amount of domain-specific labeled samples used for adaptation.

Model	0.1%	1%	10%	100%
SNLI Dataset (Dev Accuracy%)				
#Training Data	549	5,493	54,936	549,367
BERT	52.5	78.1	86.7	91.0
MT-DNN	82.1	85.2	88.4	91.5
SciTail Dataset (Dev Accuracy%)				
#Training Data	23	235	2,359	23,596
BERT	51.2	82.2	90.5	94.3
MT-DNN	81.9	88.3	91.1	95.7

- BERT에 비해 보다 효과적인 domain adaptation

Conclusion

1. MT-DNN을 이용하여 다양한 GLUE task에서 SOTA
2. Adversarial attacks에 강건한지 확인 필요

Premise/Hypothesis	Label
The man is holding a saxophone The man is holding an electric guitar	contradiction[1]
A little girl is very sad. A little girl is very unhappy.	entailment
A couple drinking wine A couple drinking champagne	neutral

Table 1: Examples from the new test set.

the event of a man holding both instruments??