Meeting Note

Author: Y.K. Lin
Date: 2024/09/23 – 2024/09/27

Weekly Summary
Improving language understanding with unsupervised learning [1, Re. 1].

Plan for Next Week
Masked Autoencoders Are Scalable Vision Learners (arXiv: 2111.06377).

Details
For the tasks of natural language understanding, the labeled data is scarce, this makes it challenging to train models using supervised approach. On the other hand, the unlabeled data is plentiful, besides, the experimental results shows that models using unsupervised learning can provide significant performance boost. The use of pre-trained word embeddings to improve the models' performance on natural language processing (NLP) tasks can be considered as the evidence.

Thus, Radford et al. [1] proposed a semi-supervised approach for NLP tasks which consists of unsupervised pre-training and supervised fine-tuning. This approach shows effectiveness on various NLP benchmarks. However, the authors also mentioned about the two challenges to pre-train model on unlabeled text data: (1) there are limited researches on what type of training objectives (e.g. language modeling [5], machine translation [6], etc.) are optimal for learning text representation that are useful for downstream tasks; (2) there is no general opinion on how to efficiently transfer the learned representation to the downstream tasks.

For the model architecture, the authors use 12-layer decoder-only Transformer, which has been shown to have excellent performance on NLP tasks such as machine translation and document generation. There are other reasons for the authors to choose Transformer, one is that Transformer provides more structured memory for handling long-term dependencies between text comparing to the other architecture such as recursive neural network (RNN); the other is that the adaptation of Transformer and traversal-style [4, Re. 2] approach enables to fine-tuning with minimal changes on the architecture of pre-trained model.

As we mentioned before, the training process is divided into two steps, and the first is the unsupervised pre-training. The authors choose the language modeling as the objective to train the model [Re. 3], which is basically to predict the next token based on a given sequence and

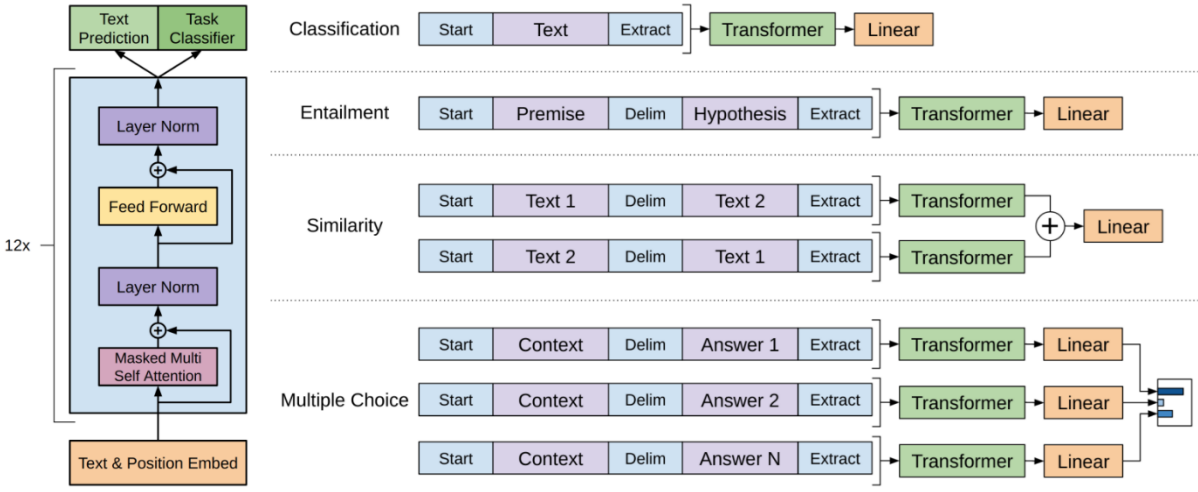

Figure 1. (left) Transformer architecture and training objectives. (right) Input transformation for various fine-tuning tasks.

can be represented as follow:

$$L_1(\mathcal{U}) = \sum_i log\ P(u_i|u_{i-k}, \ldots, u_{i-1}; \Theta)$$

where $\mathcal{U} = u_1, \ldots, u_n$ denotes the token sequence, P is the neural model with parameters Θ which are trained with stochastic gradient descent (SGD) [7], and k denotes the size of window. The second step is supervised fine-tuning. The authors first feed the labeled dataset C into the pre-trained model to obtain representation h_l^m, and then forward it to a linear layer with parameters W_y followed by a softmax function:

$$L_1(\mathcal{C}) = \sum_{(x,y)} log\ P(y|x^1, \ldots, x^m)$$

where

$$P(y|x^1, \ldots, x^m) = softmax(h_l^m W_y)$$

In the dataset C, each instance consists of an input token sequence $x^1, \ldots, x^m$ and a label y. Finally, we obtain the training objective as follow:

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

where λ is a hyperparameter set to 0.5 in this paper.

Sine the model is trained on contiguous sequence of token, structed fine-tuning text data, such as question-answering pairs, must be transformed to fit the model's input format. Unlike previous works that typically employ task-specific architectures for applying models across different neural language tasks, the authors propose a more streamlined approach focused on input transformation. As shown on the left in Figure 1, input data with different structures are

converted into an ordered sequences which can be feed into the pre-train model directly. This transformation minimizes the need for modifications across different tasks.

For the unsupervised language modeling, the authors use BooksCorpus [8] as the dataset, and for the supervised fine-tuning, they use several datasets shown at Table 1 for totally four tasks as follows: (1) Neural Language Inference (NLI), (2) Question Answering (QA), (3) Semantic Similarity, and (4) Text Classification.

As Table 2, 3, and 4 show, the proposed approach achieves the state-of-the-art performance on various tasks.

Table 1. Datasets for different fine-tuning tasks.

Task	**Datasets**
Neural Language Inference	SNLI [9], MultiNLI [10], RTE [11], SciTail [12]
Question Answering	RACE [13], Story Cloze [14]
Sentence Similarity	MASR Paraphrase Corpus [15], Quora Question Pairs [16], STS Benchmark [17]
Text Classification	Stanford Sentiment Treebank-2 [18], CoLA [19]

Table 2. Experimental results on NLI tasks.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo (5x)	-	-	89.3	-	-	-
CAFE (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network (3x)	80.6	80.1	-	-	-	-
CAFE	78.7	77.9	88.5	83.3		
GenSen	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (ours)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Table 3. Experimental results on QA tasks.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip	76.5	-	-	-
Hidden Coherence Model	77.6	-	-	-
Dynamic Fusion Net (9x)	-	55.6	49.4	51.2
BiAttention MRU (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

Table 4. Experimental results on semantic similarity.

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM	-	**93.2**	-	-	-	-
TF-KLD	-	-	**86.0**	-	-	-
ECNU (mixed ensemble)	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Remarks

[Re. 1]

As the paper "An Introduction to Convolutional Neural Networks" [2] is comparatively elementary, we have adjusted this week's reading schedule accordingly.

The model proposed in this week's paper is also known as Generative Pre-trained Transformer (GPT).

[Re. 2]

A traversal-style approach refers to a method of converting structured input into sequences, allowing the use of the same pre-trained model across different tasks without significant changes to the model's architecture. This improves reusability. For instance, in tasks like question answering or text classification, structured inputs like sentence pairs or question-answer triples are transformed into sequences and applied to existing models. An example would be pre-training on a large text corpus and then applying the model to text classification tasks, such as determining the sentiment of a product review.

[Re. 3]

The authors did not explain why they chose language modeling as the objective, despite stating that "it is unclear what type of optimization objectives are most effective at learning text representations useful for transfer".

References

[1] Radford, A. (2018). Improving language understanding by generative pre-training.

[2] O'Shea, K. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

[3] Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing*

Systems.
[4] Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiský, T., & Blunsom, P. (2015). Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.
[5] Sarzynska-Wawer, J., Wawer, A., Pawlak, A., Szymanowska, J., Stefaniak, I., Jarkiewicz, M., & Okruszek, L. (2021). Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, *304*, 114135.
[6] McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, *30*.
[7] H. Robbins and S. Monro. A stochastic approximation method. The annals of mathematical statistics, pages 400–407, 1951.
[8] Zhu, Y. (2015). Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. *arXiv preprint arXiv:1506.06724*.
[9] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. EMNLP, 2015.
[10] A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. NAACL, 2018.
[11] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The fifth pascal recognizing textual entailment challenge. In TAC, 2009.
[12] Khot, T., Sabharwal, A., & Clark, P. (2018, April). Scitail: A textual entailment dataset from science question answering. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).
[13] Lai, G., Xie, Q., Liu, H., Yang, Y., & Hovy, E. (2017). Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.
[14] Mostafazadeh, N., Roth, M., Louis, A., Chambers, N., & Allen, J. F. (2017, April). Lsdsem 2017 shared task: The story cloze test. In *2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics* (pp. 46-51). Association for Computational Linguistics.
[15] Dolan, B., & Brockett, C. (2005, January). Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.
[16] Chen, Z., Zhang, H., Zhang, X., & Zhao, L. (2018). Quora question pairs.
[17] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.
[18] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631-1642).
[19] Warstadt, A., Singh, A., & Bowman, S. R. (2019). Cola: The corpus of linguistic

acceptability (with added annotations).

Comments

This week, we improved our meeting note by:

(1) remove the rule about maximum image width. Each image should be set to a proper width.