Deep NLP and It's application on GEC

Jesse Wang

AiYunxiao Research

wangjunjie@aiyunxiao.com

Sept 2019

Overview

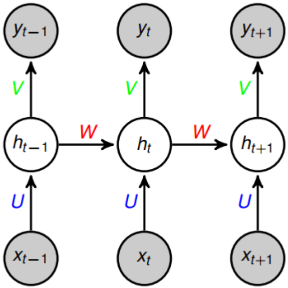
It's time to replace your RNNs with feed-forward networks!

Issues of RNNs

- Long-time dependency
- Not hardware friendly: slow speed of training and inference.

Feed-Forward Models

- Parallelization: highly parallelized structures make it possible to train a big model.
- SOTA performance, especially when more data are fed.



https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0
http://www.offconvex.org/2018/07/27/approximating-recurrent/

A general definition of attention:

- Given a set of vectors, and a vector query, attention is a technique to compute a weighted sum of the values, dependent on the query.
- We sometimes say that query attends to the values.

Several ways to compute attentions:

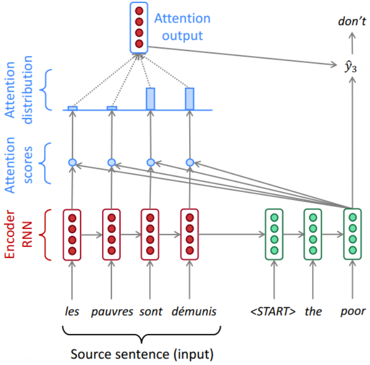
$$e_i = s^T h_i$$

$$e_i = s^T W h_i$$

$$e_i = v^T tanh(W_1 h_i + W_2 s)$$

$$\alpha^t = softmax(\mathbf{e}^t)$$

$$\alpha_t = \sum_{i=1}^{N} (\alpha_i^t \mathbf{h}_i)$$



The image is from Stanford CS224N
For a deeper understanding of attention, please read https://zhuanlan.zhihu.com/p/51747716

Self-attention

- CNNs only have local receptive fields, self-attention was added in GAN to solve the long-range dependency problem.
- Which is *Key*, *Query* or *Value*?

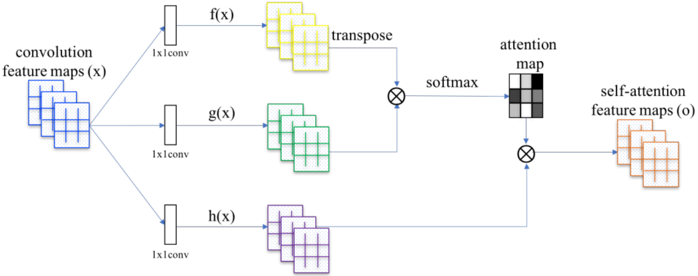


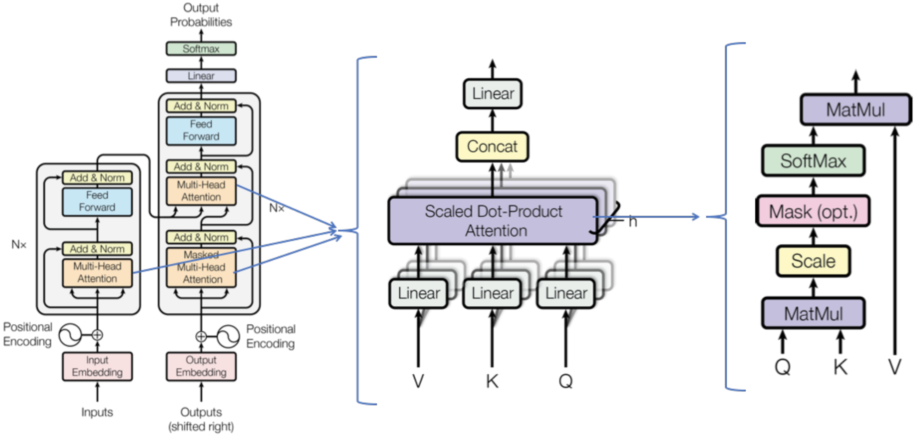
Figure 2: The proposed self-attention mechanism. The $\otimes$ denotes matrix multiplication. The softmax operation is performed on each row.

Residual connection was also added:

$$y_i = \gamma o_i + x_i \tag{1}$$

where γ is a learnable scale parameter.

Attention is all you need: the transformer

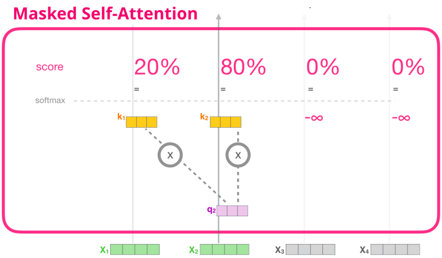


Scale Dot-Product Attention:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

Masked Multi-Head Attention

In the decoder part, texts are generated from left to right.



- Masked self-attention mechanism is used to control what a token can "see" before predicting the next token.
- In this example, x_2 can only "see" itself and x_1
- The mask do not have to be a lower triangular matrix, we will see that at XLNet.

Scores (before softmax)

0.11	0.00	0.81	0.79
0.19	0.50	0.30	0.48
0.53	0.98	0.95	0.14
0.81	0.86	0.38	0.90

Apply Attention Mask →

Masked Scores (before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Images from https://jalammar.github.io/illustrated-gpt2/

Although pre-training and fine-tuning is not new in cv tasks, it become popular in NLP tasks since BERT have obtained great success.

what researchers have done before Bert?

- word2vector: fine-tuning pre-trained word embeddings.
- pre-training General-domain LM and carefully fine-tuning the model to avoid over-fiting.

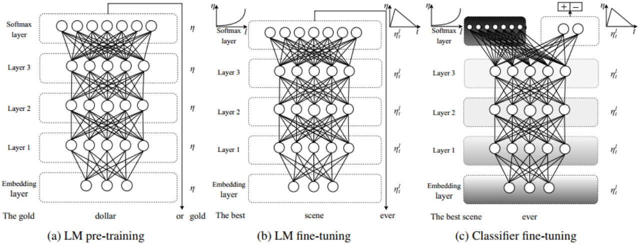


Figure 1: ULMFiT consists of three stages: a) The LM is trained on a general-domain corpus to capture general features of the language in different layers. b) The full LM is fine-tuned on target task data using discriminative fine-tuning (‘*Discr*’) and slanted triangular learning rates (STLR) to learn task-specific features. c) The classifier is fine-tuned on the target task using gradual unfreezing, ‘*Discr*’, and STLR to preserve low-level representations and adapt high-level ones (shaded: unfreezing stages; black: frozen).

Howard J , Ruder S . Universal Language Model Fine-tuning for Text Classification[J]. 2018.

Bert: Masked LM

BERT is conceptually simple and empirically powerful, which can be viewed as a transformer encoder.

- Mask 15% of all tokens
- only predict the masked words rather than reconstructing the entire input
- 80% of the time, replace with [mask] token
- 10% of the time, replace with random token
- 10% of the time, keep the word unchanged

https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

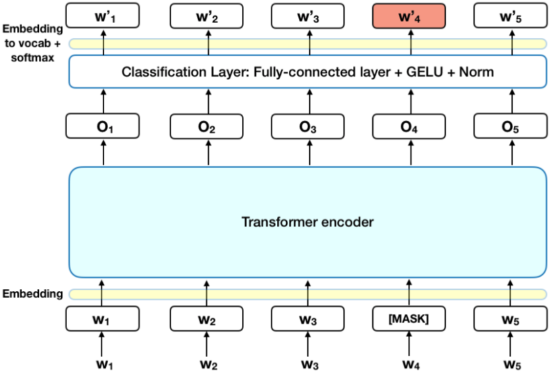
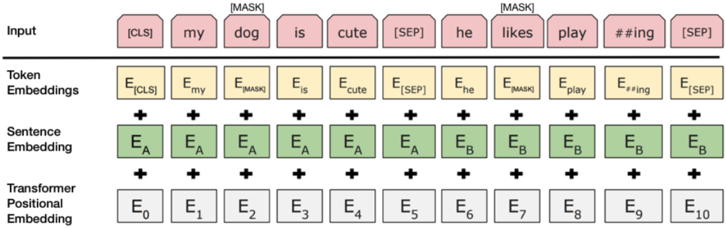


Figure: Bert pre-training task: masked lm

BERT: Next Sentence Prediction (NSP)

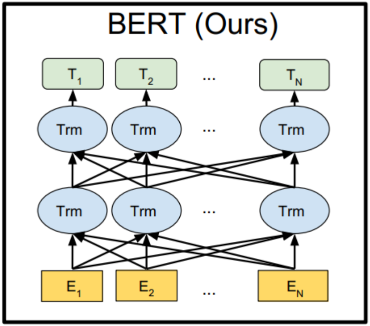
Language modeling do not capture the relationship between 2 text sentences.

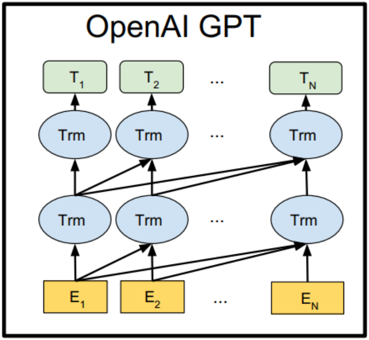


- Inserting 2 special tokens: [CLS] token at the beginning of the first sentence and [SEP] token at the end of each sentence.
- 50% of the time B is the actual next sentence that follows A, and 50% of the time it is a random sentence from the corpus
- The [CLS] token embedding is used to predict whether the last sentence is next sentence.

Is NSP necessary? read RoBERTa paper

GPT is an auto-regressive(AR) model, while Bert is an auto encoder(AE) model. The architecture of GPT can be viewed as transformer decoder blocks.





XLNet: Two-Stream Self-Attention (a little bit harder to understand)

- The introduction of the special symbol [MASK] causes a pretrain-finetune discrepancy.
- AR model is only trained to encode an uni-direction context.

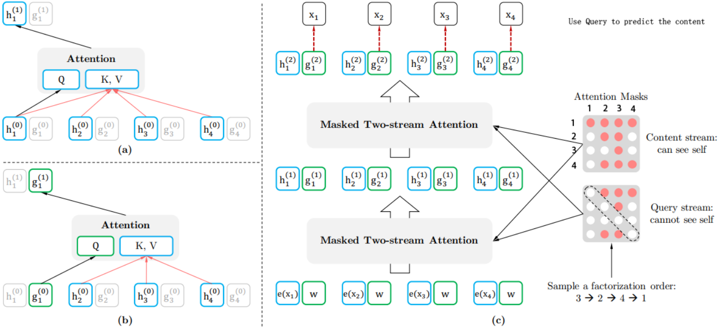


Figure 2: (a): Content stream attention, which is the same as the standard self-attention. (b): Query stream attention, which does not have access information about the content x_{z_t}. (c): Overview of the permutation language modeling training with two-stream attention.

Brilliant ideas!!!

- Using Attention mask to reorder the sequence, which make it a AR model, but can be trained to encode bi-direction contexts or part of the contexts
- Query stream can be viewed as a MASK mechanism in Bert.

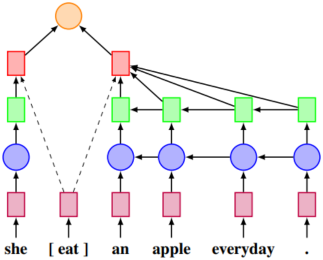
Is XLNet really better than Bert, read https://medium.com/@xlnet.team/a-fair-comparison-study-of-xlnet-and-bert-with-large-models-5a4257f59dc0

Can we integrate the ideas of XLNet into Deep Context Model?

- Using the Corenlp toolkit to locate the target words that need to be checked
- Using bi-direction information to predict the label of a word.
- The idea of Two-stream Self-attention can perfectly integrate into this work. We can apply Query stream to mask the word to be classified, and attention mask to "see"'the bi-direction contexts

Error Type	Classes
article	0 = a/an, 1 = the, 2 = None
preposition	label = preposition index
verb form	0 = base form, 1 = gerund or present participle, 2 = past participle
noun number	0 = singular, 1 = plural
subjective agreement	0 = non-3rd person singular present, 1 = 3rd person singular present

Table 1: Classification labels for different error types.



Kaili Z , Wang C , Li R , et al. A Simple but Effective Classification Model for Grammatical Error Correction[J]. 2018.
Wang C, Li R B, Lin H. Deep Context Model for Grammatical Error Correction[C]//SLaTE. 2017: 167-171.

Attention mask can do more – Unified language model pre-training

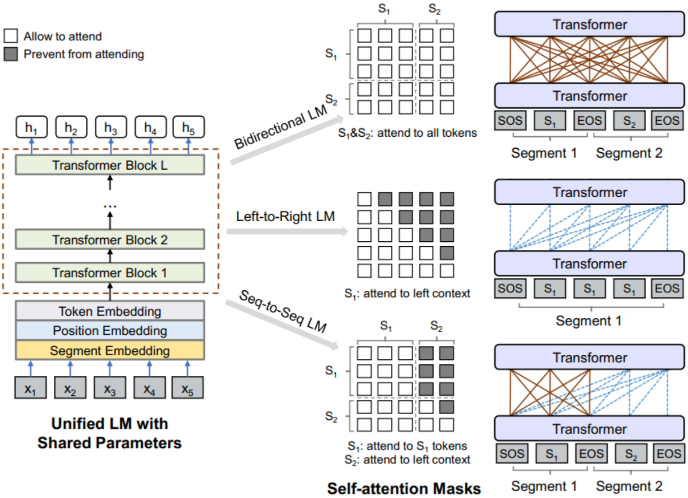


Figure 1: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). We use different self-attention masks to control the access to context for each word token. The right-to-left LM is similar to the left-to-right one, which is omitted in the figure for brevity.

How big are these NLP models?

BERT, large

- 24-layer, 1024-hidden, 16-heads, 340M parameters,
- 13GB texts. BooksCorpus (800M words) and Wikipedia (2,500M words).
- 16 Cloud TPUs (64 TPU chips total) 4days

GPT-2, large

- 48-layers, 1600-hidden, 1542M parameters.
- 40GB of Internet text (10 billion words?)
- 64 Cloud TPU v3 (256 cores), one week

XLNet

- Same architecture as Bert-large
- 128GB text (32 billion sub-words). BooksCorpus (800M words) and Wikipedia (2,500M words), Giga5, ClueWeb, Common Crawl
- 512 TPU v3 chips, 2.5days

Bert is a transformer encoder, and GPT a is transformer decoder.
How about an encoder-decoder architecture?

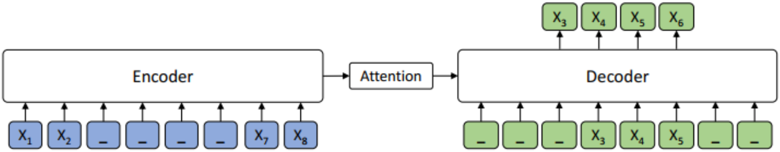
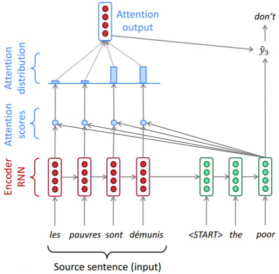


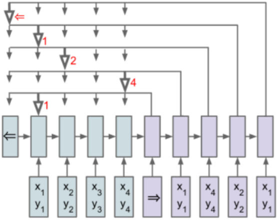
Figure 1. The encoder-decoder framework for our proposed MASS. The token "_" represents the mask symbol [M].

- Mask the inputs of both the encoder and decoder jointly.
- Force the encoder to understand the meaning of the unmasked tokens
- encourage the decoder to extract useful information from the encoder side

Song K , Tan X , Qin T , et al. MASS: Masked Sequence to Sequence Pre-training for Language Generation[J]. 2019.

- Pointer Network is to predict content of next word.




(b) Ptr-Net

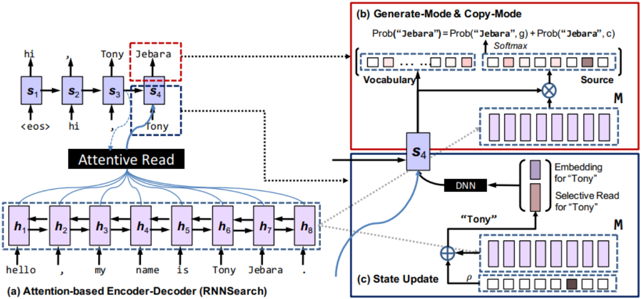
instead of computing conditional vector and concatenate with h_i as the case of Seq2Seq with attention, pointer network directly uses the "attention scores" to predict the next word:

$$\mu_j^i = v^T tanh(W_1 e_j + W_2 d_i), j \in (1, \cdots, n)$$
$$p(C_i | C_1, \cdots, C_{i-1}, P) = softmax(\mu^i)$$

Vinyals O, Fortunato M, Jaitly N. Pointer networks[C]// International Conference on Neural Information Processing Systems. 2015.

CopyNet

Decoder may not be able to translate rare and out of vocabulary words such as the name of a person. Why not just copy them from the source?



- CopyNet directly adds the two probabilities.

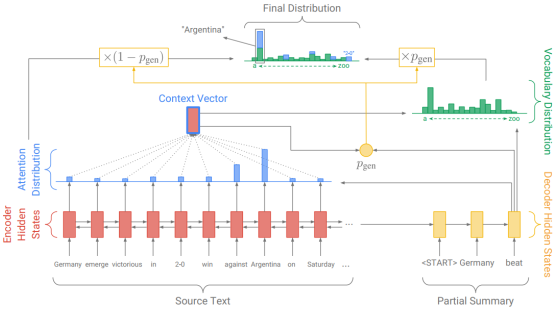
$$\psi_g(y_t = v_i) = V_i^T W_o s_t, v_i \in V \cup UNK \tag{2}$$

$$\psi_c(y_t = x_j) = \sigma(h_j^T W_c) s_t \tag{3}$$

Gu J , Lu Z , Li H , et al. Incorporating Copying Mechanism in Sequence-to-Sequence Learning[J]. 2016.

Pointer-Generator Networks

The original paper of Pointer-Generator Networks is quite similar to CopyNet, expect the details in calculating the probabilities of abstraction and extraction, and how to combine them.



Combines abstraction and extraction together

$$P(w) = P_{gen}P_{vocab}(w) + (1 - P_{gen}) \sum_{i:w_i=w} \alpha_i^t \quad (4)$$

$$P_{gen} = \sigma(w_{h^*}^T h^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (5)$$

Grammar error correction

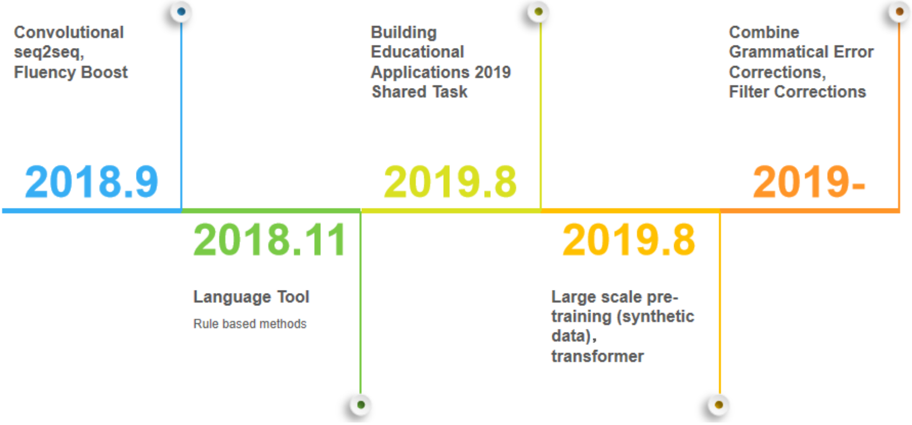
Currently, the state-of-the-art methods mainly contain a bag of tricks:

- Better spell checker
- Pre-training: large synthetic data generation
- Ensemble different models and combine Grammatical Error Corrections
- Post-processing: [unk] edits removal, re-rank with LM
- Copying mechanism may help a little.

Now, it seems that nobody uses "fluent boost" and ConvNet on GEC tasks.
Please read:

- Zhao W, Wang L, Shen K, et al. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data[J]. arXiv preprint arXiv:1903.00138, 2019.
- Kantor Y, Katz Y, Choshen L, et al. Learning to combine Grammatical Error Corrections[J]. arXiv preprint arXiv:1906.03897, 2019.
- Grundkiewicz R, Junczys-Dowmunt M, Heafield K. Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data[C]//Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. 2019: 252-263.
- Choe Y J, Ham J, Park K, et al. A Neural Grammatical Error Correction System Built On Better Pre-training and Sequential Transfer Learning[J]. arXiv preprint arXiv:1907.01256, 2019.
- Li R, Wang C, Zha Y, et al. The LAIX Systems in the BEA-2019 GEC Shared Task[C]//Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. 2019: 159-167.

Time line


Convolutional seq2seq, Fluency Boost
2018.9
2018.11
Language Tool
Rule based methods
Building Educational Applications 2019 Shared Task
2019.8
2019.8
Large scale pre-training (synthetic data), transformer
Combine Grammatical Error Corrections, Filter Corrections
2019-

Quiz: Who is Bert?