BERT

Manfred Klenner

Department of Computerlinguistik
University of Zurich

November 16, 2020

Bert

see the original paper: https://arxiv.org/pdf/1810.04805.pdf

Bert

- a transformer with only a encoder
- self attention
- bi-directional training
- masked LM
- word and sentence prediction

Bert

- a transformer with only a encoder
- self attention
- bi-directional training
- masked LM
- word and sentence prediction

Bert

- a transformer with only a encoder
- self attention
- bi-directional training
- masked LM
- word and sentence prediction

Bert

- a transformer with only a encoder
- self attention
- bi-directional training
- masked LM
- word and sentence prediction

Bert

- a transformer with only a encoder
- self attention
- bi-directional training
- masked LM
- word and sentence prediction

bi-directional?

- directional models: read the text input sequentially (left-to-right or right-to-left)
- transformer encoder reads the entire sequence of words at once
- bidirectional, well, it's non-directional
- attention is the bidirectional means

bi-directional?

- directional models: read the text input sequentially (left-to-right or right-to-left)
- transformer encoder reads the entire sequence of words at once
- bidirectional, well, it's non-directional
- attention is the bidirectional means

bi-directional?

- directional models: read the text input sequentially (left-to-right or right-to-left)
- transformer encoder reads the entire sequence of words at once
- bidirectional, well, it's non-directional
- attention is the bidirectional means

bi-directional?

- directional models: read the text input sequentially (left-to-right or right-to-left)
- transformer encoder reads the entire sequence of words at once
- bidirectional, well, it's non-directional
- attention is the bidirectional means

Masked LM

Input = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

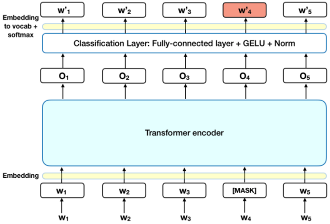
Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]

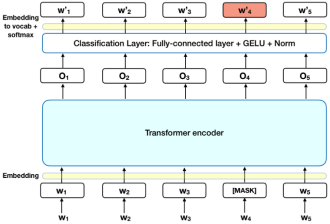
penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

Masked LM

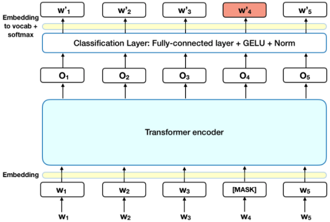


- training: 15% of the words in each sequence are replaced with a [MASK] token
- the model tries to predict the original word behind the mask
- this is based on the context of the other, non-masked, words in the sequence
- the prediction of the output words requires:
  - a classification layer on top of the encoder
  - multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension
  - calculating the probability of each word in the vocabulary with softmax



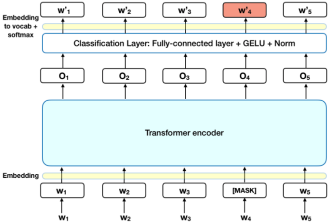
- training: 15% of the words in each sequence are replaced with a [MASK] token
- the model tries to predict the original word behind the mask
- this is based on the context of the other, non-masked, words in the sequence
- the prediction of the output words requires:
  - a classification layer on top of the encoder
  - multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension
  - calculating the probability of each word in the vocabulary with softmax

Masked LM



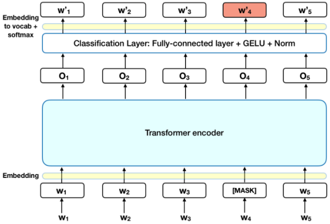
- training: 15% of the words in each sequence are replaced with a [MASK] token
- the model tries to predict the original word behind the mask
- this is based on the context of the other, non-masked, words in the sequence
- the prediction of the output words requires:
  - a classification layer on top of the encoder
  - multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension
  - calculating the probability of each word in the vocabulary with softmax

Masked LM



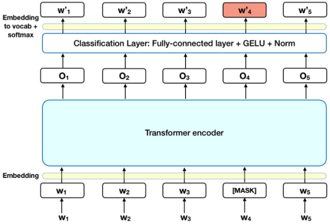
- training: 15% of the words in each sequence are replaced with a [MASK] token
- the model tries to predict the original word behind the mask
- this is based on the context of the other, non-masked, words in the sequence
- the prediction of the output words requires:
  - a classification layer on top of the encoder
  - multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension
  - calculating the probability of each word in the vocabulary with softmax

Masked LM



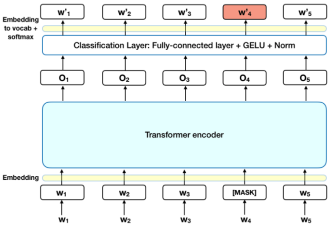
- training: 15% of the words in each sequence are replaced with a [MASK] token
- the model tries to predict the original word behind the mask
- this is based on the context of the other, non-masked, words in the sequence
- the prediction of the output words requires:
  - a classification layer on top of the encoder
  - multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension
  - calculating the probability of each word in the vocabulary with softmax

Masked LM



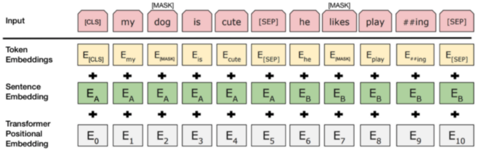
- training: 15% of the words in each sequence are replaced with a [MASK] token
- the model tries to predict the original word behind the mask
- this is based on the context of the other, non-masked, words in the sequence
- the prediction of the output words requires:
  - a classification layer on top of the encoder
  - multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension
  - calculating the probability of each word in the vocabulary with softmax

Masked LM



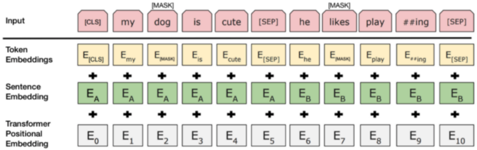
- training: 15% of the words in each sequence are replaced with a [MASK] token
- the model tries to predict the original word behind the mask
- this is based on the context of the other, non-masked, words in the sequence
- the prediction of the output words requires:
  - a classification layer on top of the encoder
  - multiplying the output vectors by the embedding matrix, transforming them into the vocabulary dimension
  - calculating the probability of each word in the vocabulary with softmax

Next Sentence Prediction (NSP)



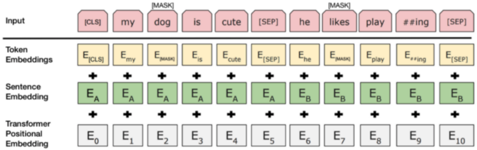
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



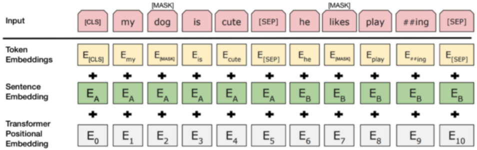
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



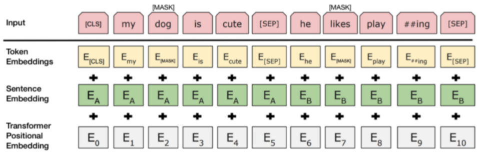
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



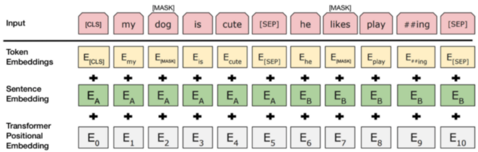
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



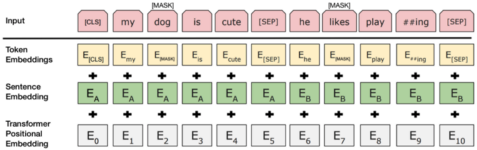
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



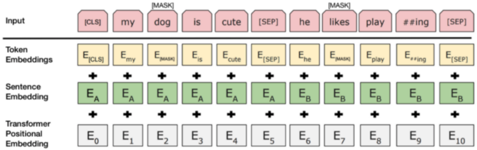
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



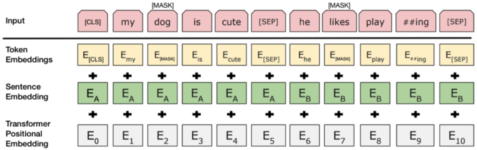
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



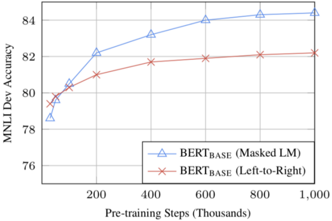
- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Next Sentence Prediction (NSP)



- pairs of sentences, learn whether the 2. sentence is realy the subsequent sentence
- 50% of the inputs are truly pairs, the others are wrong pairs
- the assumption is that the random sentence will be disconnected from the first sentence
- preprocessing:
  - a [CLS] token is inserted at the beginning of the first sentence and
  - a [SEP] token is inserted at the end of each sentence
  - a sentence embedding indicating sentence A or sentence B is added to each token
  - sentence embeddings are similar to token embeddings with a vocabulary of 2
  - a positional embedding is added to each token to indicate its position in the sequence

Learning curve



	Training Compute + Time	Usage Compute
$\text{BERT}_{\text{BASE}}$	4 Cloud TPUs, 4 days	1 GPU
$\text{BERT}_{\text{LARGE}}$	16 Cloud TPUs, 4 days	1 TPU

- model size matters: Bert has 340 million parameters. 24 layer, 1024 hidden nodes, 16 attention heads
- enough training data, more training steps == higher accuracy

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