Natural Language Processing (CS-472) Spring-2023

Muhammad Naseer Bajwa

Assistant Professor,
Department of Computing, SEECS
Co-Principal Investigator,
Deep Learning Lab, NCAI
NUST, Islamabad
naseer.bajwa@seecs.edu.pk

Overview of this week's lecture

Pretraining Transformers

- Subword Modelling
- Model pretraining from word embeddings
- GPT: Pretrained decoder
- BERT: Pretrained encoder
- T5: Pretrained encoder-decoder

Why subword models are required?

- So far we have assumed a fixed vocabulary from the training set.
  - All *novel* words seen at test time are mapped to a single UNK token.

	Word	**Vocab Mapping**
Common words	Hat	Hat (Index)
	Learn	Learn (Index)
Variations	Goooood	UNK (Index)
Misspellings	Laern	UNK (Index)
Novel Items	Transformerify	UNK (Index)

Why subword models are required?

- So far we have assumed a fixed vocabulary from the training set.
  - All *novel* words seen at test time are mapped to a single UNK token.
- Finite vocabulary assumptions make even less sense in languages other than English.

	Word	**Vocab Mapping**
Common words	Hat	Hat (Index)
	Learn	Learn (Index)
Variations	Goooood	UNK (Index)
Misspellings	Laern	UNK (Index)
Novel Items	Transformerify	UNK (Index)

Why subword models are required?

- So far we have assumed a fixed vocabulary from the training set.
  - All *novel* words seen at test time are mapped to a single UNK token.
- Finite vocabulary assumptions make even less sense in languages other than English.
- Many languages exhibit complex morphology, or word structure.
  - The effect is more word types, each occurring fewer times.

	Word	**Vocab Mapping**
Common words	Hat	Hat (Index)
	Learn	Learn (Index)
Variations	Goooood	UNK (Index)
Misspellings	Laern	UNK (Index)
Novel Items	Transformerify	UNK (Index)

NUST
Defining futures
School of Electrical Engineering
& Computer Science

The byte-pair encoding algorithm is a method make words using word-pieces

- Subword modelling encompasses methods for reasoning about structure below the word level.
  - Parts of words, characters, bytes, etc.

The byte-pair encoding algorithm is a method make words using word-pieces

- Subword modelling encompasses methods for reasoning about structure below the word level.
    - Parts of words, characters, bytes, etc.
- The dominant modern paradigm is to learn a vocabulary of parts of words (subword tokens).
    - At training and testing time, each word is split into a sequence of known subwords.

The byte-pair encoding algorithm is a method make words using word-pieces

- Subword modelling encompasses methods for reasoning about structure below the word level.
  - Parts of words, characters, bytes, etc.
- The dominant modern paradigm is to learn a vocabulary of parts of words (subword tokens).
  - At training and testing time, each word is split into a sequence of known subwords.
- Byte-pair encoding is a simple yet effective strategy for defining a subword vocabulary.
  - Start with a vocabulary containing only characters and an 'end-of-word' symbol.
  - Using a corpus of text, find the most common adjacent characters 'a, b'; add 'ab' as a Subword.
  - Replace instances of the character pair with the new subword; repeat until desired vocab size.
  - Example: Starting characters $\{a, b, \dots, z\}$. Ending vocab: $\{a, \dots, z, apple, app\#, \#ly, \dots\}$

https://www.youtube.com/watch?v=HEikzVL-lZU

Subword models take advantage of lexical morphology

- Common words end up being a part of the Subword vocabulary, while rarer words are split into (sometimes intuitive, sometimes not) components.
  - In the worst case, words are split into as many subwords as they have characters.

	Word	**Vocab Mapping**
Common words	Hat	Hat
	Learn	Learn
Variations	Goooood	Goo##oo##od
Misspellings	Laern	La##ern##
Novel Items	Transformerify	Transformer##ify

Same word in different context should have different embeddings

- Recall the adage in the early lectures,

You shall know a word by the company it keeps

(J. R. Firth, 1957)

Same word in different context should have different embeddings

- Recall the adage in the early lectures,

You shall know a word by the company it keeps

(J. R. Firth, 1957)

- This quote is a summary of distributional semantics and motivated $word2vec$.

'… the complete meaning of a word is always contextual, and no study of meaning apart from a complete context can be taken seriously.'

(J. R. Firth, 1935)

Same word in different context should have different embeddings

- Recall the adage in the early lectures,

You shall know a word by the company it keeps

(J. R. Firth, 1957)

- This quote is a summary of distributional semantics and motivated $word2vec$.

'… the complete meaning of a word is always contextual, and no study of meaning apart from a complete context can be taken seriously.'

(J. R. Firth, 1935)

- Consider the sentence,

'I record the record.'

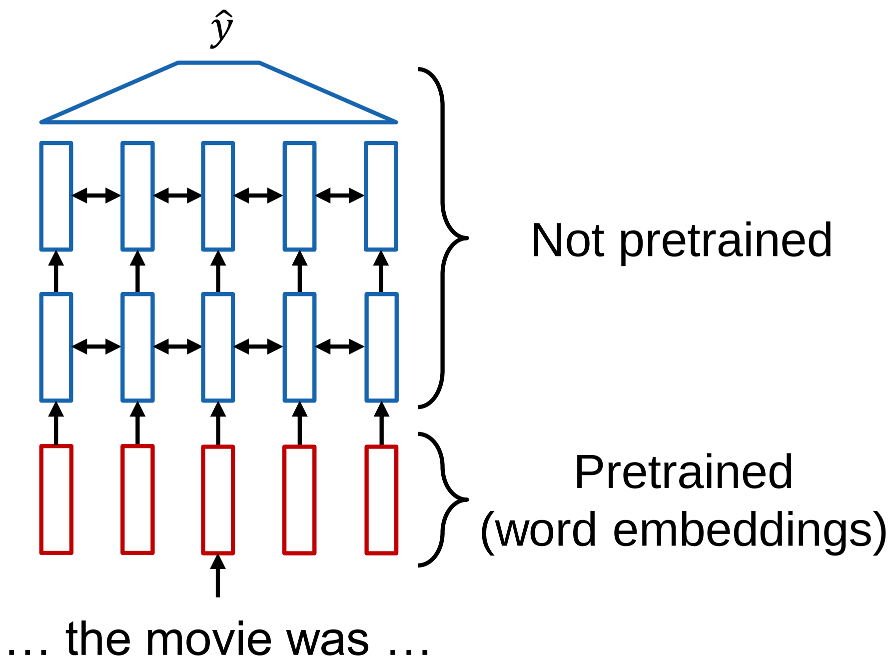
  - The two instances of the word 'record' mean differently.

Transfer learning was limited in NLP until around 2017

- Until 2017:

  - Start with pretrained word embeddings (no context!).

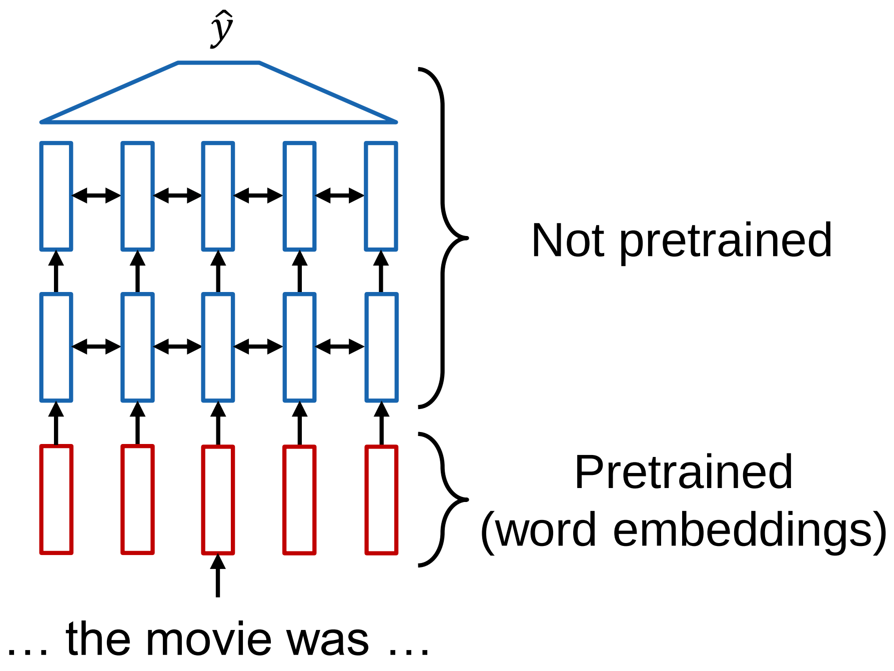
  - Learn how to incorporate context in an LSTM or Transformer while training on the upstream task (using supervision).



[The word ‘movie’ gets the same word embedding, no matter what sentence it shows up in.]

Transfer learning was limited in NLP until around 2017

- Until 2017:
  - Start with pretrained word embeddings (no context!).
  - Learn how to incorporate context in an LSTM or Transformer while training on the upstream task (using supervision).
- What's wrong with this approach?

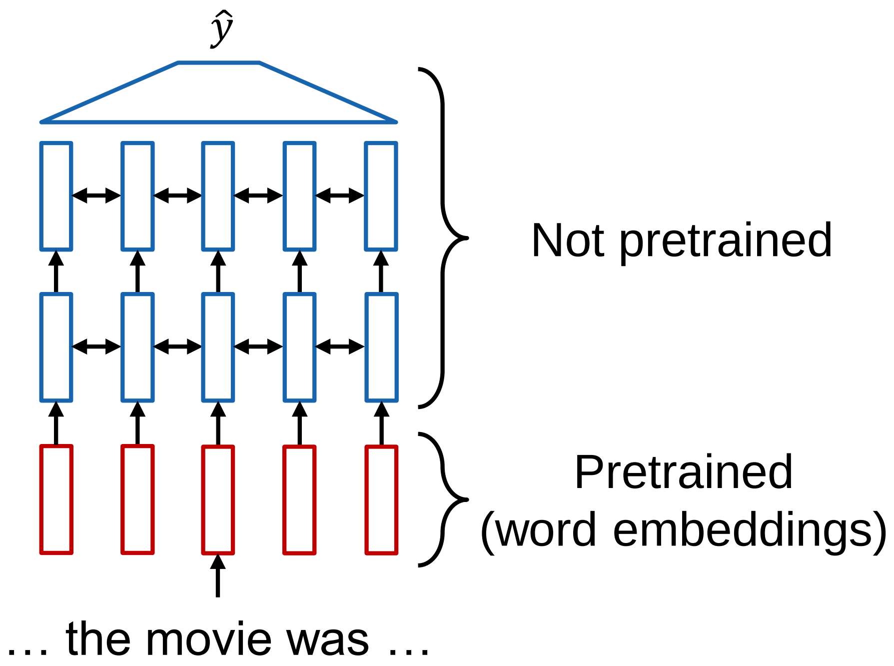


[The word 'movie' gets the same word embedding, no matter what sentence it shows up in.]

NUST
Defining futures
School of Electrical Engineering & Computer Science

Transfer learning was limited in NLP until around 2017

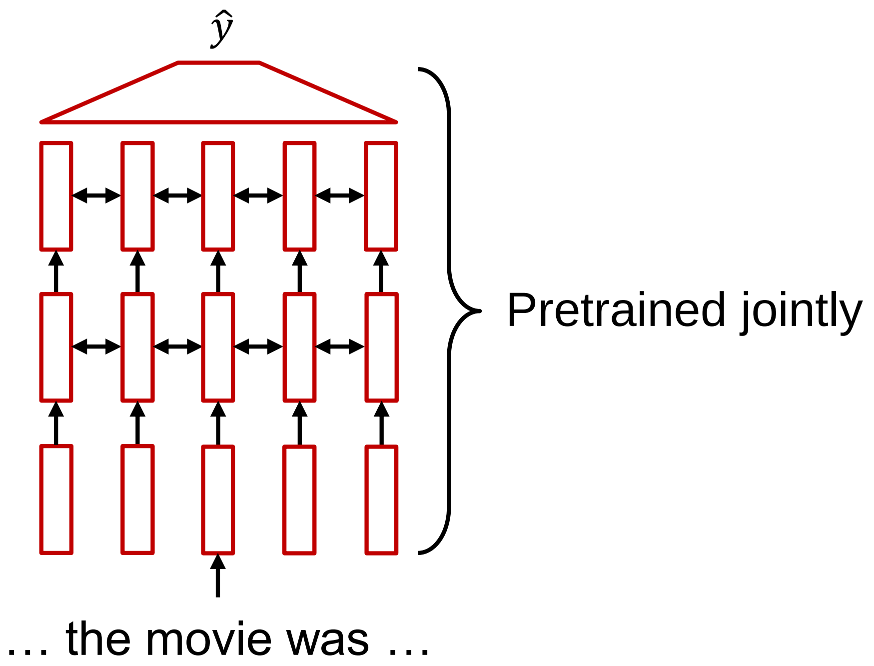
- Until 2017:
  - Start with pretrained word embeddings (no context!).
  - Learn how to incorporate context in an LSTM or Transformer while training on the upstream task (using supervision).
- What's wrong with this approach?
  - The training data we have for our downstream task (like question answering) must be sufficient to teach all contextual aspects of language.
  - Most of the parameters in our network are randomly initialized!



[The word 'movie' gets the same word embedding, no matter what sentence it shows up in.]

Transfer learning has become more prevalent in NLP now

- All (or almost all) parameters in NLP networks are initialised via pretraining.

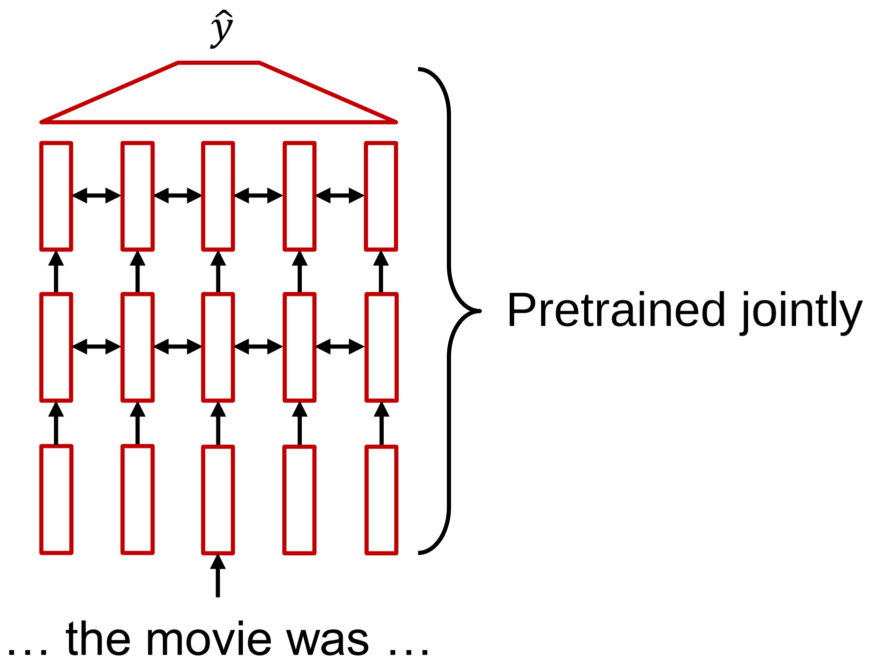


[The model learns how to represent the entire sentence through pretraining]

Transfer learning has become more prevalent in NLP now

- All (or almost all) parameters in NLP networks are initialised via pretraining.

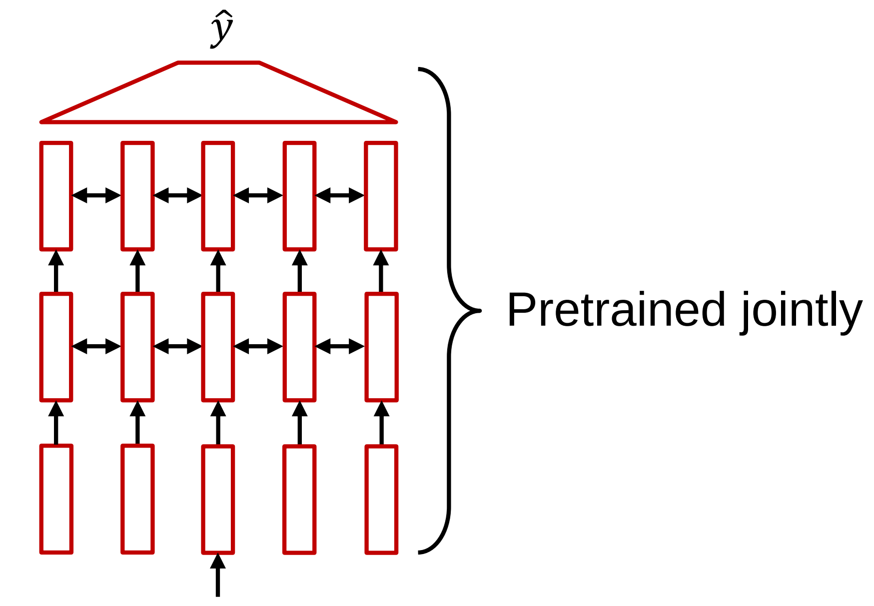
- Pretraining methods hide parts of the input from the model, and trains the model to reconstruct those parts.



[The model learns how to represent the entire sentence through pretraining]

Transfer learning has become more prevalent in NLP now

- All (or almost all) parameters in NLP networks are initialised via pretraining.
- Pretraining methods hide parts of the input from the model, and trains the model to reconstruct those parts.
- This has been exceptionally effective at building strong:
    - Representations of language
    - Parameter initialisations for strong NLP models.
    - Probability distributions over language that we can sample from.



[The model learns how to represent the entire sentence through pretraining]

What can we learn from reconstructing the input?

- Faisal Mosque is located in __________, Pakistan. (Trivial information)

What can we learn from reconstructing the input?

- Faisal Mosque is located in ___________, Pakistan. (Trivial information)
- I put _________fork down on the table. (syntactic categories)

What can we learn from reconstructing the input?

- Faisal Mosque is located in __________, Pakistan. (Trivial information)
- I put _________fork down on the table. (syntactic categories)
- The woman walked across the street, checking for traffic over _____________ shoulder. (coreference)

What can we learn from reconstructing the input?

- Faisal Mosque is located in __________, Pakistan. (Trivial information)
- I put _________fork down on the table. (syntactic categories)
- The woman walked across the street, checking for traffic over ______________ shoulder. (coreference)
- I went to the ocean to see the fish, turtles, seal, and ______________. (semantic category)

What can we learn from reconstructing the input?

- Faisal Mosque is located in __________, Pakistan. (Trivial information)
- I put _________fork down on the table. (syntactic categories)
- The woman walked across the street, checking for traffic over ______________ shoulder. (coreference)
- I went to the ocean to see the fish, turtles, seal, and ______________. (semantic category)
- Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___________. (sentiment)

What can we learn from reconstructing the input?

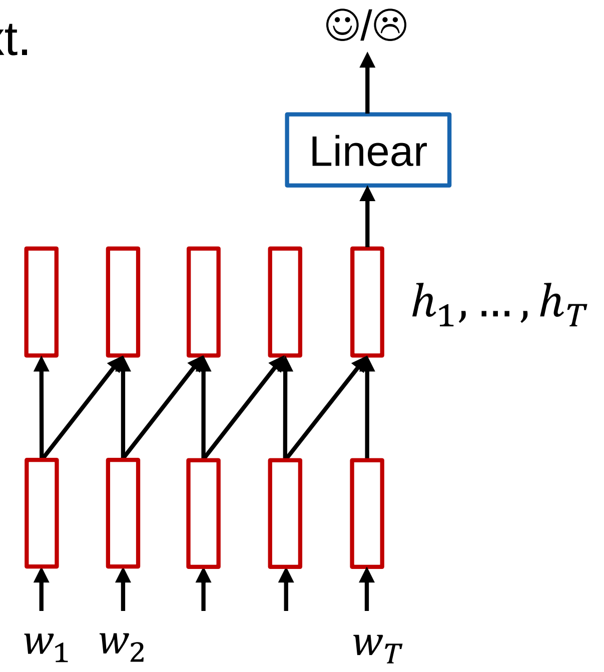
- Faisal Mosque is located in __________, Pakistan. (Trivial information)
- I put _________fork down on the table. (syntactic categories)
- The woman walked across the street, checking for traffic over _____________ shoulder. (coreference)
- I went to the ocean to see the fish, turtles, seal, and ______________. (semantic category)
- Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___________. (sentiment)
- Adriano went into the kitchen to make some tea. Standing next to Adriano, Dominique pondered his destiny. Dominique left the ____________. (spatial location/ logical reasoning)

What can we learn from reconstructing the input?

- Faisal Mosque is located in __________, Pakistan. (Trivial information)
- I put _________fork down on the table. (syntactic categories)
- The woman walked across the street, checking for traffic over ______________ shoulder. (coreference)
- I went to the ocean to see the fish, turtles, seal, and ______________. (semantic category)
- Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___________. (sentiment)
- Adriano went into the kitchen to make some tea. Standing next to Adriano, Dominique pondered his destiny. Dominique left the ____________. (spatial location/ logical reasoning)
- I was thinking about the sequence that goes, 1, 1, 2, 3, 5, 8, 13, _____(algorithm)

Pretraining decoders can be used for classification tasks

- Train a neural network to perform language modelling on a large amount of text.
    - Learn $p(w_t|w_{1:t-1})$.
    - Save the network parameters.

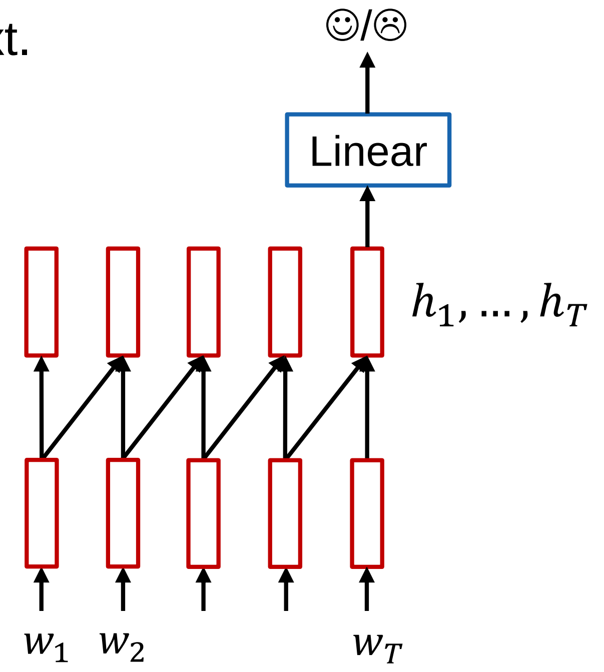


[The linear layer hasn't been pretrained and must be learnt from scratch.]

Pretraining decoders can be used for classification tasks

- Train a neural network to perform language modelling on a large amount of text.
  - Learn $p(w_t | w_{1:t-1})$.
  - Save the network parameters.

- When using decoders pretrained as language model, ignore that they were trained to model language.



[The linear layer hasn't been pretrained and must be learnt from scratch.]

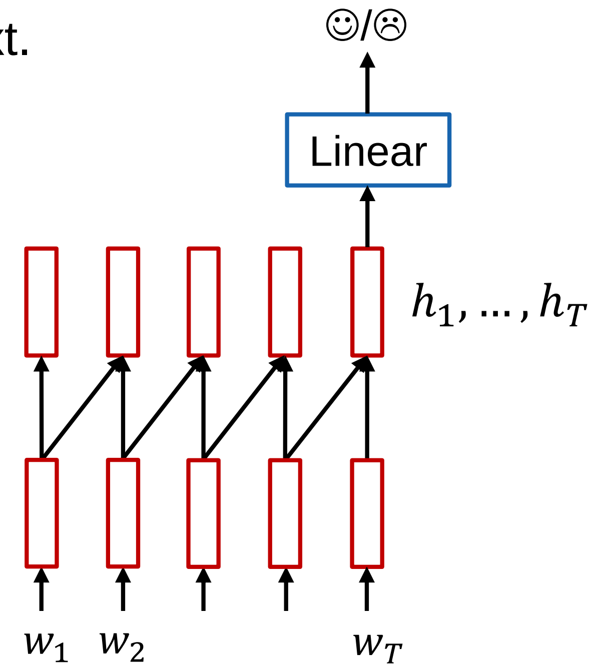
Pretraining decoders can be used for classification tasks

- Train a neural network to perform language modelling on a large amount of text.
    - Learn $p(w_t|w_{1:t-1})$.
    - Save the network parameters.
- When using decoders pretrained as language model, ignore that they were trained to model language.
- Finetune them by training a classifier on the last word's hidden state.

$$h_1, \ldots, h_T = Decoder\,(w_1, \ldots, w_T)$$

$$y \sim Ah_t + b$$

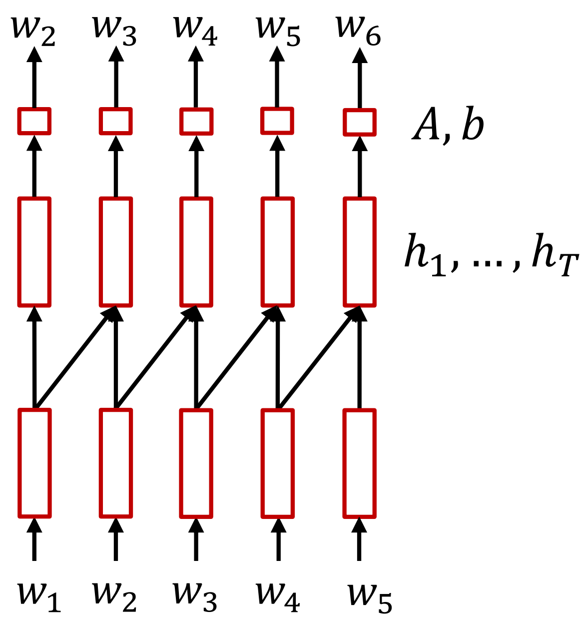
    - Where A and b are randomly initialised and specified by the downstream task.
    - Gradients backpropagate through the whole network.



[The linear layer hasn't been pretrained and must be learnt from scratch.]

Pretraining decoders can also be used for generative tasks

- Train a language model on lots of text to learn general things.



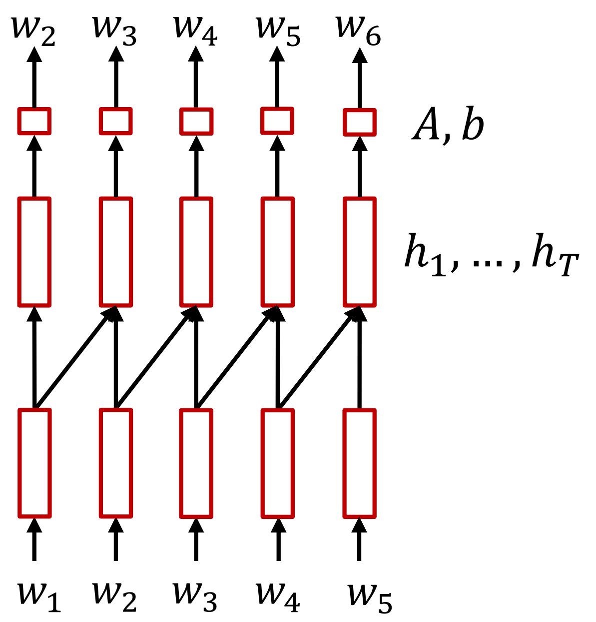
[The linear layer has been pretrained.]

Pretraining decoders can also be used for generative tasks

- Train a language model on lots of text to learn general things.

- Finetune on downstream generative task with few labels and adapt to the new task.

    - Objective remains the same, i.e. $p_\theta(w_t|w_{1:t-1})$



[The linear layer has been pretrained.]

Pretraining decoders can also be used for generative tasks

- Train a language model on lots of text to learn general things.

- Finetune on downstream generative task with few labels and adapt to the new task.

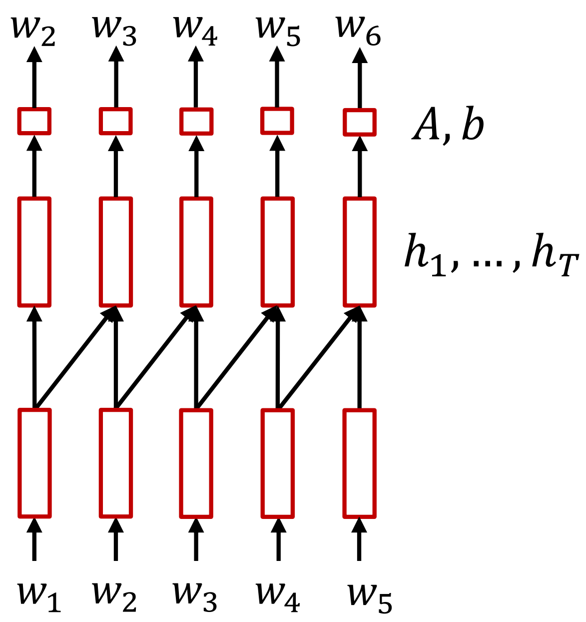
    - Objective remains the same, i.e. $p_\theta(w_t|w_{1:t-1})$

- This is helpful in tasks where the output is a sequence with a vocabulary like that at pretraining time like dialogue generation or summarisation.

$$h_1, \dots, h_T = Decoder\ (w_1, \dots, w_T)$$

$$w_t \sim Ah_t + b$$

    - Where A and b were pretrained in the language model.



[The linear layer has been pretrained.]

GPT was a big success in pretraining a decoder

- GPT consisted of
  - A transformer decoder with 12 layers.
  - 768-dimensional hidden states, 3072-dimensional feedforward hidden layers.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT was a big success in pretraining a decoder

- GPT consisted of
    - A transformer decoder with 12 layers.
    - 768-dimensional hidden states, 3072-dimensional feedforward hidden layers.
- It used byte-pair encoding with 40,000 merges.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT was a big success in pretraining a decoder

- GPT consisted of
  - A transformer decoder with 12 layers.
  - 768-dimensional hidden states, 3072-dimensional feedforward hidden layers.
- It used byte-pair encoding with 40,000 merges.
- It was trained on BooksCorpus: over 7000 unique books.
  - Contains long spans of contiguous text, for learning long-distance dependencies.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT was a big success in pretraining a decoder

- GPT consisted of
  - A transformer decoder with 12 layers.
  - 768-dimensional hidden states, 3072-dimensional feedforward hidden layers.
- It used byte-pair encoding with 40,000 merges.
- It was trained on BooksCorpus: over 7000 unique books.
  - Contains long spans of contiguous text, for learning long-distance dependencies.
- The acronym GPT never showed up in the original paper. It could stand for ‘Generative PreTraining’ or ‘Generative Pretrained Transformer’.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

How input is formatted for finetuning tasks in GPT?

- Natural Language Inference is a sentence classification task with three labels as *entailing/contradictory/neutral*.

    - Premise: He drives daily to work.

    - Hypothesis: He owns a car.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

How input is formatted for finetuning tasks in GPT?

- Natural Language Inference is a sentence classification task with three labels as *entailing/contradictory/neutral*.
    - Premise: He drives daily to work.
    - Hypothesis: He owns a car.
- The input is formatted as a sequence of tokens for the decoder.

[START] The man is in the doorway [DELIM] The person is near the door [EXTRACT]

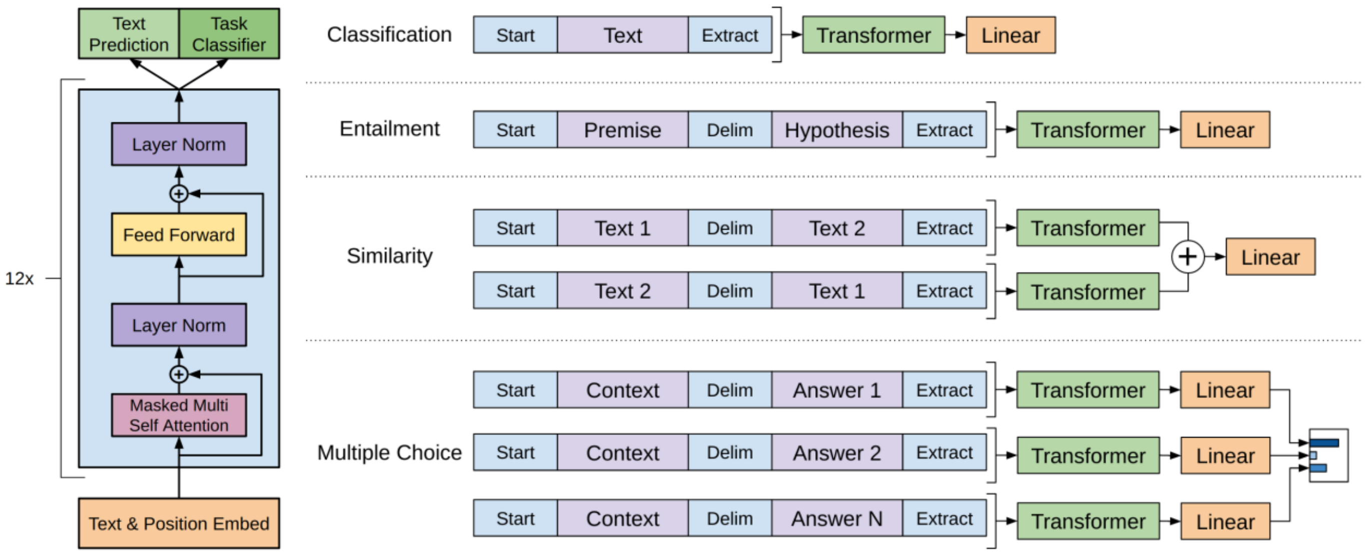


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

How input is formatted for finetuning tasks in GPT?

- Natural Language Inference is a sentence classification task with three labels as *entailing/contradictory/neutral*.
    - Premise: He drives daily to work.
    - Hypothesis: He owns a car.
- The input is formatted as a sequence of tokens for the decoder.

[START] The man is in the doorway [DELIM] The person is near the door [EXTRACT]

- The linear classifier is applied to the representation of the [EXTRACT] token.

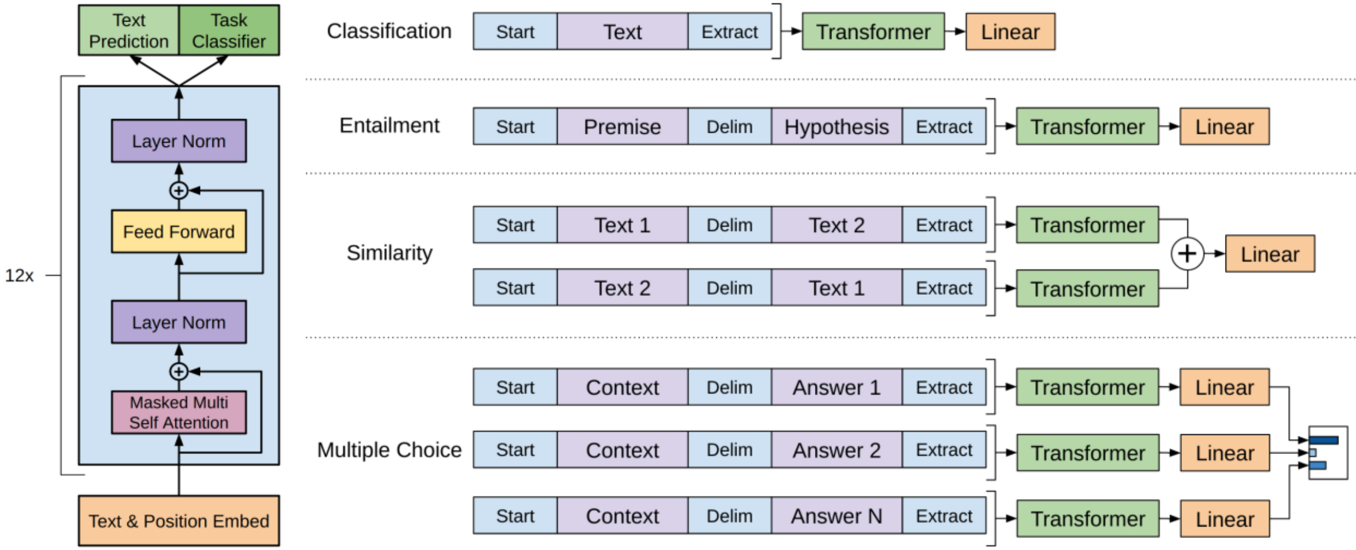


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT set new state-of-the-art on many NLP tasks

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo [44] (5x)	-	-	89.3	-	-	-
CAFE [58] (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network [35] (3x)	80.6	80.1	-	-	-	-
CAFE [58]	78.7	77.9	88.5	83.3		
GenSen [64]	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn [64]	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: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip [55]	76.5	-	-	-
Hidden Coherence Model [7]	77.6	-	-	-
Dynamic Fusion Net [67] (9x)	-	55.6	49.4	51.2
BiAttention MRU [59] (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (ours)	**86.5**	**62.9**	**57.4**	**59.0**

Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT-2 generated increasingly convincing text

- GPT-2 is just bigger GPT trained on larger dataset.

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

GPT-2 generated increasingly convincing text

- GPT-2 is just bigger GPT trained on larger dataset.

- It is used as a generative model that can produce reasonably good text.

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

GPT-2 generated increasingly convincing text

- GPT-2 is just bigger GPT trained on larger dataset.

- It is used as a generative model that can produce reasonably good text.

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.

Pretrained transformer encoders can be used for learning embeddings

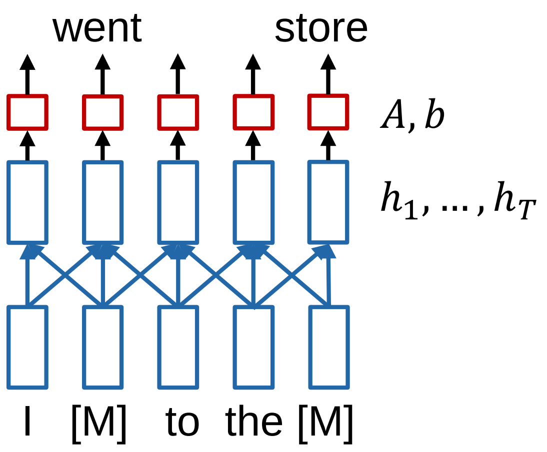
- What pretraining objective should be used for encoders?
  - Since encoders have bidirectional context, they cannot be trained as language models.

Pretrained transformer encoders can be used for learning embeddings

- What pretraining objective should be used for encoders?
  - Since encoders have bidirectional context, they cannot be trained as language models.
- Transformer encoders are pretrained on the task of Masked Language Modelling.
  - Replace some fraction of words in the input with a special [MASK] token
  - Predict the masked words using bidirectional context.

$$h_1, \ldots, h_T = Encoder\,(w_1, \ldots, w_T)$$

$$y_i \sim Aw_i + b$$

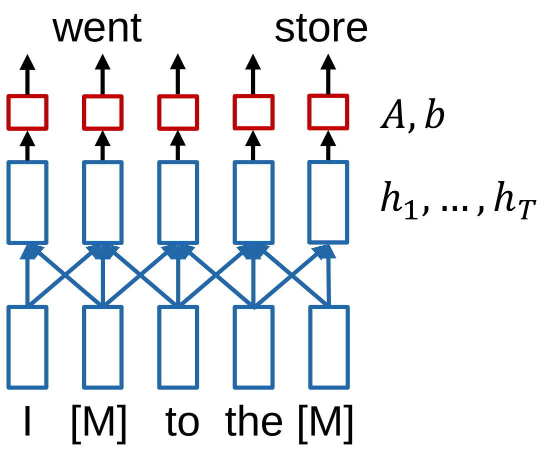


Pretrained transformer encoders can be used for learning embeddings

- What pretraining objective should be used for encoders?
  - Since encoders have bidirectional context, they cannot be trained as language models.
- Transformer encoders are pretrained on the task of Masked Language Modelling.
  - Replace some fraction of words in the input with a special [MASK] token
  - Predict the masked words using bidirectional context.

$$h_1, \ldots, h_T = Encoder\,(w_1, \ldots, w_T)$$

$$y_i \sim Aw_i + b$$



- Only add loss terms from words that are masked out.
  - Let $\tilde{x}$ be the masked version of x, learn $p_\theta(x|\tilde{x})$

BERT is the most famous pretrained encoder model.

- BERT was released in 2018 by Google.

Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT is the most famous pretrained encoder model.

- BERT was released in 2018 by Google.
- It predicts a random 15% of (sub)word tokens.
  - Eighty percent of these 15% tokens are replaced with [MASK] token.
  - Ten percent token are replaced with a random token.
  - Remaining 10% are left unchanged yet still predicted by the model.

Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT is the most famous pretrained encoder model.

- BERT was released in 2018 by Google.
- It predicts a random 15% of (sub)word tokens.
  - Eighty percent of these 15% tokens are replaced with [MASK] token.
  - Ten percent token are replaced with a random token.
  - Remaining 10% are left unchanged yet still predicted by the model.
- Why replace token with random words or leave them unchanged but still predict?
  - It doesn’t let the model complacent and encourage building strong representations of non-masked words also.
  - No masks are provided at fine-tuning time.

Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT pretraining input is a two separate contiguous chunks of text

- The models is trained to predict whether one chunk follows the other or is it randomly sampled.
  - Later works showed this was not necessary.

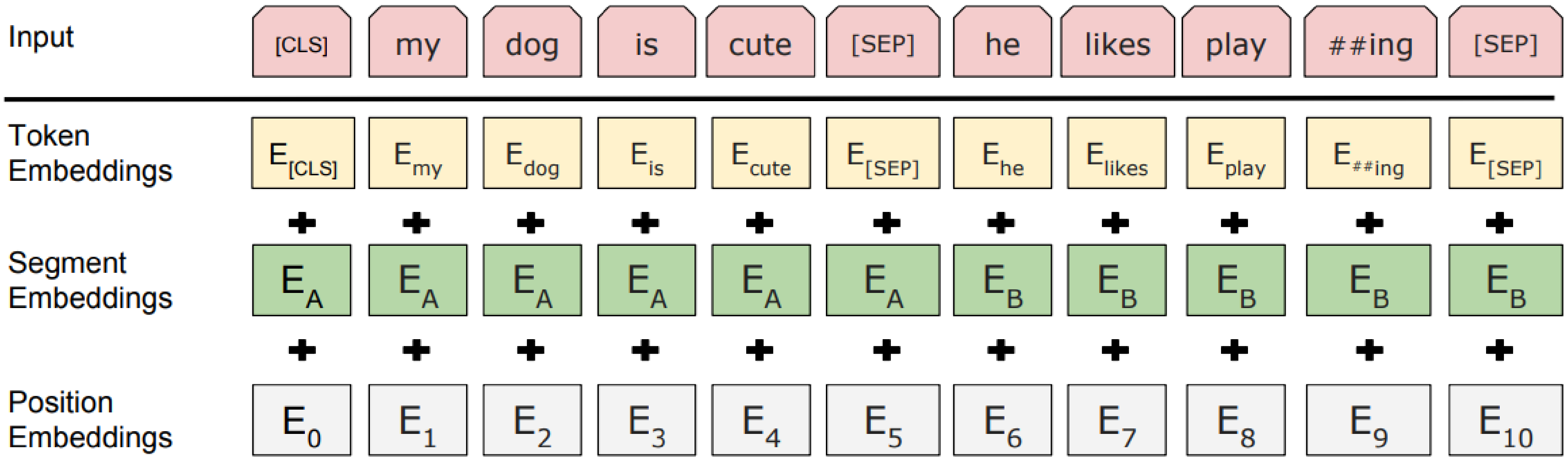


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT is a large and computation heavy model

- BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million parameters.
- BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million parameters.

Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT is a large and computation heavy model

- BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million parameters.

- BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million parameters.

- The model was trained on BookCorpus (800 million words) and English Wikipedia (2.5 billion words).

Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT is a large and computation heavy model

- BERT-base: 12 layers, 768-dim hidden states, 12 attention heads, 110 million parameters.

- BERT-large: 24 layers, 1024-dim hidden states, 16 attention heads, 340 million parameters.

- The model was trained on BookCorpus (800 million words) and English Wikipedia (2.5 billion words).

- Pretrained with 64 TPUs for four days.

    - Finetuning can be done on single GPU on smaller datasets.

Devlin, Jacob, et al. "BERT: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

BERT is massively popular and hugely versatile

- Finetuning BERT led to new state-of-the-art on a range of NLP tasks.

QQP: Quora Question Pairs

QNLI: Natutal Language Inference over Question Answering

SST-2: Stanford Sentiment Treebank

CoLA: Corpus of Linguistic Acceptability

STS-B: Semantic Textual Similarity

MRPC: Microsoft Paraphrase Corpus

RTE: A small NLI corpus.

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

There are many variants of BERT

- RoBERTa: Just trained BERT for longer and removed next sentence prediction.

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with Books + Wiki	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with Books + Wiki	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with Books + Wiki	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6

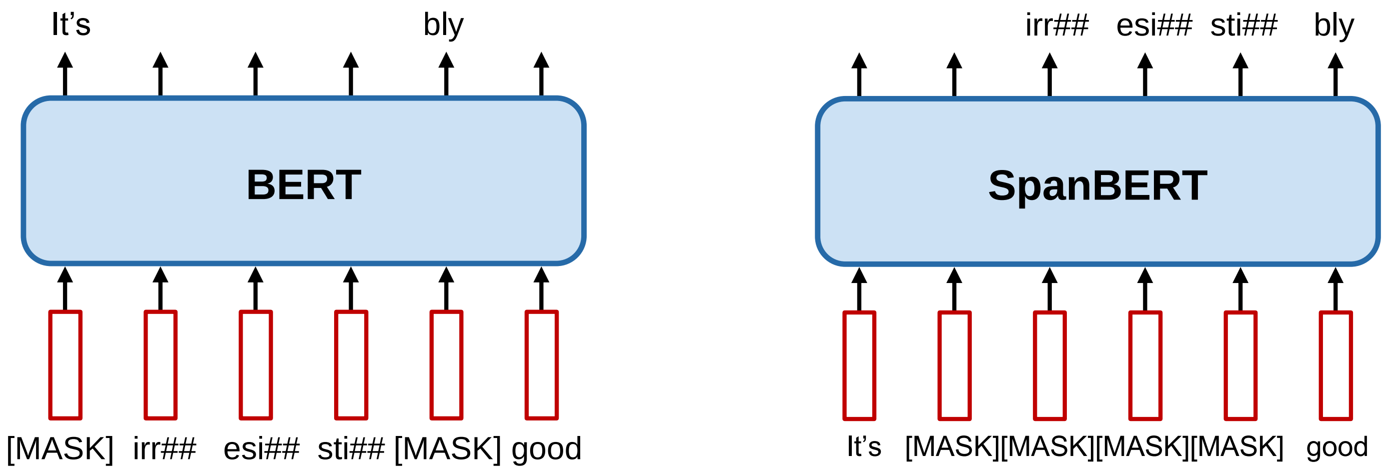
Liu, Yinhan, et al. "RoBERTa: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

There are many variants of BERT

- RoBERTa: Just trained BERT for longer and removed next sentence prediction.

- SpanBERT: Masking contiguous spans of (sub)words makes a harder, more useful pretraining task.

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6



Liu, Yinhan, et al. "RoBERTa: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

Joshi, Mandar, et al. "SpanBERT: Improving pre-training by representing and predicting spans." Transactions of the Association for Computational Linguistics 8 (2020): 64-77.

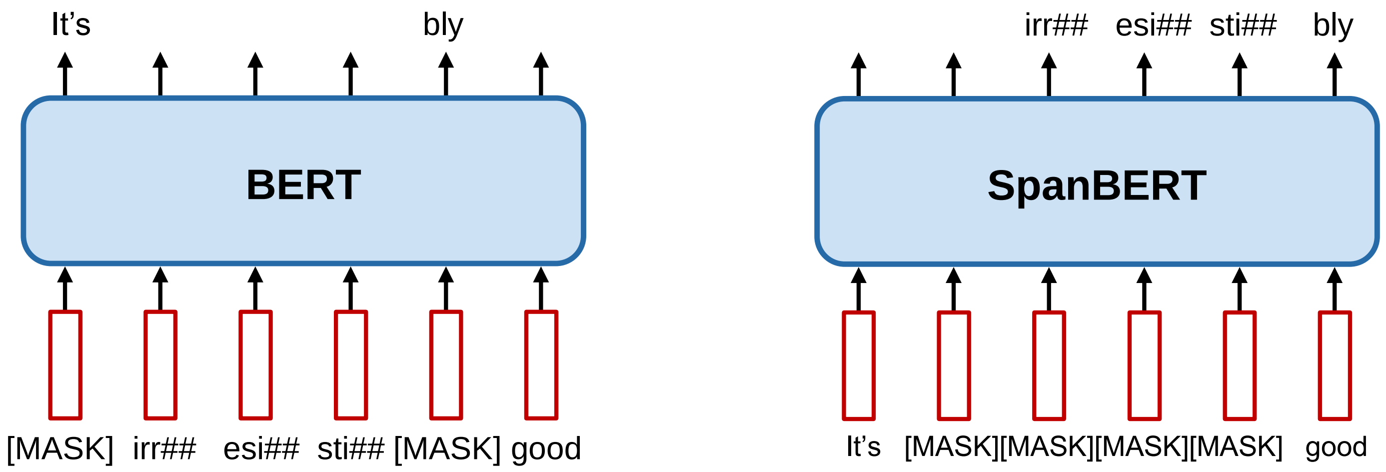
There are many variants of BERT

- RoBERTa: Just trained BERT for longer and removed next sentence prediction.

- SpanBERT: Masking contiguous spans of (sub)words makes a harder, more useful pretraining task.

- FinBERT: BERT trained on financial data for sentiment analysis.

Model	data	bsz	steps	SQuAD (v1.1/2.0)	MNLI-m	SST-2
RoBERTa						
with BOOKS + WIKI	16GB	8K	100K	93.6/87.3	89.0	95.3
+ additional data (§3.2)	160GB	8K	100K	94.0/87.7	89.3	95.6
+ pretrain longer	160GB	8K	300K	94.4/88.7	90.0	96.1
+ pretrain even longer	160GB	8K	500K	**94.6/89.4**	**90.2**	**96.4**
$\text{BERT}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	90.9/81.8	86.6	93.7
$\text{XLNet}_{\text{LARGE}}$						
with BOOKS + WIKI	13GB	256	1M	94.0/87.8	88.4	94.4
+ additional data	126GB	2K	500K	94.5/88.8	89.8	95.6



Liu, Yinhan, et al. "RoBERTa: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).
Araci, Dogu. "FinBERT: Financial sentiment analysis with pre-trained language models." arXiv preprint arXiv:1908.10063 (2019).
Joshi, Mandar, et al. "SpanBERT: Improving pre-training by representing and predicting spans." Transactions of the Association for Computational Linguistics 8 (2020): 64-77.

Pretraining encoders has some limitations

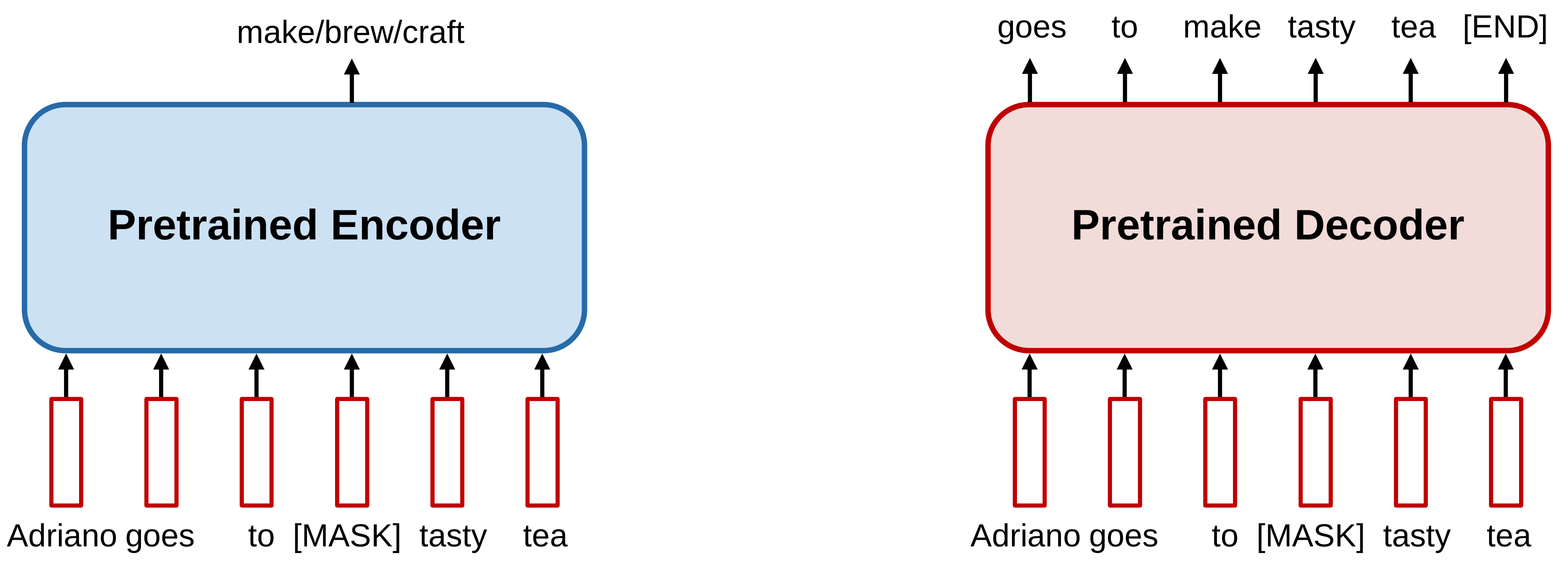
- BERT gave excellent results and is versatile but we can’t use it for everything.

Pretraining encoders has some limitations

- BERT gave excellent results and is versatile but we can’t use it for everything.

- For generative tasks, pretrained decoders are still better choice.

  - BERT and other pretrained encoders don’t naturally lead to nice autoregressive generation methods.



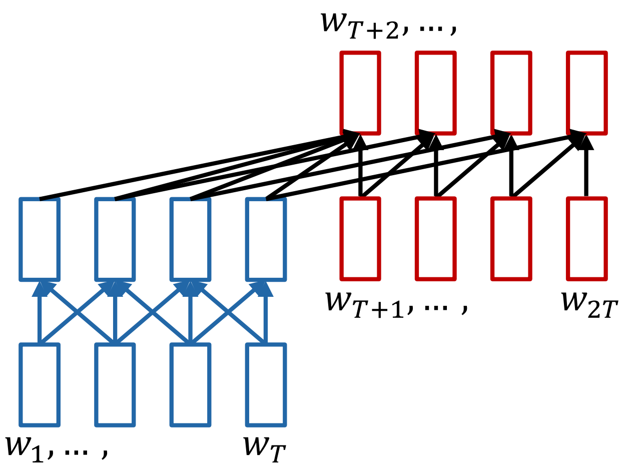
How can we pretrain an encoder-decoder model?

- Train encoder-decoder model like language modelling.

$$h_1, \dots, h_T = Encoder\ (w_1, \dots, w_T)$$

$$h_{T+1}, \dots, h_{2T} = Decoder\ (w_1, \dots, w_T, h_1, \dots, h_T)$$

$$y_i \sim Aw_i + b, i > T$$



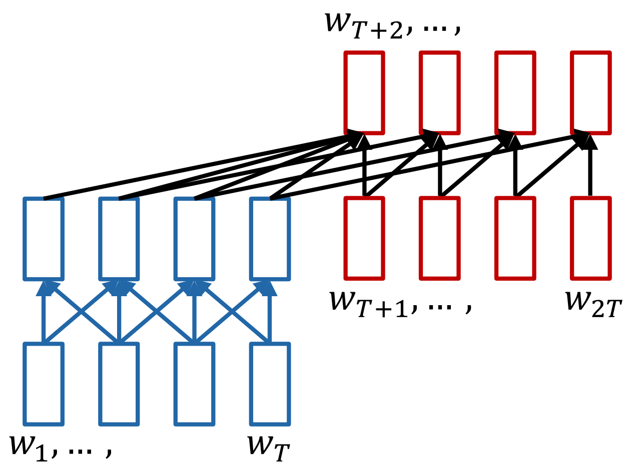
How can we pretrain an encoder-decoder model?

- Train encoder-decoder model like language modelling.

$$h_1, \dots, h_T = Encoder\ (w_1, \dots, w_T)$$

$$h_{T+1}, \dots, h_{2T} = Decoder\ (w_1, \dots, w_T, h_1, \dots, h_T)$$

$$y_i \sim Aw_i + b, i > T$$



- The encoder portion benefits from bidirectional context; the decoder portion is used to train the whole model through language modelling.

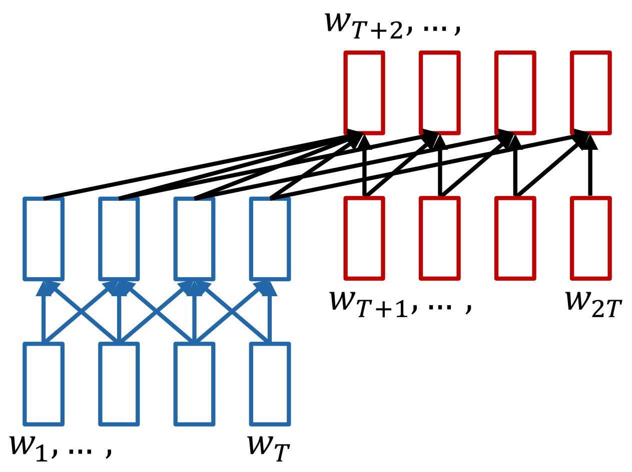
How can we pretrain an encoder-decoder model?

- Train encoder-decoder model like language modelling.

$$h_1, \dots, h_T = Encoder\ (w_1, \dots, w_T)$$

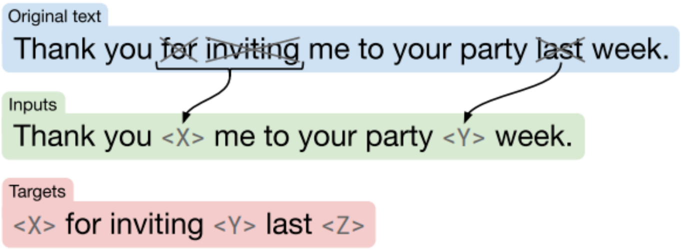
$$h_{T+1}, \dots, h_{2T} = Decoder\ (w_1, \dots, w_T, h_1, \dots, h_T)$$

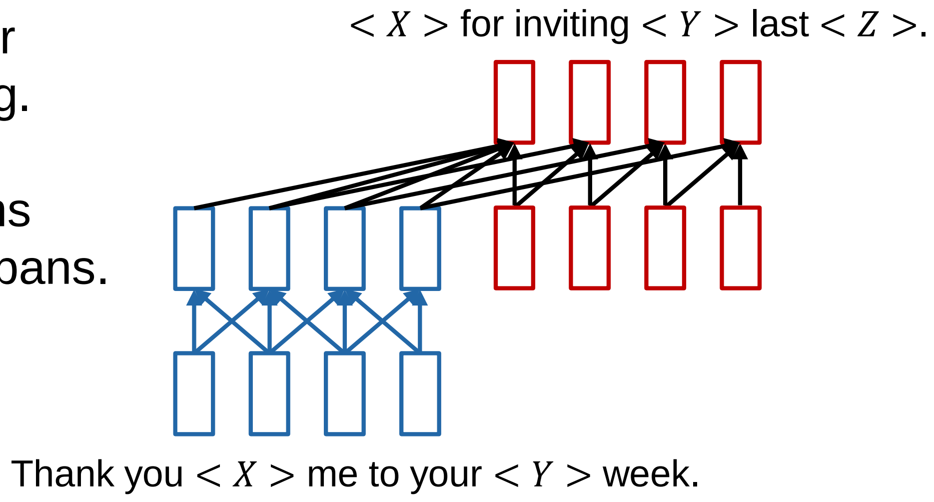
$$y_i \sim Aw_i + b, i > T$$



- The encoder portion benefits from bidirectional context; the decoder portion is used to train the whole model through language modelling.

- T5 used Span Corruption for training. Replace different-length spans from the input with unique placeholders and decode the replaced spans.





Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research 21.1 (2020): 5485-5551.

Pretraining encoder-decoder worked better than decoders for some tasks

- Training T5 model using span corruption was found to work better than language modelling.

Architecture	Objective	Params	Cost	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Encoder-decoder	Denoising	$2P$	M	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	**39.82**	**27.65**
Enc-dec, shared	Denoising	P	M	82.81	18.78	**80.63**	**70.73**	26.72	39.03	**27.46**
Enc-dec, 6 layers	Denoising	P	$M/2$	80.88	18.97	77.59	68.42	26.38	38.40	26.95
Language model	Denoising	P	M	74.70	17.93	61.14	55.02	25.09	35.28	25.86
Prefix LM	Denoising	P	M	81.82	18.61	78.94	68.11	26.43	37.98	27.39
Encoder-decoder	LM	$2P$	M	79.56	18.59	76.02	64.29	26.27	39.17	26.86
Enc-dec, shared	LM	P	M	79.60	18.13	76.35	63.50	26.62	39.17	27.05
Enc-dec, 6 layers	LM	P	$M/2$	78.67	18.26	75.32	64.06	26.13	38.42	26.89
Language model	LM	P	M	73.78	17.54	53.81	56.51	25.23	34.31	25.38
Prefix LM	LM	P	M	79.68	17.84	76.87	64.86	26.28	37.51	26.76

In-context learning and very large models

- So far the pretrained models we learnt are used in two ways
  - Sample from the distribution they define (maybe providing a prompt).
  - Fine-tune the on a task we care about and take their prediction.

In-context learning and very large models

- So far the pretrained models we learnt are used in two ways
  - Sample from the distribution they define (maybe providing a prompt).
  - Fine-tune the on a task we care about and take their prediction.
- Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide them within context.

In-context learning and very large models

- So far the pretrained models we learnt are used in two ways
    - Sample from the distribution they define (maybe providing a prompt).
    - Fine-tune the on a task we care about and take their prediction.
- Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide them within context.
- The in-context examples seem to specify the task to be performed, and the conditional distribution mocks performing the task to a certain extent.
    - Thanks -> Danke
    - Please -> Bitte

Hello -> Hallo
Excuse me -> ?

In-context learning and very large models

- So far the pretrained models we learnt are used in two ways
  - Sample from the distribution they define (maybe providing a prompt).
  - Fine-tune the on a task we care about and take their prediction.
- Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide them within context.
- The in-context examples seem to specify the task to be performed, and the conditional distribution mocks performing the task to a certain extent.
  - Thanks -> Danke
  - Please -> Bitte

  Hello -> Hallo
  Excuse me -> Entschuldegung
- **GPT-3** is the canonical example of this. The largest T5 model had 11 billion parameters. GPT-3 has 175 billion parameters.

Exact mechanism of in-context learning is not well understood

Learning via SGD during unsupervised pre-training

5 + 8 = 13	*gaot* → *goat*	*thanks* → *danke*
7 + 2 = 9	*sakne* → *snake*	*hello* → *hallo*
1 + 0 = 1	*brid* → *bird*	*bread* → *brot*
3 + 4 = 7	*fsih* → *fish*	*wall* → *wand*
5 + 9 = 14	*dcuk* → *duck*	*mint* → *minze*
9 + 8 = 17	*cmihp* → *chimp*	*othter* → *andere*
In-context learning	In-context learning	In-context learning

Do you have any problem?

Some material (images, tables, text etc.) in this presentation has been borrowed from different books, lecture notes, and the web. The original contents solely belong to their owners, and are used in this presentation only for clarifying various educational concepts. Any copyright infringement is ***not at all*** intended.

EOP