

Deep Learning for Natural Language Processing

Lecture 9 – Transformer architectures and BERT

Dr. Ivan Habernal

June 8, 2021

Trustworthy Human Language Technologies
Department of Computer Science
Technical University of Darmstadt



www.trusthlt.org

BERT¹ — The “gamechanger”

Best paper award at NAACL
2019

State-of-the-art results on
various NLP tasks

Directly applicable to other
domains and languages

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova

Google AI Language

{jacobdevlin, mingweichang, kentonl, kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language repre-

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The *feature-based* approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that

¹J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of NAACL*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186

What we won't do



Figure 2: To spice up the lectures, the lecturer is dressed in an ELMo costume

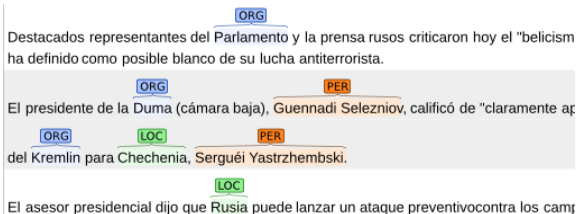
E. Artemova, M. Apishev, D. Kirianov, V. Sarkisyan, S. Aksenov, and O. Serikov (2021). “Teaching a Massive Open Online Course on Natural Language Processing”. In: *Proceedings of the Fifth Workshop on Teaching NLP*. Online: Association for Computational Linguistics, pp. 13–27. URL: <https://www.aclweb.org/anthology/2021.teachingnlp-1.2>

NLP tasks

Short recap of "NLP tasks"

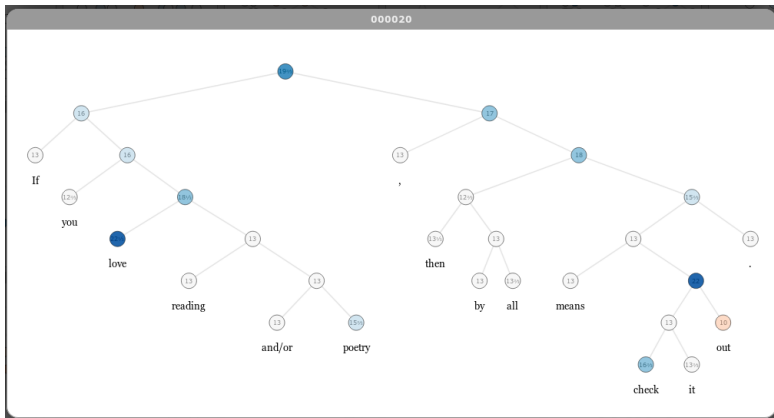
Single-sentence "tagging" tasks, such as

- Part of speech tagging (not in BERT paper)
- Named Entity Recognition



Short recap of "NLP tasks" – Single sentence

– Sentiment of a sentence²



²<https://nlp.stanford.edu/sentiment/treebank.html>

More complicated NLP tasks...

Reasoning about two sentences: Natural Language Inference³

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

³<https://nlp.stanford.edu/projects/snli/>

More complicated NLP tasks...

Question answering

- Natural language questions with locations of their answers in Wikipedia articles

Why are these NLP tasks hard?

Although some methods can "exploit" artifacts in data,⁴ the tasks can be truly solved only by

- Understanding meaning of words (semantics)
- Understanding relations between meanings
- Understanding syntax (negations, quantifiers, etc.)
- Reasoning about the world

⁴S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith (2018). "Annotation Artifacts in Natural Language Inference Data". In: *Proceedings of NAACL*. New Orleans, LA: Association for Computational Linguistics, pp. 107–112

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Deep learning prerequisites

We know the "nail", let's take the hammer

Prerequisites: We know

- Neural network basics (layers, activations, softmax, convolutions)
- Where are the learnable parameters ("weight matrices" and biases),
- What are loss functions (e.g., cross-entropy for classification)
- How to train them (back-propagation, batches, SGD or Adam)
- Word embeddings (dense semantic representation)

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Neural networks ✓

- Learn non-linear dependencies, learn representations

Embeddings ✓

- Dense token representation

Neural Machine Translation

Neural machine translation (NMT)

Why machine translation here?

BERT builds upon techniques from MT

What is machine translation?

- Another popular NLP task
- Many large-scale parallel corpora available



Figure 1: MT is a challenging task!

Image source: <https://languagelog.ldc.upenn.edu/nll/?p=3978>

Traditionally **encoder-decoder** architectures

- One recurrent neural network processes the entire input and generate its dense representation (**encoder**)
- Other recurrent network produces one token at the time conditioned on the previous states and generated tokens (**decoder**)

Neural MT: Typical architectures (up to 2016-2017)

Long short-term memory (LSTM) / GRU networks

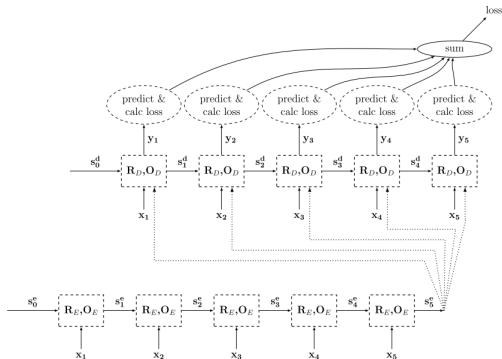


Figure 2: Encoder-decoder RNN.⁵

⁵Figure from (Goldberg, 2016)

Bottlenecks of RNN for machine translation?

Inherently **sequential** nature

- No parallelization
- Big memory footprint (you must "remember" the entire sequence)
- Long-range dependencies modeling: Distance plays a role!

...but when the goal is to learn a good representation of the input sequence, why not use...

- Convolutional neural networks?

Convolutional neural nets (CNN) recap

One particular property of CNNs

- Modeling dependencies for a **local context**, but by **stacking layers**, one exactly controls the context size

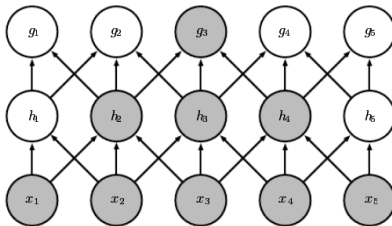


Figure 3: Receptive field of units in deeper layers is larger.

Source: I. Goodfellow, Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. URL: www.deeplearningbook.org

Convolutional neural nets for MT

CNNs competitive with RNNs for MT⁶

- Input tokens as word embeddings (not new) or sub-words (will be explained later)
- Fixed-length input? Set-up a maximum length and use **<PAD>**ding
- But positional information of tokens is lost...

⁶J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin (2017). “Convolutional Sequence to Sequence Learning”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Sydney, Australia: PMLR, pp. 1243–1252 (from Facebook AI Research)

Convolutional neural nets for MT by Gehring, Auli, Grangier, Yarats, and Dauphin (2017)

Solution: Positional embeddings

- For each input position n , train another embedding vector P_n : $P_1 = (1.12, -78.6, \dots)$, P_2, \dots, P_N
- Word embeddings and position embeddings are simply summed up for each input token
- Why? The model knows with which part of the input/output is dealing with
 - Notice: Removing positional embeddings \rightarrow only slightly worse performance

State-of-the-art results and **9.3–21.3 \times faster** than LSTMs on GPU

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Neural networks ✓

- Learn non-linear dependencies, learn representations

Embeddings ✓

- Dense token representation

Neural machine translation ✓

- Sequence to sequence models
- Positional embeddings

Attention “is all you need”

Attention: Modeling dependencies

Recap: How to model long-range dependencies in input?

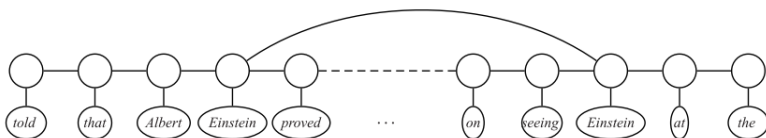
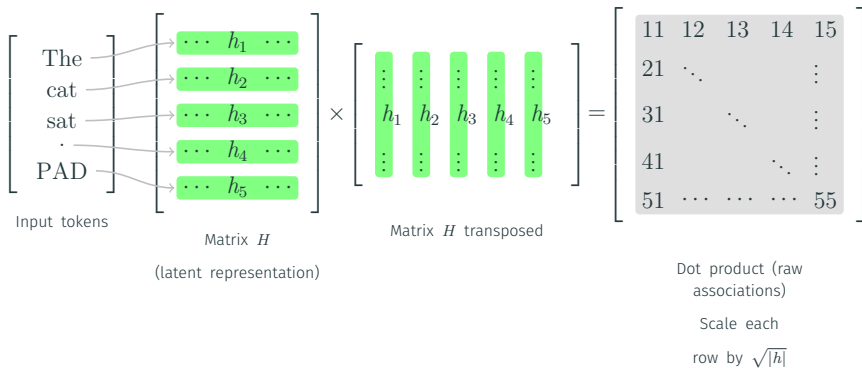


Figure 1: An example of the label consistency problem. Here we would like our model to encourage entities *Albert Einstein* and *Einstein* to get the same label, so as to improve the chance that both are labeled *PERSON*.

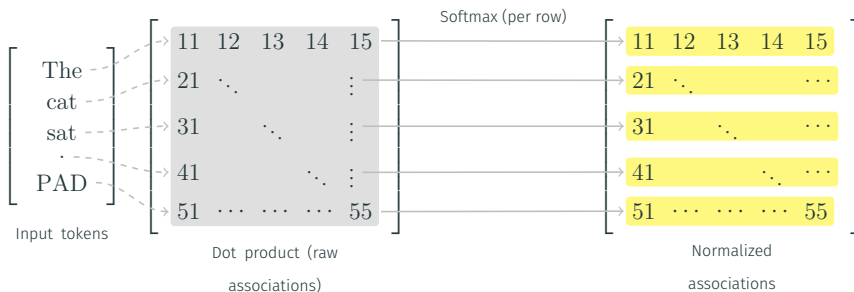
- RNNs or stacking CNNs
- **Self-Attention**: Utilize associations between all input word pairs

Figure source: V. Krishnan and C. D. Manning (2006). “An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition”. In: *Proceedings of ACL*. Sydney, Australia: Association for Computational Linguistics, pp. 1121–1128

Self-Attention in detail (1)

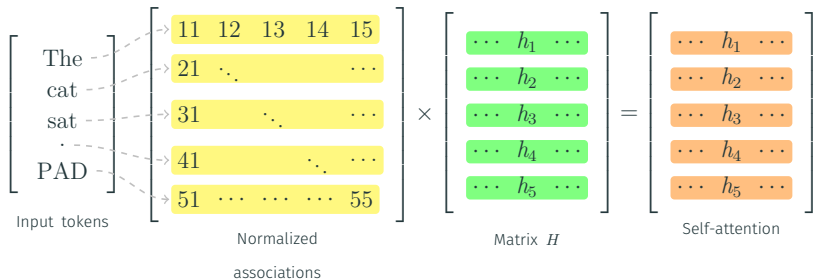


Self-Attention in detail (2)



- Each row corresponds to an input token
- Each row sums up to 1
- Each cell shows the "association strength" with all other tokens

Self-Attention in detail (3)



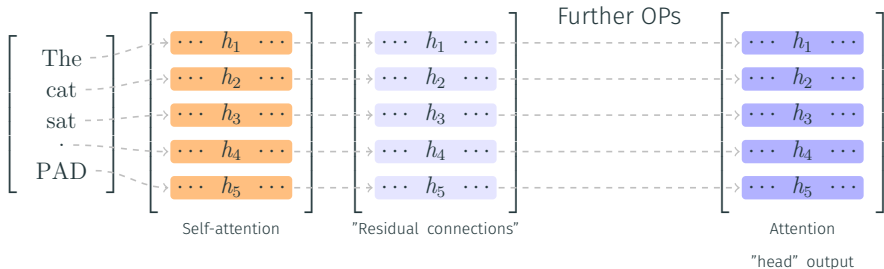
Each position in the latent representation of a token is weighted by the association strength with other tokens

Self-Attention in detail (4)

$$\begin{bmatrix} \cdots h_1 \cdots \\ \cdots h_2 \cdots \\ \cdots h_3 \cdots \\ \cdots h_4 \cdots \\ \cdots h_5 \cdots \end{bmatrix} + \begin{bmatrix} \cdots h_1 \cdots \\ \cdots h_2 \cdots \\ \cdots h_3 \cdots \\ \cdots h_4 \cdots \\ \cdots h_5 \cdots \end{bmatrix} = \begin{bmatrix} \cdots h_1 \cdots \\ \cdots h_2 \cdots \\ \cdots h_3 \cdots \\ \cdots h_4 \cdots \\ \cdots h_5 \cdots \end{bmatrix}$$

Self-attention Matrix H "Residual connections"

Self-Attention in detail: Head



Further operations

- Layer normalization
- Feed-forward layer with ReLU
- Another residual connection and layer normalization

Self-attention: More subtleties

- Run N attention “heads” in parallel and concatenate
- Stack on top of each other M -times

Why self-attention?

- Self-attention layer connects all positions with a constant number of sequentially executed operations
- Recurrent layer requires $O(n)$ sequential operations
- Self-attention layers are **fast**

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems 30*. Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Neural networks ✓

- Learn non-linear dependencies, learn representations

Embeddings ✓

- Dense token representation

Neural machine translation ✓

- Sequence to sequence models
- Positional embeddings

Attention ✓

- Efficient long-range dependencies

Out-of-vocabulary words

Pitfalls of machine translation: Vocabulary and Out-of-vocabulary (OOV)

- MT often with fixed word vocabularies
 - Even though translation is fundamentally an open vocabulary problem (names, numbers, dates etc.).
 - Initially, the most frequent words were used, and all others **<UNK>**⁷
- Translation of out-of-vocabulary (OOV) words
 - Rare words (OOV) handled with a back-off dictionary, or simply copied 1:1 from source to target

⁷P. Koehn (2017). “Neural Machine Translation”. In: *arXiv preprint*.
URL: <http://arxiv.org/abs/1709.07809>

Sub-word units: Motivation?

Sub-words for voice search (Japanese, Korean)⁸

- Too large vocabularies for these two languages would produce way too many OOVs

Later known as WordPiece model

- Adapted by Google's Neural Machine Translation⁹
and eventually by BERT

⁸M. Schuster and K. Nakajima (2012). "Japanese and Korean voice search". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, pp. 5149–5152

⁹Y. Wu et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation". In: *arXiv*, pp. 1–23. URL: <http://arxiv.org/abs/1609.08144>

Sub-word units: Motivation?

But why should sub-word units give better translations than copying or back-off dictionary?

- Open-vocabulary MT better by representing rare and unseen words as a sequence of subword units¹⁰

¹⁰R. Sennrich, B. Haddow, and A. Birch (2016). “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of ACL*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725

What are WordPiece units?

- Similar to a vocabulary: A list of all known (sub-)words, including characters
 - Each word is either entirely a WordPiece unit, or can be split into several WordPiece units
- Splitting a text into the trained WordPiece model shipped along with BERT:

```
tokenizer.tokenize("All human beings are born  
free and equal in dignity and rights.")  
['all', 'human', 'beings', 'are', 'born',  
'free', 'and', 'equal', 'in', 'dignity',  
'and', 'rights', '.']
```

WordPiece units: Multilingual

- `print(tokenizer.tokenize("Alle Menschen sind frei und gleich an Würde und Rechten geboren."))`
- `['all', '##e', 'men', '##schen', 'sin', '##d', 'fr', '##ei', 'und', 'g', '##lei', '##ch', 'an', 'wu', '##rde', 'und', 'rec', '##ht', '##en', 'ge', '##bor', '##en', '.']`
- BERT WordPiece tokenizer: Lower casing, punctuation removal**
- More languages?**
- `tokenizer.tokenize("Все люди рождаются свободными и равными в своем достоинстве и правах.")`
`tokenizer.tokenize("Všichni lidé se rodí svobodní a sobě rovní co do důstojnosti a práv.")`
`tokenizer.tokenize("ყველა ადამიანი იბადება თავისუფალი და თანასწორი თავისი ღირსებითა და უფლებებით.")`
- `['в', '##с', '##е', 'л', '##ю', '##д', '##и', 'р', '##о', '##ж', '##д', '##а', '##ю', '##т', '##с', '##я', 'с', '##в', '##о', '##б', '##о', '##д', '##н', '##ы', '##м', '##и', 'и', 'р', '##а', '##в', '##н', '##ы', '##м', '##и', 'в', 'с', '##в', '##о', '##е', '##м', 'д', '##о', '##с', '##т', '##о', '##и', '##н', '##с', '##т', '##в', '##е', 'и', 'п', '##р', '##а', '##в', '##а', '##х', '.']`
`['vs', '##ich', '##ni', 'lid', '##e', 'se', 'rod', '##i', 'sv', '##ob', '##od', '##ni', 'a', 'sob', '##e', 'ro', '##vn', '##i', 'co', 'do', 'dust', '##oj', '##nos', '##ti', 'a', 'pr', '##av', '.']`
`['[UNK]', 'д', '##д', '##д', '##д', '##о', '##д', '##б', '##о', 'о', '##д', '##д', '##д', '##д', '##д', '##д', '[UNK]', 'д', '##д', '[UNK]', 'т', '##д', '##з', '##о', '##с', '##о', '[UNK]', 'д', '##д']`

Training WordPiece inventory

1. Init the WordPiece inventory with all characters (in all alphabets)
2. For each possible tuple of known WordPieces
 - Create a new candidate WordPiece from the tuple (simply concatenate)
 - Build a language model and compute likelihood on the corpus
3. Select the candidate with the maximum likelihood increase and add to the WordPiece inventory; Go back to 2 or finish, if WordPiece inventory has the desired size

M. Schuster and K. Nakajima (2012). "Japanese and Korean voice search". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, pp. 5149–5152

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Neural networks ✓

- Learn non-linear dependencies, learn representations

Embeddings ✓

- Dense token representation

Neural machine translation ✓

- Sequence to sequence models
- Positional embeddings

Attention ✓

- Efficient long-range dependencies

Out-of-vocabulary words ✓

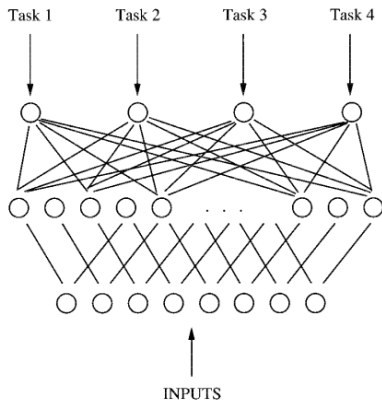
- WordPiece sub-word units can be truly multi-lingual and prevent OOV

Multi-task learning

Multi-task Learning

Approach to inductive transfer that improves generalization

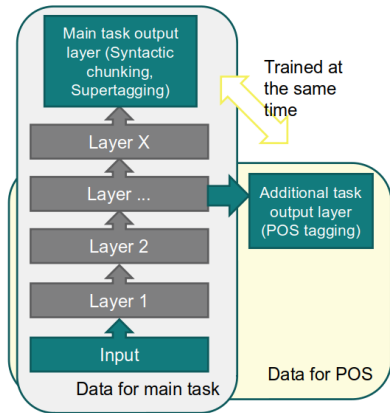
By learning tasks in parallel while using a shared representation



R. Caruana (1997). "Multi-task Learning". In: *Machine Learning* 28.1, pp. 41–75

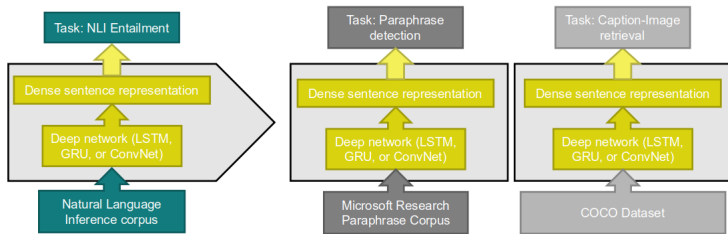
Multi-task learning in NLP

"In case we suspect the existence of a hierarchy between the different tasks, we show that it is worth-while to incorporate this knowledge in the MTL architecture's design, by making lower level tasks affect the lower levels of the representation."



A. Søgaard and Y. Goldberg (2016). "Deep multi-task learning with low level tasks supervised at lower layers". In: *Proceedings of ACL*. Berlin, Germany: Association for Computational Linguistics, pp. 231–235

Learn a sentence representation on a different task



"Models learned on NLI can perform better than models trained in unsupervised conditions or on other supervised tasks."¹¹

¹¹A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes (2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *Proceedings of EMNLP*. Copenhagen, Denmark, pp. 670–680

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Neural networks ✓

- Learn non-linear dependencies, learn representations

Embeddings ✓

- Dense token representation

Neural machine translation ✓

- Sequence to sequence models
- Positional embeddings

Attention ✓

- Efficient long-range dependencies

Out-of-vocabulary words ✓

- WordPiece sub-word units can be truly multi-lingual and prevent OOV

Multi-task learning ✓

- Shared representation improves generalization; transfer learning

“Unsupervised” Pre-Training

"Un-supervised" Pre-Training

- Deep neural nets are trained with full supervision
 - Even autoencoders are supervised by the reconstruction error
- "Unsupervised" training scenario usually means:
 - I don't have any labeled data for my target task (e.g., no labels for "word similarity")
 - But I can design a proxy supervised task (e.g., "given a context of a missing word, predict that word")
 - And create positive and negative instances by exploiting a large unlabeled corpus (e.g., words in their context as positive, and randomly swapped words with their context as negative)

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Neural networks ✓

- Learn non-linear dependencies, learn representations

Embeddings ✓

- Dense token representation

Neural machine translation ✓

- Sequence to sequence models
- Positional embeddings

Attention ✓

- Efficient long-range dependencies

Out-of-vocabulary words ✓

- WordPiece sub-word units can be truly multi-lingual and prevent OOV

Multi-task learning ✓

- Shared representation improves generalization; transfer learning

"Unsupervised"

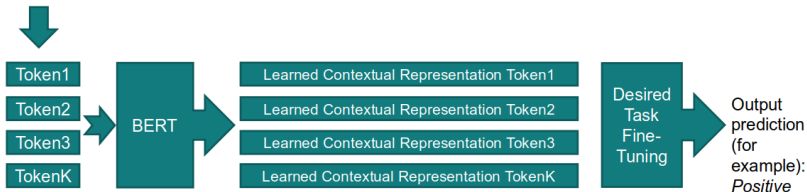
Pre-Training ✓

- Proxy task and unlimited data from unlabeled corpora

BERT

BERT: Very abstract view

Input text: *Lorem ipsum dolor*



BERT: Tokenization

Tokenizing into a multilingual WordPiece inventory

- Recall that WordPiece units are sub-word units
- 30,000 WordPiece units (newer models 110k units, 100 languages)

Implications: BERT can "consume" any language

BERT: Input representation

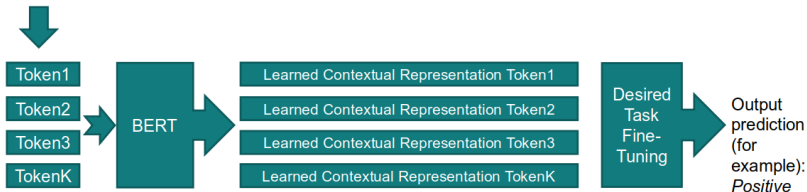
- Each WordPiece token from the input is represented by a **WordPiece embedding** (randomly initialized)
- Each position from the input is associated with a **positional embedding** (also randomly initialized)
- Input length limited to **512** WordPiece tokens, using **<PAD>**ding
- Special tokens
 - The first token is always a special token **[CLS]**
 - If the task involves two sentences (e.g., NLI), these two sentences are separated by a special token **[SEP]**; also special two **segment position embeddings**

BERT: Input representation summary

Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[\text{CLS}]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[\text{SEP}]}$	E_{he}	E_{likes}	E_{play}	$E_{\text{##ing}}$	$E_{[\text{SEP}]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

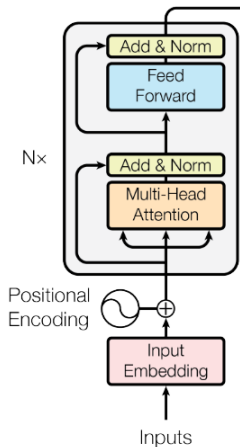
BERT: Very abstract view

Input text: *Lorem ipsum dolor*



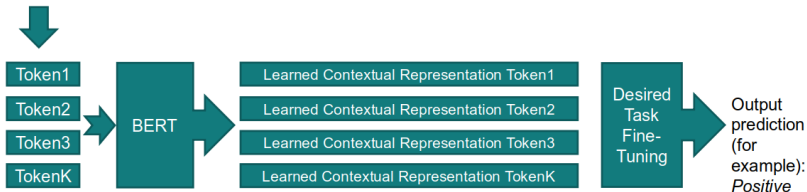
BERT: The Transformer

- The good-old-friend
"Self-Attention"
- Multiple parallel attention
"heads" (16 heads)
- With residual connections
- With layer normalization
- Stacked on top of each other
(24-times)
- 310,000,000 trainable
parameters
- ...we've seen that already

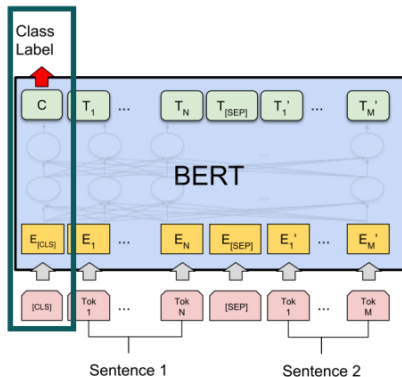


BERT: Very abstract view

Input text: *Lorem ipsum dolor*



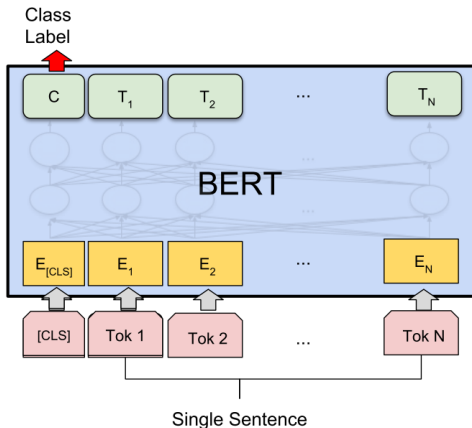
BERT: Representing various NLP tasks



(a) Sentence Pair Classification Tasks:
MNLI, QQP, QNLI, STS-B, MRPC,
RTE, SWAG

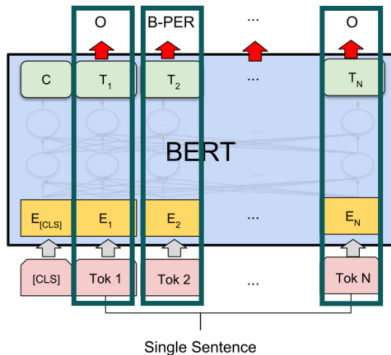
That explains the special [CLS] token at sequence start

BERT: Representing various NLP tasks



(b) Single Sentence Classification Tasks:
SST-2, CoLA

BERT: Representing various NLP tasks

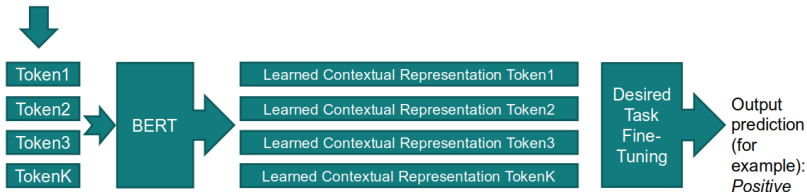


(d) Single Sentence Tagging Tasks:
CoNLL-2003 NER

Not conditioned on surrounding predictions

BERT: Very abstract view

Input text: *Lorem ipsum dolor*



BERT: "Unsupervised" multi-task pre-training

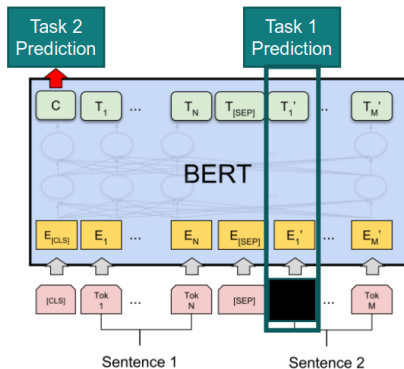
Prepare two auxiliary tasks that need no labeled data

Task 1: Cloze-test task

- Predict the masked WordPiece unit (multi-class, 30k classes)

Task 2: Consecutive segment prediction

- Did the second text segment appear after the first segment? (binary)



BERT: Pre-training data generation

Take the entire Wikipedia (in 100 languages; 2,5 billion words)

To generate a single training instance, sample two segments (max combined length 512 WordPiece tokens)

- For Task 2, replace the second segment randomly in 50% (negative samples)
- For Task 1, choose random 15% of the tokens, and in 80% replace with a [MASK]

BERT: Pre-training data – Simplified example

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]

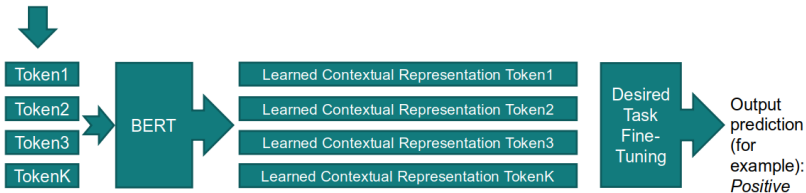
penguin [MASK] are flight ##less birds [SEP]

Label = NotNext

- <PAD>ding is missing
- The actual segments are longer and not necessarily actual sentences (just spans)
- The WordPiece tokens match full words / morphology well in this English text, but recall the ones we have seen before

BERT: Very abstract view

Input text: *Lorem ipsum dolor*



Pretraining this monster took them 4 days on 64 TPU chips (estimated \$500 USD)

Once pre-trained, transfer and "fine-tune" on your small-data task and get state-of-the-art results :)

Roadmap

NLP tasks ✓

- Long-range dependencies, hard to represent meaning

Neural networks ✓

- Learn non-linear dependencies, learn representations

Embeddings ✓

- Dense token representation

Neural machine translation ✓

- Sequence to sequence models
- Positional embeddings

Attention ✓

- Efficient long-range dependencies

Out-of-vocabulary words ✓

- WordPiece sub-word units can be truly multi-lingual and prevent OOV

Multi-task learning ✓

- Shared representation improves generalization; transfer learning

"Unsupervised"

Pre-Training ✓

- Proxy task and unlimited data from unlabeled corpora

BERT stays on the shoulders of many clever concepts and techniques, mastered into a single model

What do we know about how
BERT works?

Highly recommended reading

“BERTology has clearly come a long way, but it is fair to say we still have more questions than answers about how BERT works.”

A. Rogers, O. Kovaleva, and A. Rumshisky (2020). “A Primer in BERTology: What We Know About How BERT Works”. In: *Transactions of the Association for Computational Linguistics* 8, pp. 842–866

License and credits

Licensed under Creative Commons
Attribution-ShareAlike 4.0 International
(CC BY-SA 4.0)



Credits

Ivan Habernal

Content from ACL Anthology papers licensed under CC-BY
<https://www.aclweb.org/anthology>

References




Artemova, E., M. Apishev, D. Kirianov, V. Sarkisyan, S. Aksenov, and O. Serikov (2021). “Teaching a Massive Open Online Course on Natural Language Processing”. In: *Proceedings of the Fifth Workshop on Teaching NLP*. Online: Association for Computational Linguistics, pp. 13–27. URL: <https://www.aclweb.org/anthology/2021.teachingnlp-1.2>.






Caruana, R. (1997). “Multi-task Learning”. In: *Machine Learning* 28.1, pp. 41–75.



Conneau, A., D. Kiela, H. Schwenk, L. Barrault, and A. Bordes (2017). “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”. In: *Proceedings of EMNLP*. Copenhagen, Denmark, pp. 670–680.

-  Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of NAACL*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186.
-  Gehring, J., M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin (2017). “Convolutional Sequence to Sequence Learning”. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by D. Precup and Y. W. Teh. Sydney, Australia: PMLR, pp. 1243–1252.



-  Goldberg, Y. (2016). “A Primer on Neural Network Models for Natural Language Processing”. In: *Journal of Artificial Intelligence Research* 57, pp. 345–420.
-  Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press. URL: www.deeplearningbook.org.
-  Gururangan, S., S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith (2018). “Annotation Artifacts in Natural Language Inference Data”. In: *Proceedings of NAACL*. New Orleans, LA: Association for Computational Linguistics, pp. 107–112.





Koehn, P. (2017). “Neural Machine Translation”. In: *arXiv preprint*. URL:
<http://arxiv.org/abs/1709.07809>.



Krishnan, V. and C. D. Manning (2006). “An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition”. In: *Proceedings of ACL*. Sydney, Australia: Association for Computational Linguistics, pp. 1121–1128.

-  Rogers, A., O. Kovaleva, and A. Rumshisky (2020). “A Primer in BERTology: What We Know About How BERT Works”. In: *Transactions of the Association for Computational Linguistics* 8, pp. 842–866.
-  Schuster, M. and K. Nakajima (2012). “Japanese and Korean voice search”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, pp. 5149–5152.

-  Sennrich, R., B. Haddow, and A. Birch (2016). “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of ACL*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725.
-  Søgaard, A. and Y. Goldberg (2016). “Deep multi-task learning with low level tasks supervised at lower layers”. In: *Proceedings of ACL*. Berlin, Germany: Association for Computational Linguistics, pp. 231–235.



Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). “Attention Is All You Need”. In: *Advances in Neural Information Processing Systems 30*. Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008.



Wu, Y. et al. (2016). “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”. In: *arXiv*, pp. 1–23. URL: <http://arxiv.org/abs/1609.08144>.