Modern Language Models

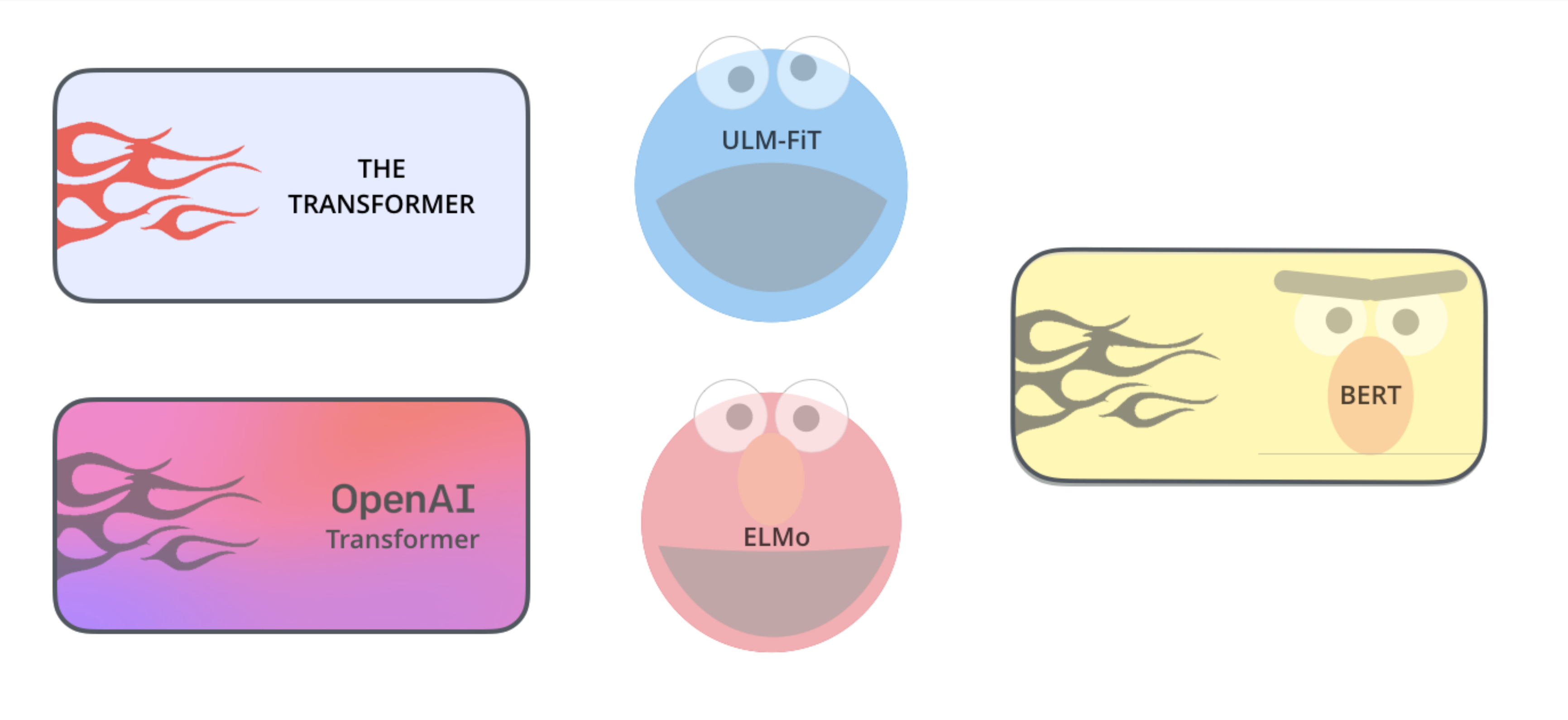
(ImageNet for NLP)

Емельянов Антон (king-menin)

31.01.2019

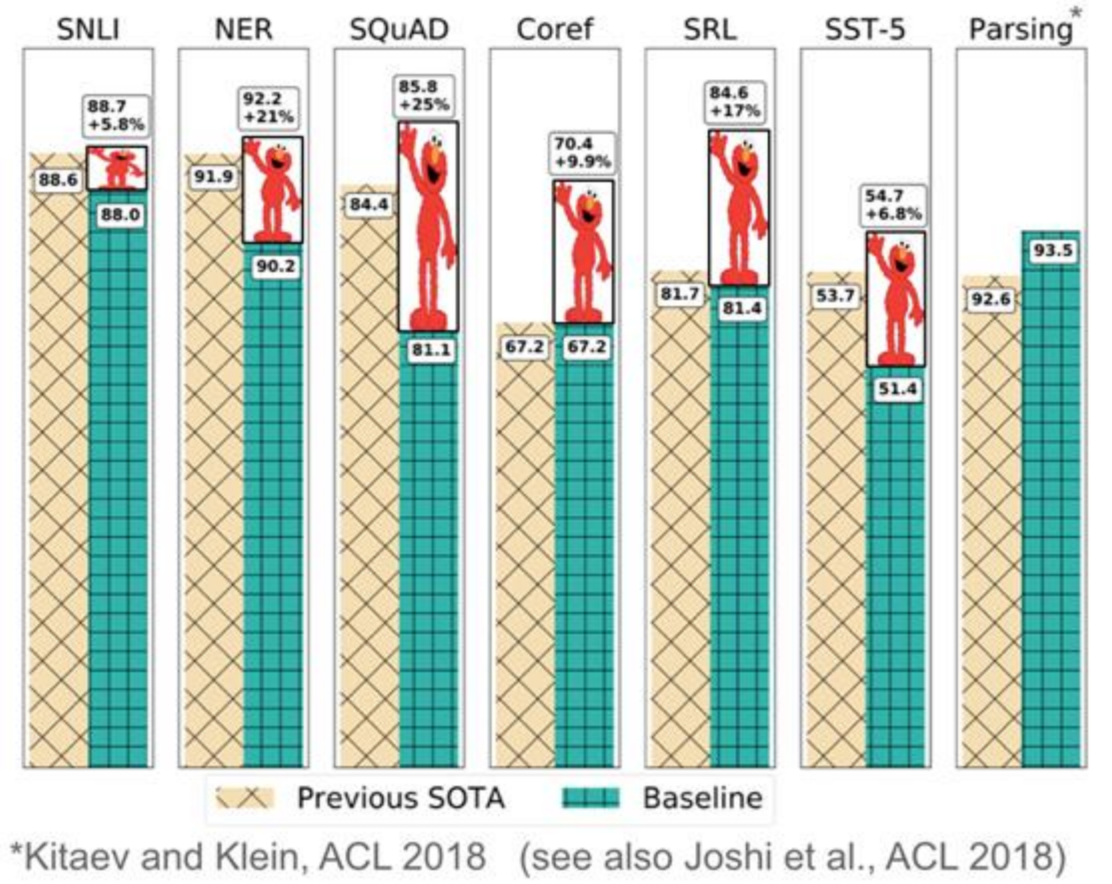
Plan

1. ELMo
2. ULMFiT
3. BERT
4. Anything else?



ELMo – Embeddings from Language MOdel

ELMo enables NLP models to better disambiguate between the correct sense of a given word. On in it's release it enabled near instant SOTA results in many downstream tasks, including tasks such as co-reference were previously not as viable for practical usage.

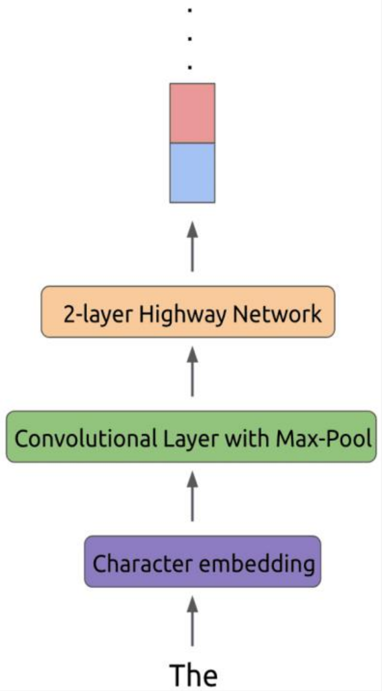


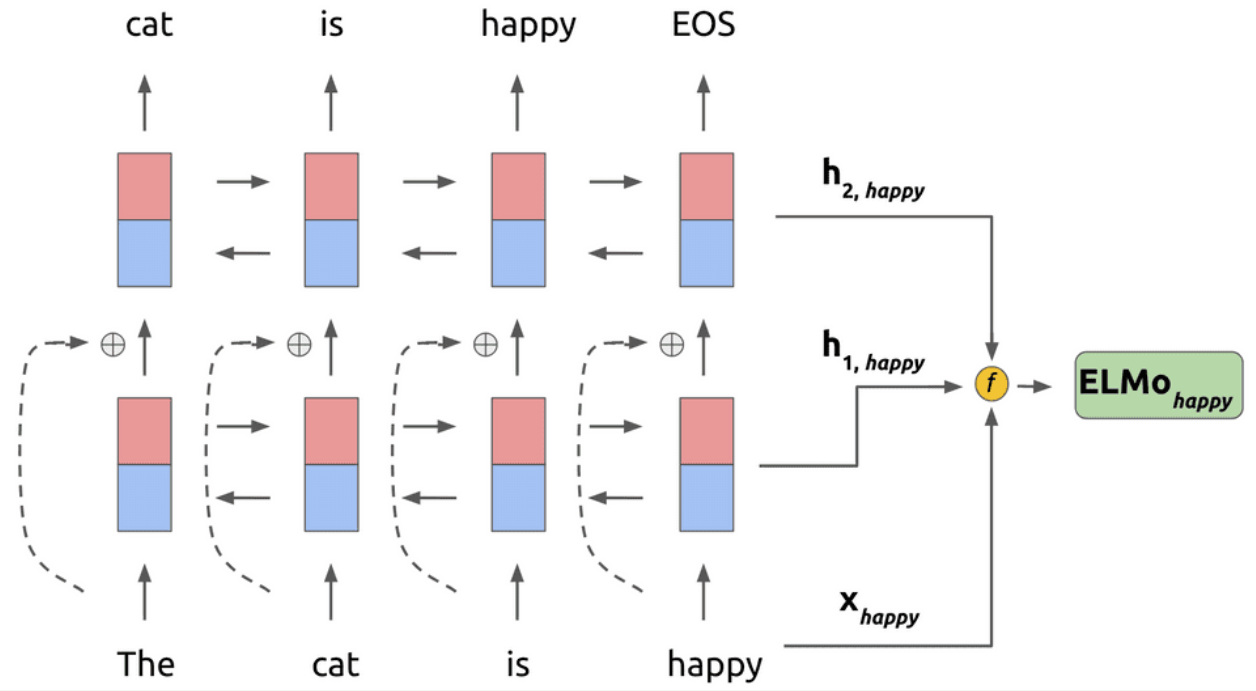
Deep contextualized word representations Matthew, E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, URL: https://arxiv.org/pdf/1802.05365.pdf

ELMo – В чем понт?

Architecture:

1. Character embedding (2048 convolutional filters, 2 highway projection 512 dim);
2. LM 2-Bi-LSTM (4096 units and 512 dim projection) with residual connection from first;





Deep contextualized word representations Matthew, E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, URL: https://arxiv.org/pdf/1802.05365.pdf

ELMo – В чем понт?

Architecture:

1. Character embedding (2048 convolutional filters, 2 highway projection 512 dim);
2. LM 2-Bi-LSTM (4096 units and 512 dim projection) with residual connection from first;
3. ELMo Output – *f function* (weighted sum of all layers)

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

here

- s_i represent softmax-normalized weights on the hidden representations from the language model;
- γ_k represents a task-specific scaling factor.

Deep contextualized word representations Matthew, E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, URL: https://arxiv.org/pdf/1802.05365.pdf

ELMo – В чем понт?

How ELMo is built?

- First off, the ELMo language model is trained on a sizable dataset: the 1B Word Benchmark.

- Note here that we learn a separate ELMo representation for each task (question answering, sentiment analysis, etc.) the model is being used for:

    1. First freeze weights in *f* and trained language model.

    2. The weighting factors are then learned during training of the task-specific model.

$$\mathbf{ELMo}_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} \mathbf{h}_{k,j}^{LM}$$

Deep contextualized word representations Matthew, E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, URL: https://arxiv.org/pdf/1802.05365.pdf

ULMFiT – Universal Language Model Fine-Tuning for Text Classification

Paper explores the benefits of using a pre-trained model on text classification. It proposes ULMFiT, a transfer learning method that can be applied to any task in NLP. SOTA results in classification task.

	Model	Test		Model	Test
IMDb	CoVe (McCann et al., 2017)	8.2	TREC-6	CoVe (McCann et al., 2017)	4.2
	oh-LSTM (Johnson and Zhang, 2016)	5.9		TBCNN (Mou et al., 2015)	4.0
	Virtual (Miyato et al., 2016)	5.9		LSTM-CNN (Zhou et al., 2016)	3.9
	ULMFiT (ours)	**4.6**		ULMFiT (ours)	**3.6**

Table 2: Test error rates (%) on two text classification datasets used by McCann et al. (2017).

	AG	DBpedia	Yelp-bi	Yelp-full
Char-level CNN (Zhang et al., 2015)	9.51	1.55	4.88	37.95
CNN (Johnson and Zhang, 2016)	6.57	0.84	2.90	32.39
DPCNN (Johnson and Zhang, 2017)	6.87	0.88	2.64	30.58
ULMFiT (ours)	**5.01**	**0.80**	**2.16**	**29.98**

Table 3: Test error rates (%) on text classification datasets used by Johnson and Zhang (2017).

Universal Language Model Fine-tuning for Text Classification, Jeremy Howard, Sebastian Ruder, URL: https://arxiv.org/pdf/1801.06146.pdf

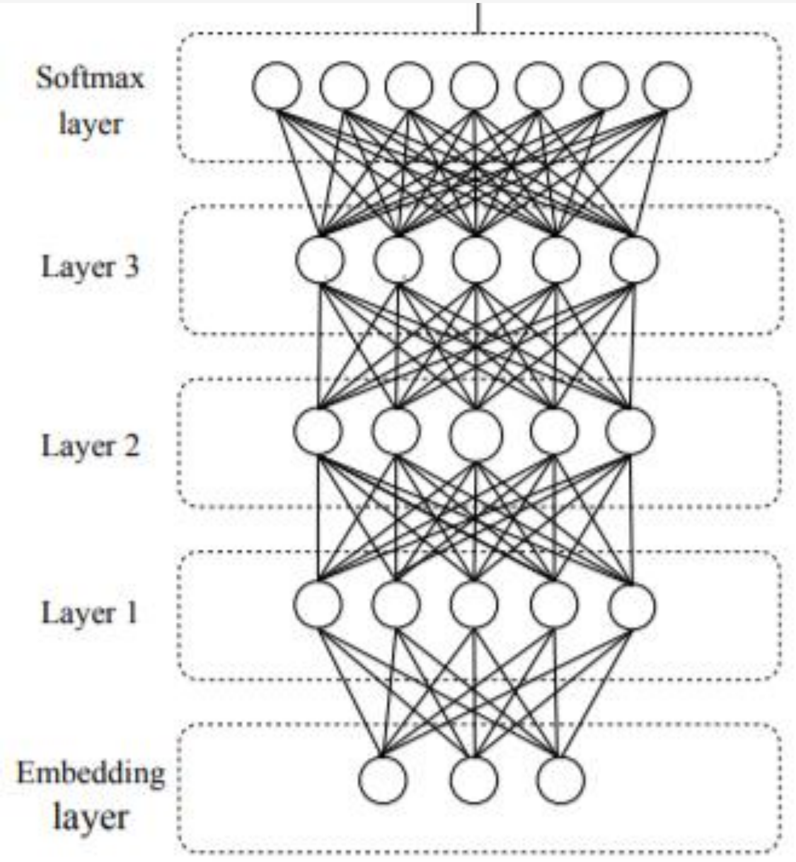
ULMFiT – В чем понт?

Architecture:

1. Word embeddings or BPE embeddings (400 dim);
2. LM: 3 Bi-AWD-LSTM (1150 projection dim) ;
3. Concat Pooling (for classification)

$$\mathbf{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_T\}:$$

$$\mathbf{h}_c = [\mathbf{h}_T, \mathtt{maxpool}(\mathbf{H}), \mathtt{meanpool}(\mathbf{H})]$$

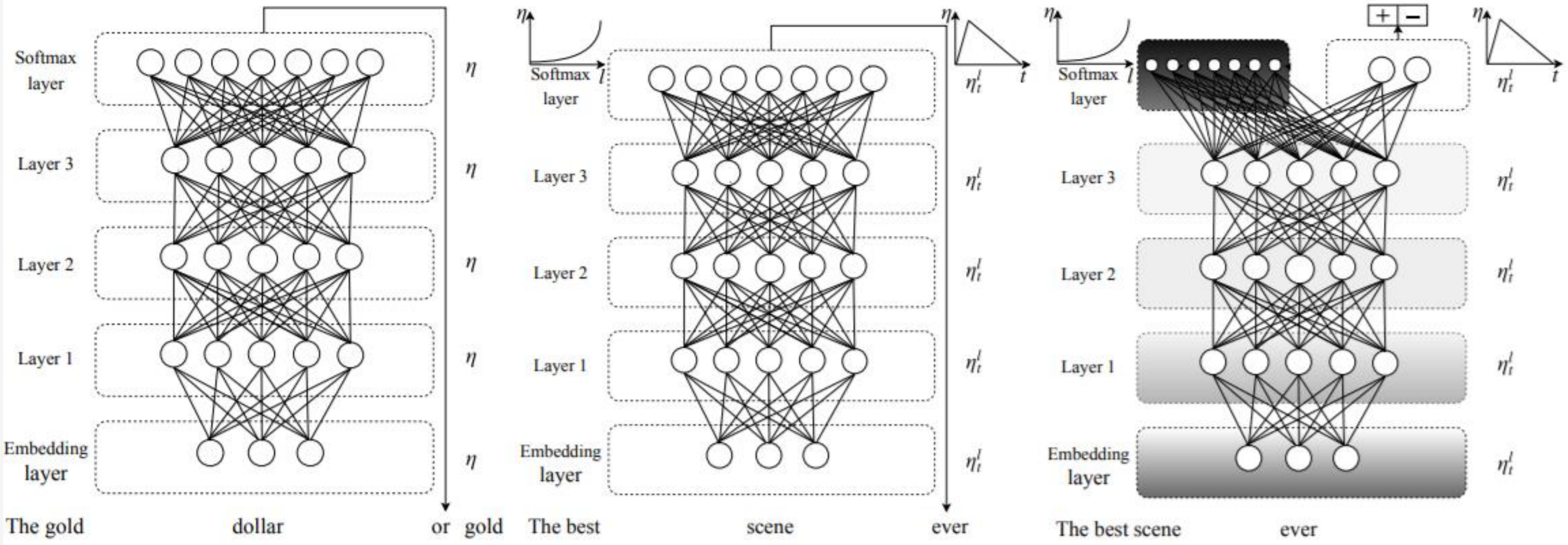


Universal Language Model Fine-tuning for Text Classification, Jeremy Howard, Sebastian Ruder, URL: https://arxiv.org/pdf/1801.06146.pdf

ULMFiT – В чем понт?

Train details?

- Learning PIPELINE

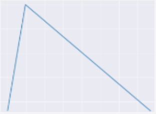


(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

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).

Universal Language Model Fine-tuning for Text Classification, Jeremy Howard, Sebastian Ruder, URL: https://arxiv.org/pdf/1801.06146.pdf

ULMFiT – В чем понт?



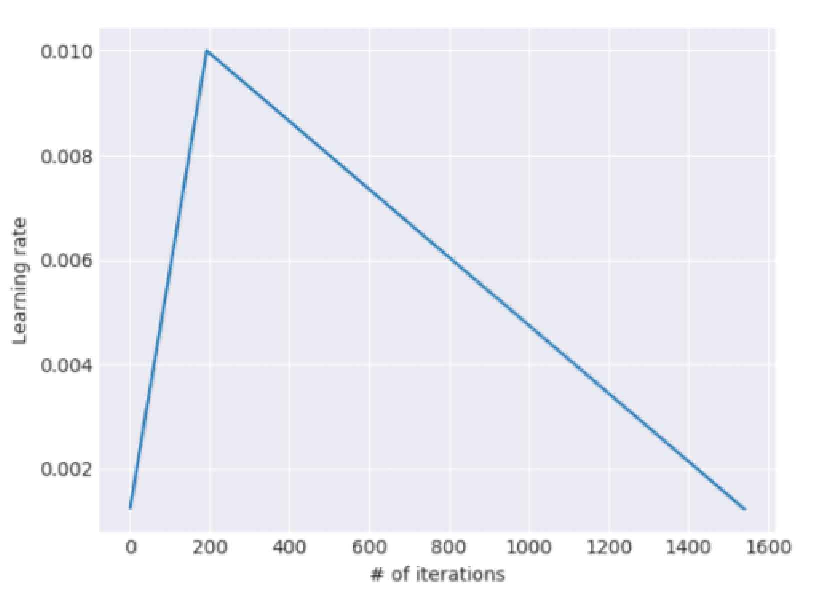
Train details?

- Slanted triangular learning rates (STLR)

$$cut = \lfloor T \cdot cut\_frac \rfloor$$

$$p = \begin{cases} t/cut, & \text{if } t < cut \\ 1 - \frac{t-cut}{cut \cdot (1/cut\_frac - 1)}, & \text{otherwise} \end{cases}$$

$$\eta_t = \eta_{max} \cdot \frac{1 + p \cdot (ratio - 1)}{ratio}$$



Here

- *T* is the number of training iterations,
- *cut_frac* is the fraction of iterations we increase the LR (0.1),
- *cut* is the iteration when we switch from increasing to decreasing the LR,
- *p* is the fraction of the number of iterations we have increased or will decrease the LR respectively,
- *ratio* specifies how much smaller the lowest LR is from the maximum LR (32),
- η_{max} is maximum LR (0.01).

Universal Language Model Fine-tuning for Text Classification, Jeremy Howard, Sebastian Ruder, URL: https://arxiv.org/pdf/1801.06146.pdf

ULMFiT – В чем понт?

Train details?

- Gradual unfreezing:
  - first unfreeze the last layer and fine-tune all unfrozen layers for one epoch;
  - then unfreeze the next lower frozen layer and repeat;
  - until we finetune all layers until convergence at the last iteration.

- BPTT for Text Classification (BPT3C):
  - divide the document into fixed length batches of size b;
  - at the beginning of each batch, the model is initialized with the final state of the previous batch;
  - keep track of the hidden states for mean and max-pooling;
  - gradients are back-propagated to the batches whose hidden states contributed to the final prediction.

Universal Language Model Fine-tuning for Text Classification, Jeremy Howard, Sebastian Ruder, URL: https://arxiv.org/pdf/1801.06146.pdf

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

New era. SOTA results on different tasks.

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	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. 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. OpenAI GPT=(L=12, H=768, A=12); $\text{BERT}_{\text{BASE}}$ =(L=12, H=768, A=12); $\text{BERT}_{\text{LARGE}}$ =(L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

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

New era. SOTA results on different tasks.

Table 2: SQuAD results.

The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev EM	Dev F1	Test EM	Test F1
Leaderboard (Oct 8th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
#1 Single - nlnet	-	-	83.5	90.1
#2 Single - QANet	-	-	82.5	89.3
Published				
BiDAF+ELMo (Single)	-	85.8	-	-
R.M. Reader (Single)	78.9	86.3	79.5	86.6
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$BERT_{BASE}$ (Single)	80.8	88.5	-	-
$BERT_{LARGE}$ (Single)	84.1	90.9	-	-
$BERT_{LARGE}$ (Ensemble)	85.8	91.8	-	-
$BERT_{LARGE}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$BERT_{LARGE}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

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

New era. SOTA results on different tasks.

System	Dev F1	Test F1
ELMo+BiLSTM+CRF	95.7	92.2
CVT+Multi (Clark et al., 2018)	-	92.6
$\text{BERT}_{\text{BASE}}$	96.4	92.4
$\text{BERT}_{\text{LARGE}}$	**96.6**	**92.8**

Table 3: CoNLL-2003 NER results. The hyperparameters were selected using the Dev set, and the reported Dev and Test scores are averaged over 5 random restarts using those hyperparameters.

BERT – Pre-training of Deep Bidirectional Transformers for Language Unders

New era. SOTA results on different tasks.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
BERT$_{\text{BASE}}$	81.6	-
BERT$_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

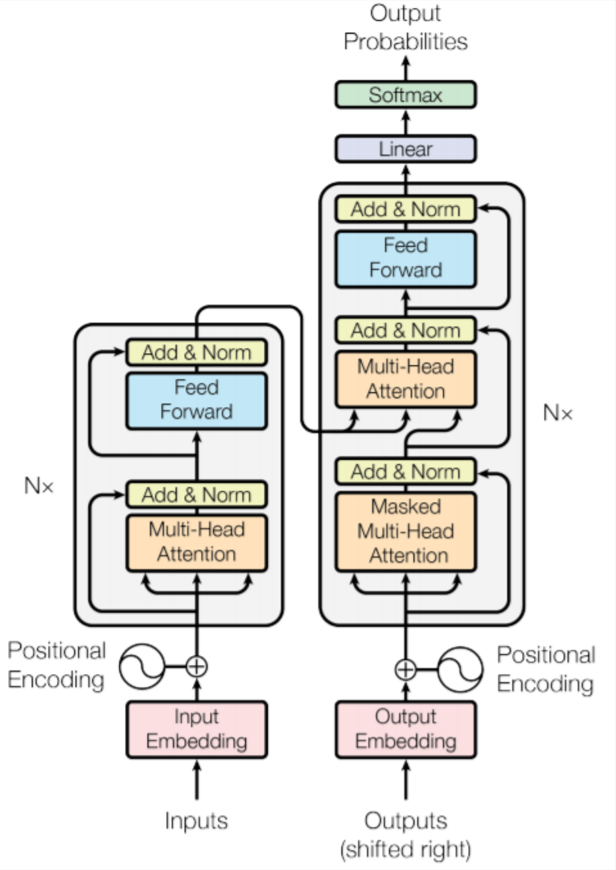
Table 4: SWAG Dev and Test accuracies. Test results were scored against the hidden labels by the SWAG authors. Human performance is measure with 100 samples, as reported in the SWAG paper.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

BERT – В чем понт?

Architecture:

BERT's model architecture is a multi-layer bidirectional Transformer encoder based on the original implementation.



RECAP Transformer.

Different BERT's models:

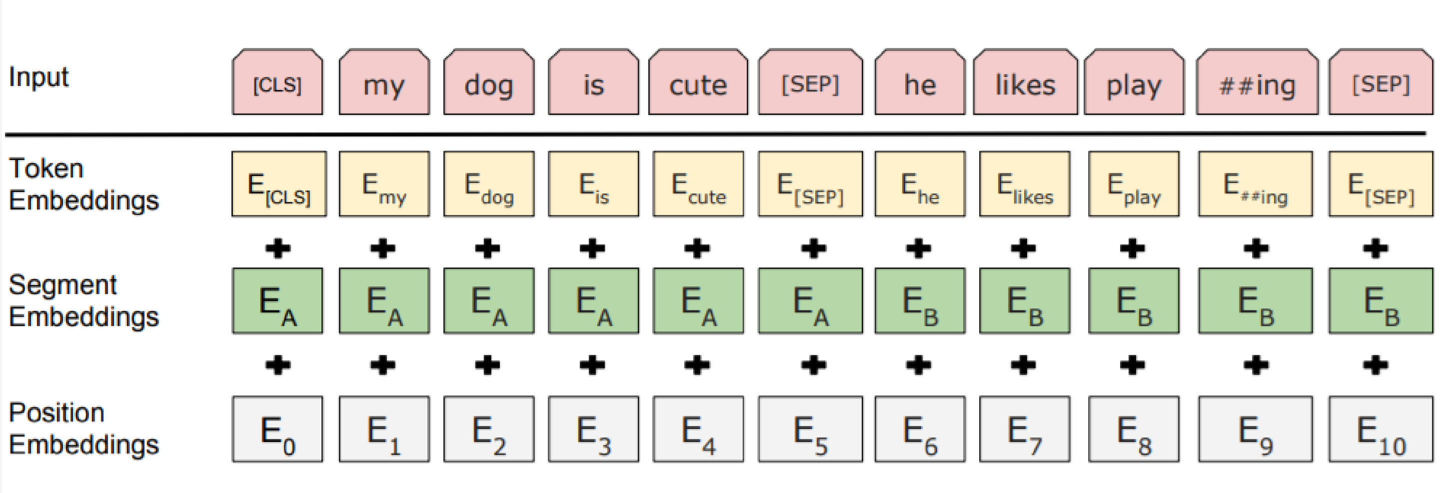
- $\text{BERT}_{\text{BASE}}$: L=12, H=768, A=12, Total Parameters=110M;
- $\text{BERT}_{\text{LARGE}}$: L=24, H=1024, A=16, Total Parameters=340M.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

BERT – В чем понт?

Architecture: Input embeddings

- Input representation is able to unambiguously represent both a single text sentence or a pair of text sentences (e.g., [Question, Answer]) in one token sequence.



- For a given token, its input representation is constructed by summing the corresponding token, segment and position embeddings:
  - Tokens: WordPiece (30 000 token vocabulary);
  - Positional (512 sequence length supported);

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

BERT – В чем понт?

Pre-training Tasks:

Task #1: Masked LM

- Rather than always replacing the chosen words with [MASK], the data generator will do the following:
  - 80% of the time: Replace the word with the [MASK] token, e.g.,

  my dog is hairy → my dog is [MASK]

  - 10% of the time: Replace the word with a random word, e.g.,

  my dog is hairy → my dog is apple

  - 10% of the time: Keep the word unchanged (the purpose of this is to bias the representation towards the actual observed word.), e.g.,

  my dog is hairy → my dog is hairy.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

BERT – В чем понт?

Pre-training Tasks:

Task #2: Next Sentence Prediction

- Binarized next sentence prediction. Length of sentence ≤ 512 tokens.
- Specifically, when choosing the sentences A and B for each pretraining example, 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.
- For 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*
- The final pre-trained model achieves 97%-98% accuracy at this task.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

BERT – В чем понт?

Pre-training Procedure:

Data:

- The concatenation of BooksCorpus (800M words);
- Text passages from English Wikipedia (2 500M words).

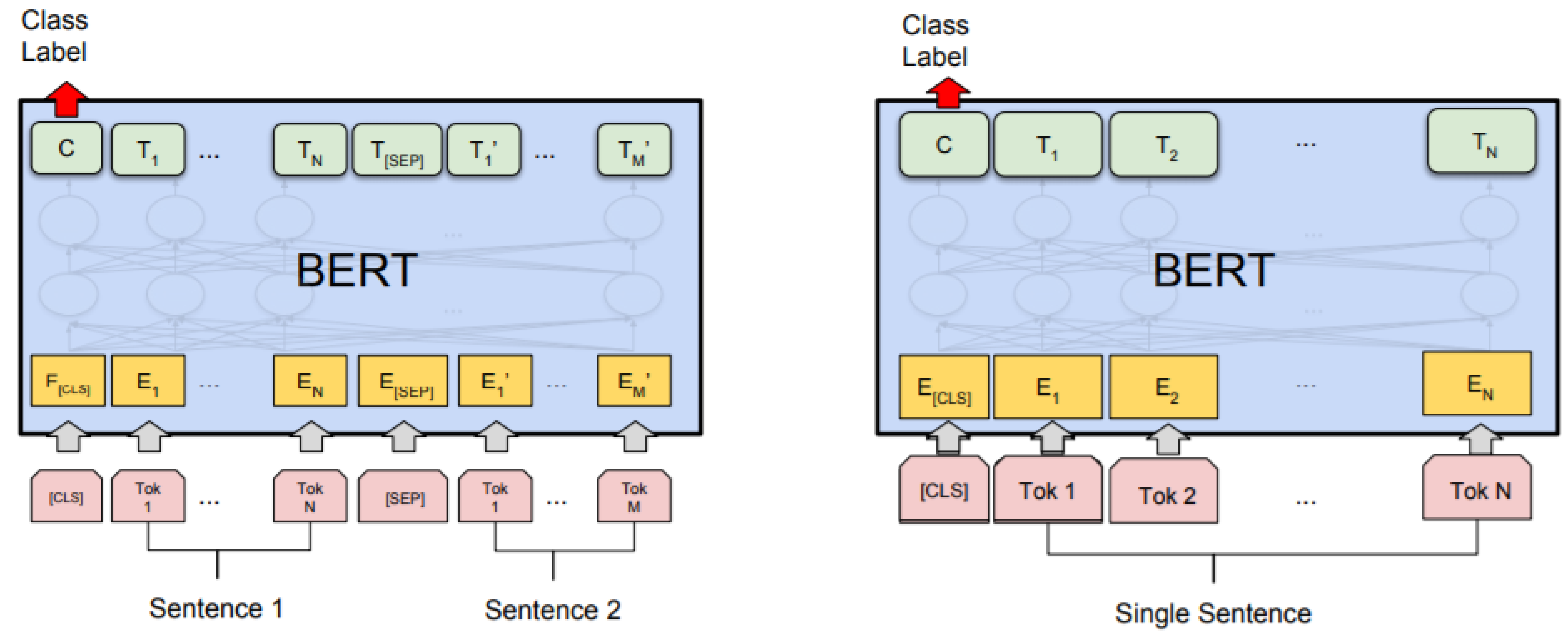
Generate each training input sequence:

- Sample two spans (refer to sentences)
  - The first sentence receives the A embedding and the second receives the B embedding. 50% of the time B is the actual next sentence that follows A and 50% of the time it is a random sentence.
  - The LM masking is applied after WordPiece tokenization with a uniform masking rate of 15%, and no special consideration given to partial word pieces.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

BERT – В чем понт?

Fine-tuning Procedure:



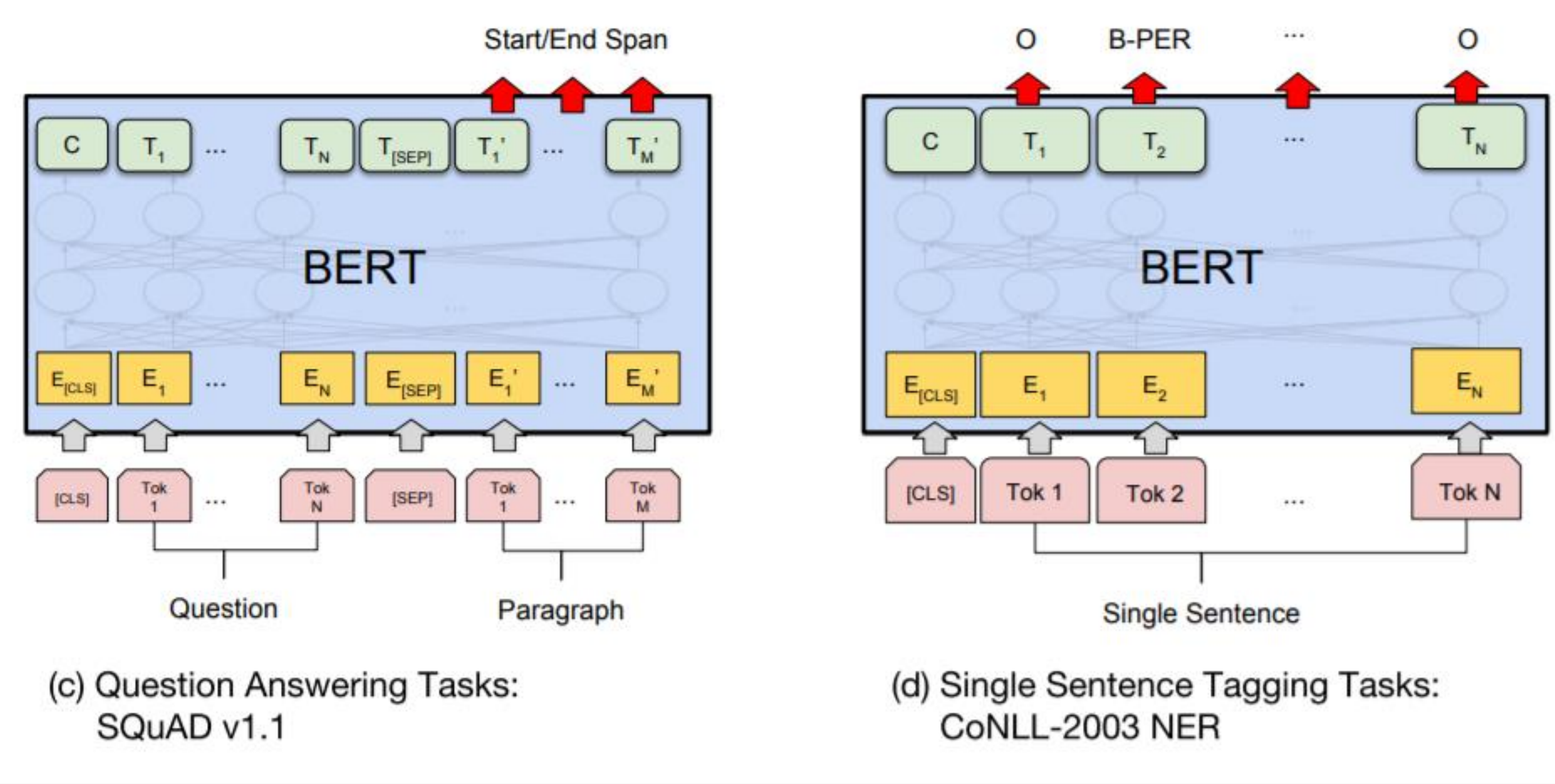
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

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

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

BERT – В чем понт?

Fine-tuning Procedure:



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language team, URL: https://arxiv.org/abs/1810.04805

~~Any~~ Nothing else (matter)?

NLP's ImageNet moment has arrived (by [Sebastian Ruder]())

- [OpenAI GPT]() (OpenAI)
- [Flair]() (Zalando research)
- [Transformer-XL: Unleashing the Potential of Attention Models]() (Google)
- [Zero-shot transfer across 93 languages: Open-sourcing enhanced LASER library]() (Facebook)
- Something (GLUE leaderboard)…

Rank	Name	Model	URL	Score	C
1	Microsoft D365 AI & MSR /	BIGBIRD		81.9	
2	Jacob Devlin	BERT: 24-layers, 1024-hidden, 16		80.4	

References

- ELMO: tf - ru, en; pytorch - for many languages
- ULMFit: pytorch - en
- Flair: pytorch - en chars
- BERT: tf - multilingual, pytorch - multilingual
- Transformer-XL tf/pytorch multilingual
- LASER – pytorch multilingual

Any Questions?

ЩЕЛКНИТЕ СТРЕЛКУ В РЕЖИМЕ СЛАЙД-ШОУ

Literature

ELMo:

1. ELMo was based on https://arxiv.org/pdf/1602.02410.pdf
2. Paper: https://arxiv.org/pdf/1802.05365.pdf
3. Explained: https://www.mihaileric.com/posts/deep-contextualized-word-representations-elmo/
4. HighWay layer: https://arxiv.org/pdf/1505.00387.pdf

ULMFit:

1. Paper: https://arxiv.org/pdf/1801.06146.pdf
2. Short explained: https://yashuseth.blog/2018/06/17/understanding-universal-language-model-fine-tuning-ulmfit/
3. AWD-LSTM: https://arxiv.org/pdf/1708.02182.pdf

Literature

BERT:

1. Paper: https://arxiv.org/pdf/1810.04805.pdf
2. Attention Is All You Need (Transformer): https://arxiv.org/pdf/1706.03762.pdf
3. Positional embedding: https://arxiv.org/pdf/1609.08144.pdf