# Bertology

## Sesame Street

Anton Emelyanov, Katya Artemova

ATP, MIPT; Computational Pragmatics Lab, HSE; Sberbank

March 19, 2020

# Today

# Transfer learning

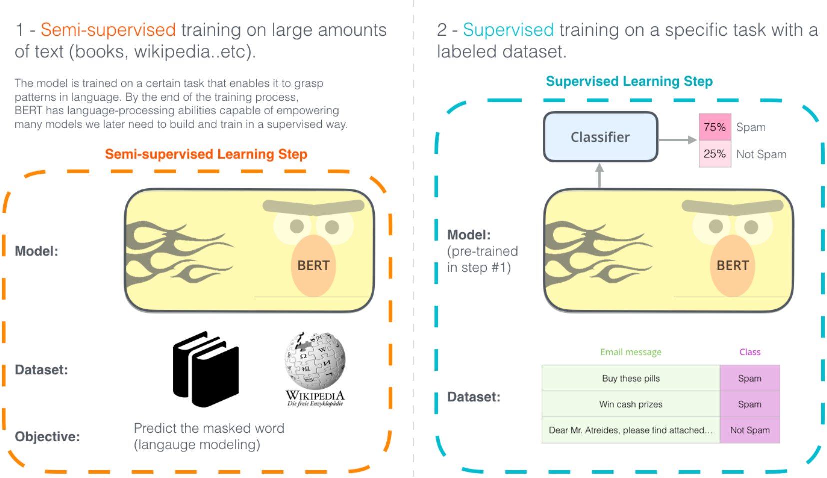



Figure: The two steps of how BERT is developed. You can download the model pre-trained in step 1 (trained on un-annotated data), and only worry about fine-tuning it for step 2.

# Today

# ELMo[1] - Embeddings from Language Models

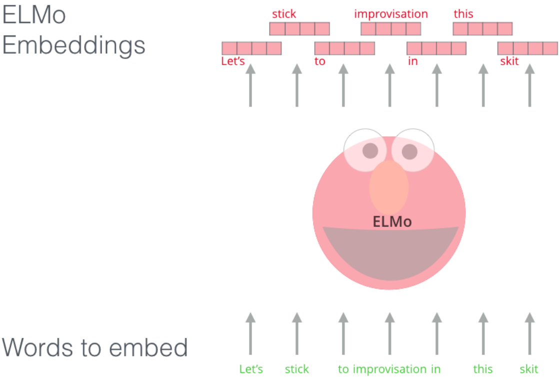



Figure: ELMo provided a significant step towards pre-training in the context of NLP. The ELMo LSTM would be trained on a massive dataset in the language of our dataset, and then we can use it as a component in other models that need to handle language.

# ELMo[1]

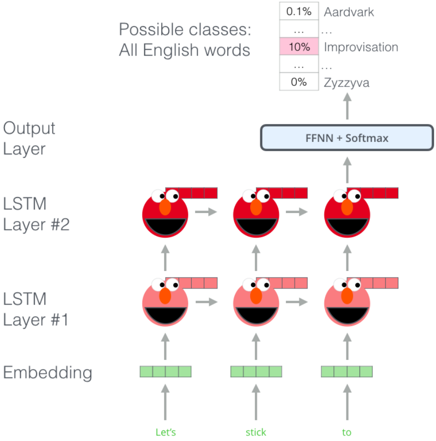



Figure: A step in the pre-training process of ELMo

# ELMo[1]

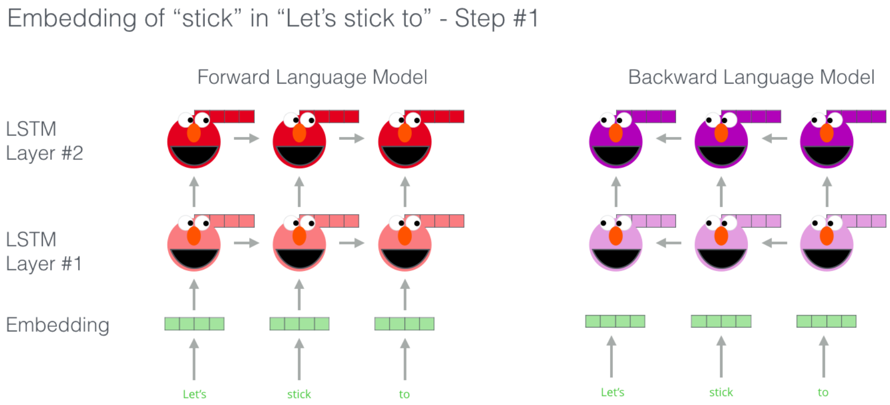



Figure: ELMo actually goes a step further and trains a bi-directional LSTM – so that its language model doesn't only have a sense of the next word, but also the previous word.

# ELMo[1]. Domain adaptation

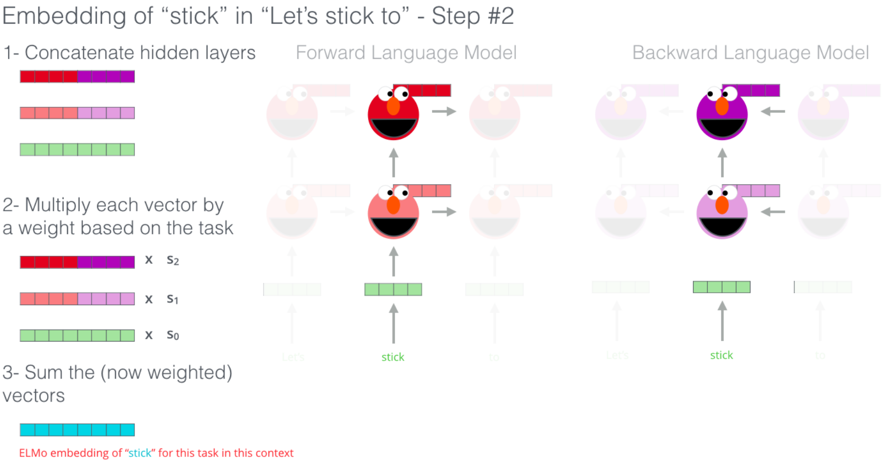



Figure: ELMo comes up with the contextualized embedding through grouping together the hidden states (and initial embedding) in a certain way (concatenation followed by weighted summation). Learn $s_0$, $s_1$, $s_2$ on other task.

# Today

# ULMFiT[2] - Universal Language Model Fine-tuning for Text Classification

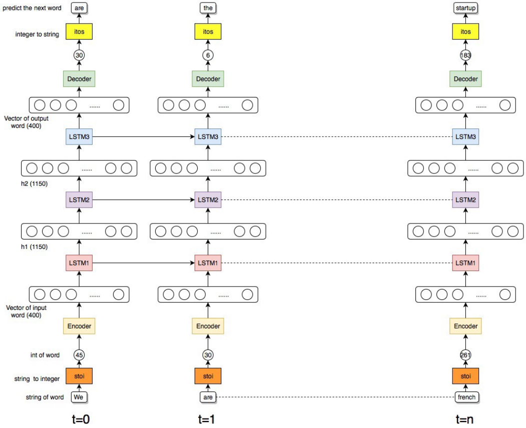
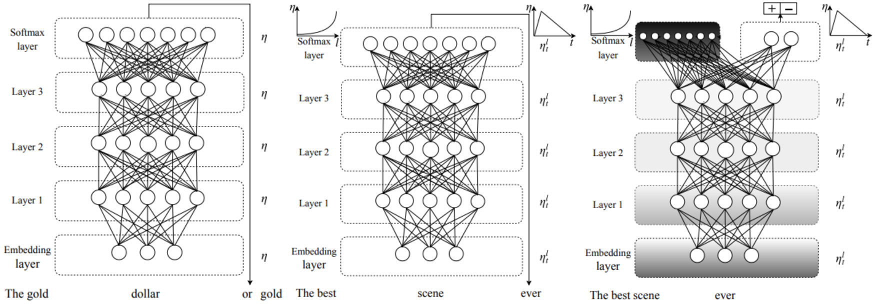



Figure: Structure of the language model.

# ULMFiT[2] involves 3 major stages




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

# STRL - Slanted triangular learning rates

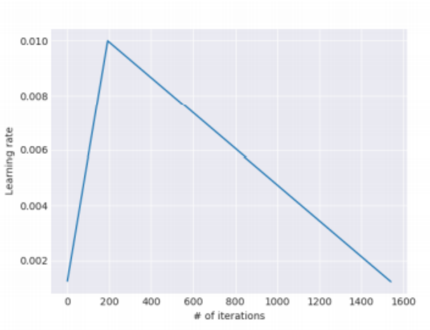



Figure: STRL example.

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

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

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

Where,

- T is number of training iterations
- cut_frac is the fraction of iterations
- 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 $\eta_{max}$
- $\eta_t$ is the learning rate at iteration t

# ULMFiT[2]. Classification

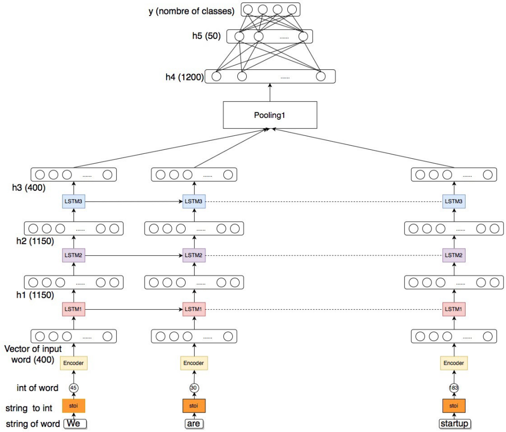



Figure: Classification architecture.

$$hc = [hT\ ,\ \mathrm{maxpool}(H),\ \mathrm{meanpool}(H)]$$

# Today

# BERT[3] - Bidirectional Encoder Representations from Transformers

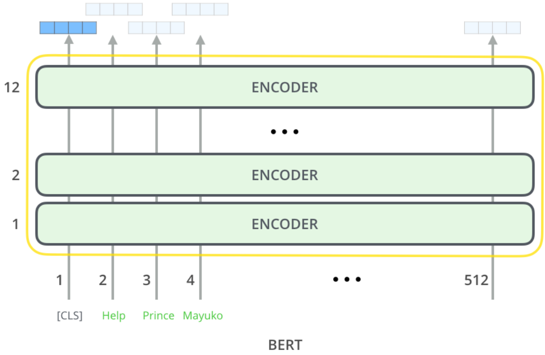



Figure: Model Architecture

Base architecture: (L= 12,H= 768,A= 12,110M params).

# BERT[3]. Input

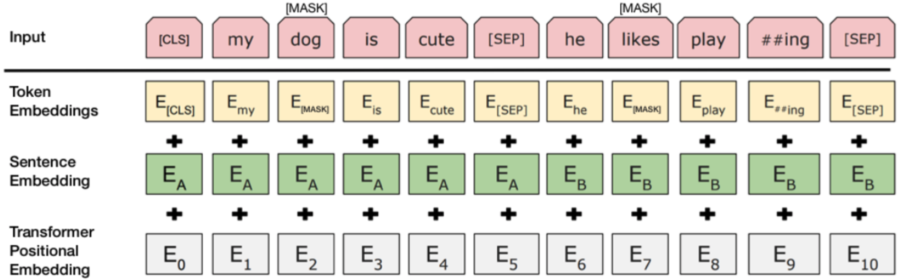



Figure: Input embeddings to transformer to transformers in BERT

# BERT[3]. MLM

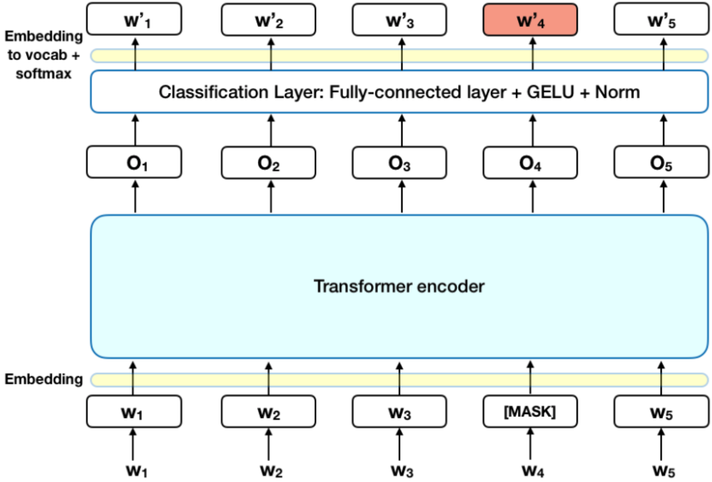



Figure: MLM prediction

# BERT[3]. MLM

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, e.g., *my dog is hairy* → *my dog is hairy*. The purpose of this is to bias the representation towards the actual observed word.

# BERT[3]. NSP

Next Sentence Prediction (NSP):

- Specifically, when choosing the sentences *A* and *B* for each pre-training example, 50% of the time B is the actual next sentence that follows *A* (labeled as *IsNext*), and 50% of the time it is a random sentence from the corpus (labeled as *NotNext*).

# BERT[3]. Domain adaptation

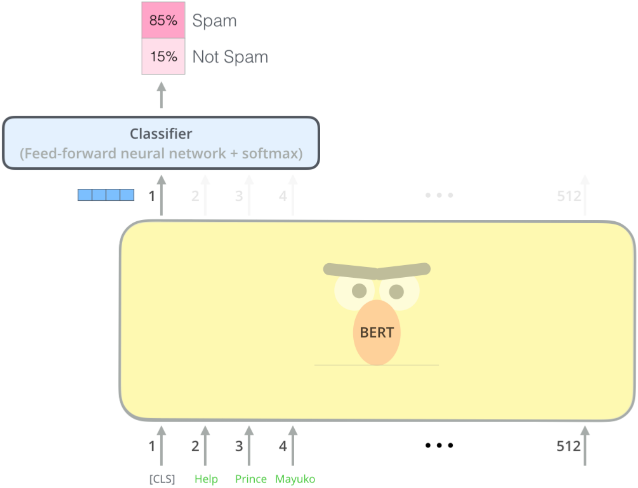
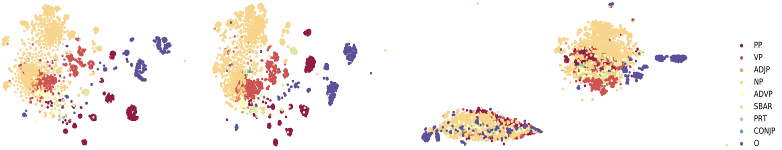



Figure: Example of classification with BERT.

# What does BERT learn about the structure of language?[4]

**Obervation 1**: BERT mostly captures phrase-level information in the lower layers and this information gets gradually diluted in higher layers




(a) Layer 1 (b) Layer 2 (c) Layer 11 (d) Layer 12

Figure: 2D t-SNE plot of span embeddings computed from the first and last two layers of BERT

# What does BERT learn about the structure of language?[4]

**Obervation 2**: BERT embeds a rich hierarchy of linguistic signals: surface information at the bottom, syntactic information in the middle, semantic information at the top

| Layer | SentLen (Surface) | WC (Surface) | TreeDepth (Syntactic) | TopConst (Syntactic) | BShift (Syntactic) | Tense (Semantic) | SubjNum (Semantic) | ObjNum (Semantic) | SOMO (Semantic) | CoordInv (Semantic) |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 93.9 (2.0) | 24.9 (24.8) | 35.9 (6.1) | 63.6 (9.0) | 50.3 (0.3) | 82.2 (18.4) | 77.6 (10.2) | 76.7 (26.3) | 49.9 (-0.1) | 53.9 (3.9) |
| 2 | 95.9 (3.4) | 65.0 (64.8) | 40.6 (11.3) | 71.3 (16.1) | 55.8 (5.8) | 85.9 (23.5) | 82.5 (15.3) | 80.6 (17.1) | 53.8 (4.4) | 58.5 (8.5) |
| 3 | **96.2 (3.9)** | 66.5 (66.0) | 39.7 (10.4) | 71.5 (18.5) | 64.9 (14.9) | 86.6 (23.8) | 82.0 (14.6) | 80.3 (16.6) | 55.8 (5.9) | 59.3 (9.3) |
| 4 | 94.2 (2.3) | **69.8 (69.6)** | 39.4 (10.8) | 71.3 (18.3) | 74.4 (24.5) | 87.6 (25.2) | 81.9 (15.0) | 81.4 (19.1) | 59.0 (8.5) | 58.1 (8.1) |
| 5 | 92.0 (0.5) | 69.2 (69.0) | 40.6 (11.8) | 81.3 (30.8) | 81.4 (31.4) | 89.5 (26.7) | 85.8 (19.4) | 81.2 (18.6) | 60.2 (10.3) | 64.1 (14.1) |
| 6 | 88.4 (-3.0) | 63.5 (63.4) | **41.3 (13.0)** | 83.3 (36.6) | 82.9 (32.9) | 89.8 (27.6) | **88.1 (21.9)** | 82.0 (20.1) | 60.7 (10.2) | 71.1 (21.2) |
| 7 | 83.7 (-7.7) | 56.9 (56.7) | 40.1 (12.0) | **84.1 (39.5)** | 83.0 (32.9) | 89.9 (27.5) | 87.4 (22.2) | **82.2 (21.1)** | 61.6 (11.7) | 74.8 (24.9) |
| 8 | 82.9 (-8.1) | 51.1 (51.0) | 39.2 (10.3) | 84.0 (39.5) | 83.9 (33.9) | 89.9 (27.6) | 87.5 (22.2) | 81.2 (19.7) | 62.1 (12.2) | 76.4 (26.4) |
| 9 | 80.1 (-11.1) | 47.9 (47.8) | 38.5 (10.8) | 83.1 (39.8) | **87.0 (37.1)** | **90.0 (28.0)** | 87.6 (22.9) | 81.8 (20.5) | 63.4 (13.4) | **78.7 (28.9)** |
| 10 | 77.0 (-14.0) | 43.4 (43.2) | 38.1 (9.9) | 81.7 (39.8) | 86.7 (36.7) | 89.7 (27.6) | 87.1 (22.6) | 80.5 (19.9) | 63.3 (12.7) | 78.4 (28.1) |
| 11 | 73.9 (-17.0) | 42.8 (42.7) | 36.3 (7.9) | 80.3 (39.1) | 86.8 (36.8) | 89.9 (27.8) | 85.7 (21.9) | 78.9 (18.6) | 64.4 (14.5) | 77.6 (27.9) |
| 12 | 69.5 (-21.4) | 49.1 (49.0) | 34.7 (6.9) | 76.5 (37.2) | 86.4 (36.4) | 89.5 (27.7) | 84.0 (20.2) | 78.7 (18.4) | **65.2 (15.3)** | 74.9 (25.4) |

Figure: Probing task performance for each BERT layer

Probing tasks: predict sentence length, presence of words, test for sensitivity to word order, the depth of the syntactic tree, top level constituents in the syntax tree, check for the tense, the subject number, the sensitivity to random replacement of a noun/verb.

# What does BERT learn about the structure of language?[4]

**Obervation 3**: BERT requires deeper layers to model long-range dependency information

| **Layer** | **0** (1.5) | **1** (5.2) | **2** (7.7) | **3** (10.5) | **4** (13.3) |
|---|---|---|---|---|---|
| 1 | 90.89 | 40.43 | 23.22 | 21.46 | 20 |
| 2 | 92.01 | 42.6 | 25.84 | 24.78 | 26.02 |
| 3 | 92.77 | 47.05 | 29.77 | 27.22 | 29.56 |
| 4 | 94.39 | 52.97 | 33.02 | 29.13 | 30.09 |
| 5 | 94.98 | 63.12 | 43.68 | 36.61 | 36.11 |
| 6 | 95.45 | 67.28 | 46.93 | 38.22 | 36.46 |
| 7 | 95.52 | 72.44 | 53.03 | 43.5 | 41.06 |
| 8 | **95.68** | **75.66** | **58.74** | 48.88 | 45.49 |
| 9 | 95.54 | 73.84 | 57.96 | **50.34** | **48.85** |
| 10 | 95.09 | 69.21 | 51.5 | 43.26 | 41.59 |
| 11 | 94.33 | 66.62 | 51.69 | 46.09 | 42.65 |
| 12 | 94.06 | 62.78 | 51.07 | 46.04 | 46.37 |

Figure: Subject-verb agreement scores for each BERT

The task of predicting the verb number becomes harder when there are more nouns with opposite number intervening between the subject and the verb.

# Today

# RoBERTa: A Robustly Optimized BERT Pretraining Approach [5]

RoBERTa is an improved pretraining procedure for BERT. It is trained with:

1. dynamic masking
2. on full sentences that may cross document boundaries (a special separator token is added) without NSP loss
3. with larger batches (2K, 8K)
4. a larger byte-level BPE

on approx. 160GB uncompressed unannotated texts.
RoBERTa achieves state-of-the-art results on GLUE, RACE and SQuAD, without multi-task finetuning for GLUE or additional data for SQuAD.

# ALBERT: A Lite BERT for Self-supervised Learning of Language Representations [6]

1. factorized embedding parameterization: in orginal model, $E = H$. Here the one-hot vectors are projected into a lower dimensional embedding space of size $E$, and then project it to the hidden space $H$
2. cross-layer parameter sharing: FFN parameters and Attention parameters are shared
3. inter-sentence coherence loss: sentence order predictions $(< s_1, s_2, 1 >, < s_2, s_1, 0 >)$

ALBERT has about 18x fewer parameters compared to BERT.
ALBERT-xxlarge (1M params) outperforms both BERT and RoBERTa.

# DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter [7]

Knowledge distillation is a compression technique in which a compact model - the student, DistilBERT - is trained to reproduce the behaviour of a larger model - the teacher, BERT.

1. a linear combination of the distillation loss $L_{ce} = t_i \log(s_i)$ , $L_{mlm}$ and cosine embedding loss $L_{cos}$
2. student architecture: same architecture, but $H/2$ hidden layers
3. the student is initialized from the teacher by taking one layer out of two

DistilBERT compares surprisingly well to BERT, retaining 97% of the performance with 40% fewer parameters on GLUE

# BERT & Family

1. More data and even more GPU! ... this approach is heavily criticized
2. GLUE benchmarking leads to ensemble training and a lot of fine tuning
3. BERT ancestors share a lot: data sources, NSP loss is omitted
4. MLM remains the core objective
5. BUT all of these models lack of common sense! understanding
6. BERT separately reconstructs all masked tokens
7. Pretrain-finetune discrepancy: the input contains artificial symbols like [MASK] that never occur in downstream tasks

# Pretrained language model based on BERT

1. BioBERT [8]
2. ClinicalBERT [9]
3. SciBERT [10]

There is no surprise, all these models show significant improvement over current SoTA.

Experiment design:

1. to create a new vocabulary or not
2. to train the model from scratch or to fine-tune BERT

# Today

# AR vs AE LMs

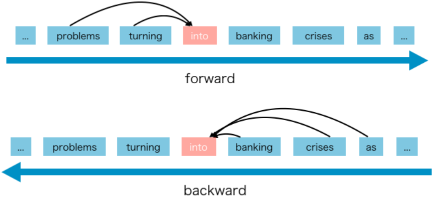



Figure: Autoregressive language model predicts the word based on its left (or right) context

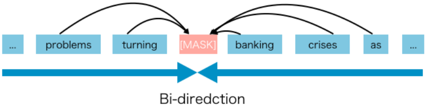



Figure: Aautoencoder language model aims to reconstruct the original data from corrupted input

# XLNet: Generalized Autoregressive Pretraining for Language Understanding [11]

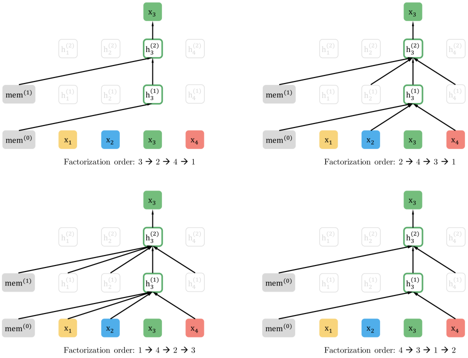



Figure: Permutation language model objective

# XLNet: Generalized Autoregressive Pretraining for Language Understanding [11]

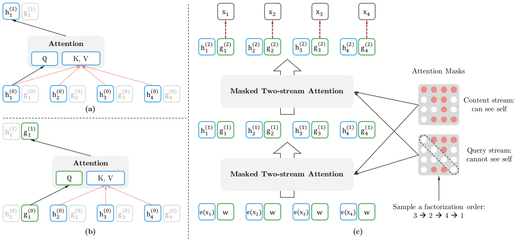



Figure: Architecture overview

# XLNet: Generalized Autoregressive Pretraining for Language Understanding [11]

- XLNet significantly improves upon BERT on 20 tasks
- It costs $245,000 to train the XLNet model

| | BERT | RoBERTa | DistilBERT | XLNet |
|---|---|---|---|---|
| **Size (millions)** | **Base**: 110<br>**Large**: 340 | **Base**: 110<br>**Large**: 340 | **Base:** 66 | **Base**: ~110<br>**Large**: ~340 |
| **Training Time** | **Base**: 8 x V100 x 12 days*<br>**Large:** 64 TPU Chips x 4 days (or 280 x V100 x 1 days*) | **Large:** 1024 x V100 x 1 day; 4-5 times more than BERT. | **Base:** 8 x V100 x 3.5 days; 4 times less than BERT. | **Large:** 512 TPU Chips x 2.5 days; 5 times more than BERT. |
| **Performance** | Outperforms state-of-the-art in Oct 2018 | 2-20% improvement over BERT | 3% degradation from BERT | 2-15% improvement over BERT |
| **Data** | 16 GB BERT data (Books Corpus + Wikipedia).<br>3.3 Billion words. | 160 GB (16 GB BERT data + 144 GB additional) | 16 GB BERT data.<br>3.3 Billion words. | **Base**: 16 GB BERT data<br>**Large**: 113 GB (16 GB BERT data + 97 GB additional).<br>33 Billion words. |
| **Method** | BERT (Bidirectional Transformer with MLM and NSP) | BERT without NSP** | BERT Distillation | Bidirectional Transformer with Permutation based modeling |

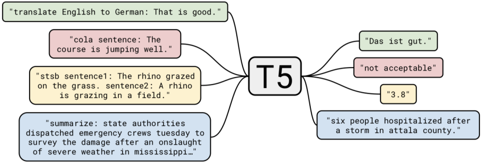
Image source: towardsdatascience

# Today

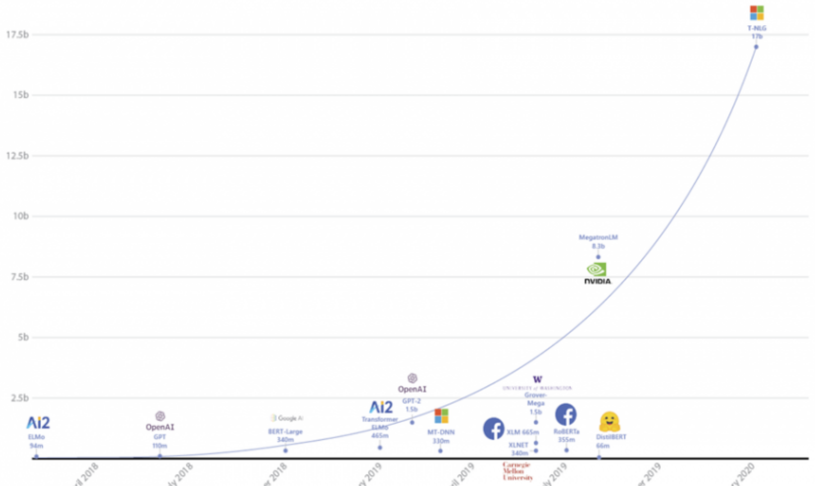
# T5: Text-To-Text Transfer Transformer [12]

In short:

1. NMT Transfromer architecture
2. new SoTA on GLUE
3. an extensive study of transfer learning techniques (a must read!)



# Bonus: more parameters



# Reference I

M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *ArXiv*, vol. abs/1802.05365, 2018.

J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *ACL*, 2018.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019.

G. Jawahar, B. Sagot, D. Seddah, S. Unicomb, G. Iñiguez, M. Karsai, Y. Léo, M. Karsai, C. Sarraute, É. Fleury, *et al.*, "What does bert learn about the structure of language?" In *57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy*, 2019.

# Reference II


- Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
- Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.
- V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter*, 2019. arXiv: `1910.01108 [cs.CL]`.
- J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: Pre-trained biomedical language representation model for biomedical text mining," *arXiv preprint arXiv:1901.08746*, 2019.

# Reference III

E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical bert embeddings," *arXiv preprint arXiv:1904.03323*, 2019.

I. Beltagy, K. Lo, and A. Cohan, *Scibert: A pretrained language model for scientific text*, 2019. arXiv: 1903.10676 [cs.CL].

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, *Xlnet: Generalized autoregressive pretraining for language understanding*, 2019. arXiv: 1906.08237 [cs.CL].

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2019. arXiv: 1910.10683 [cs.LG].