TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2019

Pretraining

Pretraining

We train a model on a large data set and then either use the model as features, or fine-tune the model, on a different problem typically with a smaller training set.

Pretraining

Pretraining can be either supervised or self-supervised.

Supervised: Trained on human annotations done explicitly for the purpose of training machine learning models. Examples include imagenet training and training parsers on the Penn tree bank.

Self-Supervised: Anything not involving human annotation. Examples include language modeling (e.g. BERT) and image colorization.

Pretraining

Pretraining is currently essential for achieving state of the art (SOTA) performance on both NLP and vision benchmarks.

NLP SOTA uses language modeling (self-supervised) pretraining.

Vision SOTA still uses ImageNet pretraining, although self-supervised pretraining is making advances.

ImageNet Pretraining in Vision

CBNet: A Novel Composite Backbone Network Architecture for Object Detection Liu et al., Sept. 2019 (COCO leader as of February 26, 2019).

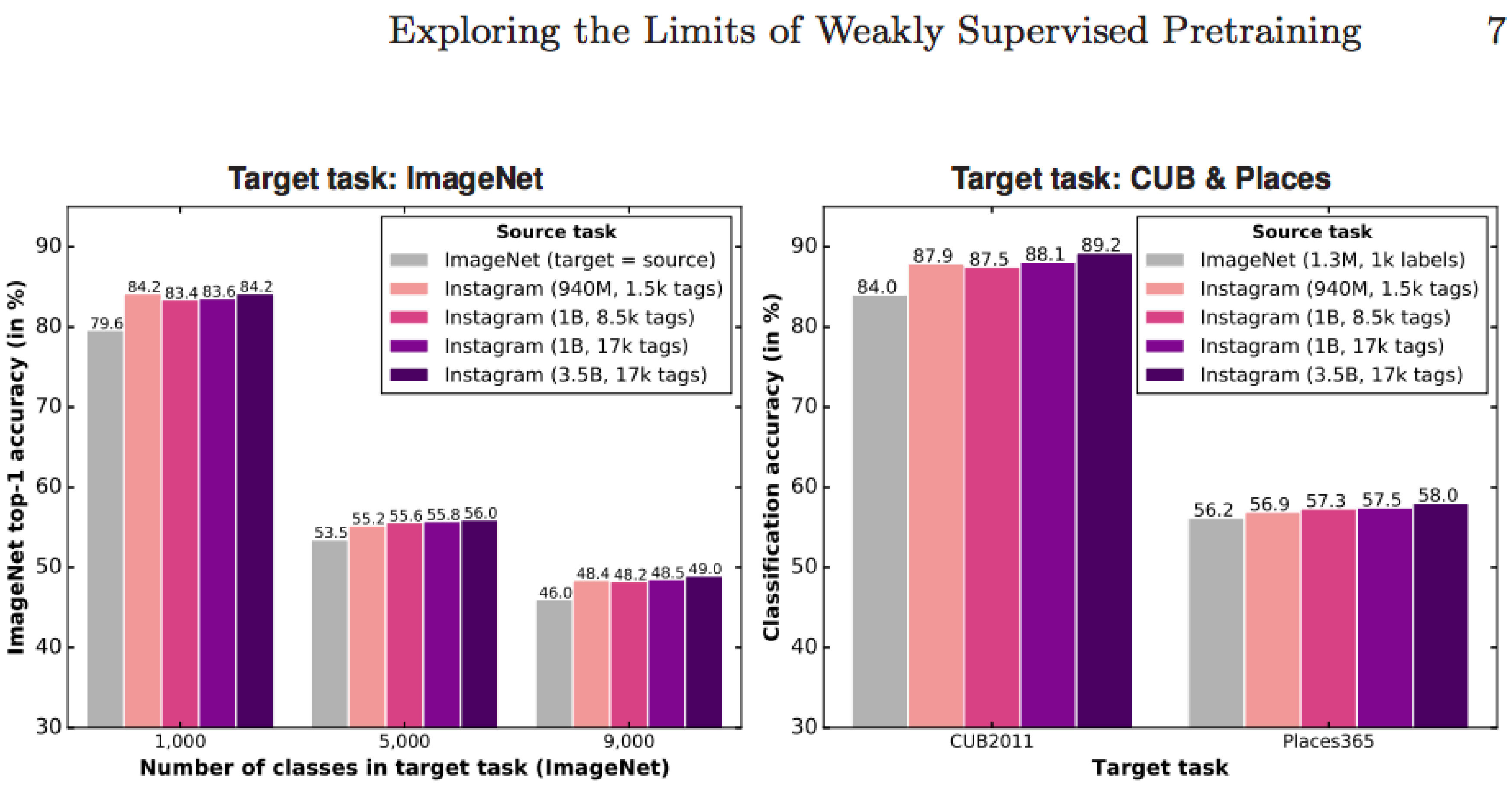
> Generally speaking, in a typical CNN based object detector, a backbone network is used to extract basic features for detecting objects, which is usually designed for the image classification task and pretrained on the ImageNet dataset (Deng et al. 2009).

Instagram Pretraining, Mahajan et al., May 2018

In our experiments, we train standard convolutional network architectures to predict hashtags on up to 3.5 billion public Instagram images.

To make training at this scale practical, we adopt a distributed synchronous implementation of stochastic gradient descent with large (8k image) minibatches, following Goyal et al. 2017.

Instagram Pretraining

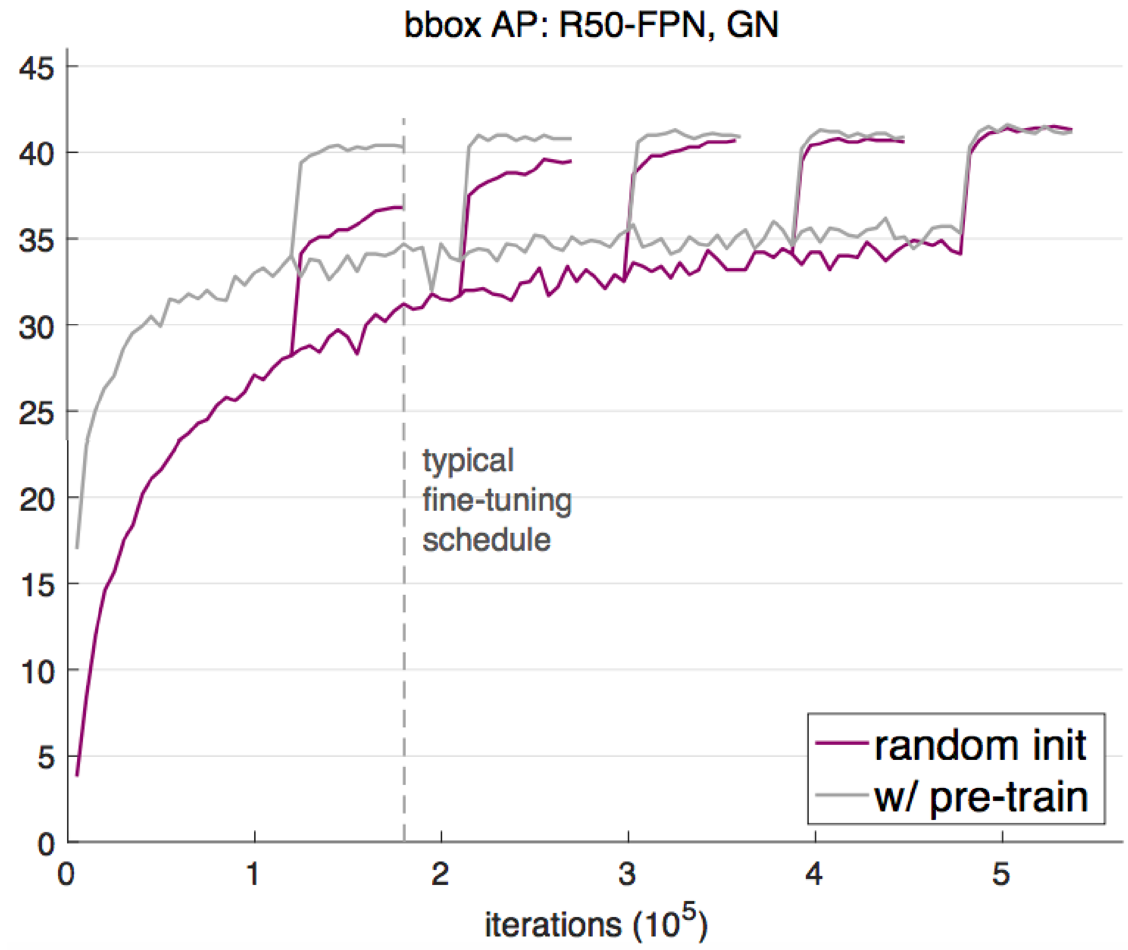


Exploring the Limits of Weakly Supervised Pretraining 7

Rethinking ImageNet Pretraining, He et al., Nov. 2018

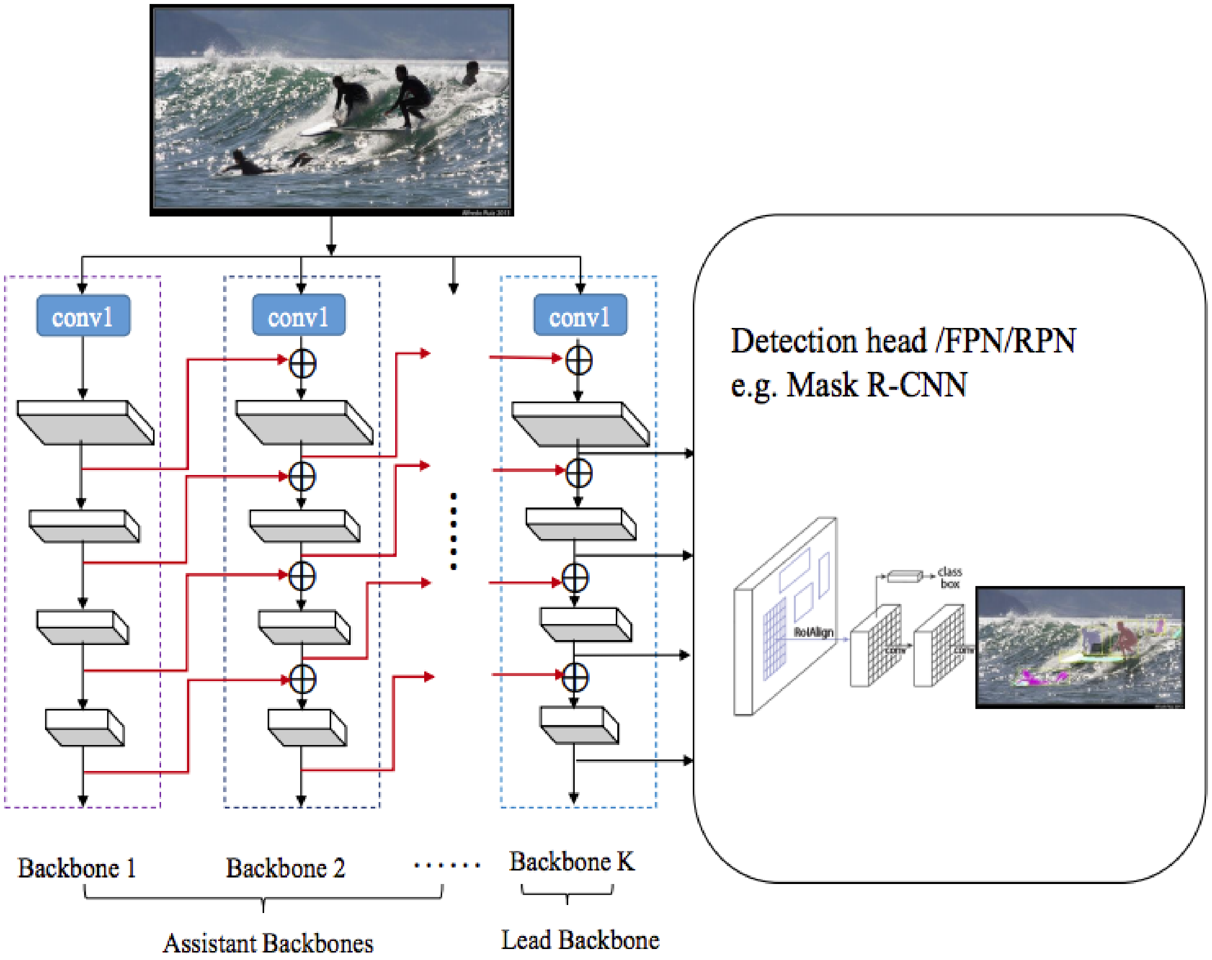
We report competitive results on object detection and instance segmentation on the COCO dataset using standard models trained from **random initialization**.

Rethinking ImageNet Pretraining, He et al., Nov. 2018



faster but not better in a long time

CBNet (COCO leader as of Feb, 2020)



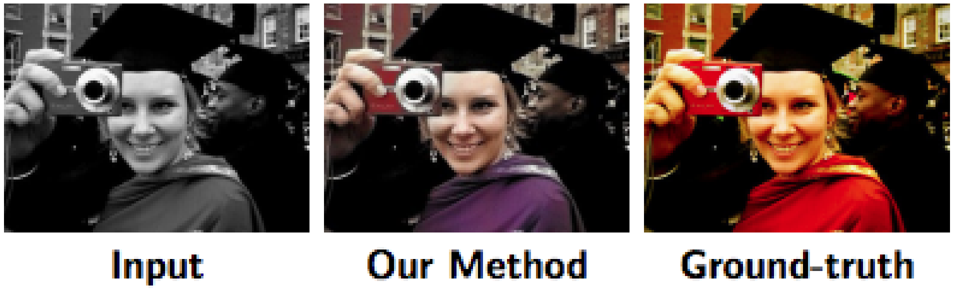
CBNet (COCO leader as of Feb, 2020)

We initialize each assembled backbone of CBNet with the pretrained model of the single backbone which is widely and freely available today, such as ResNet and ResNeXt

Self-Supervised Feature Learning for Vision

Learning Representations for Colorization

Larsson et al., 2016



x $y_\Phi(x)$ y

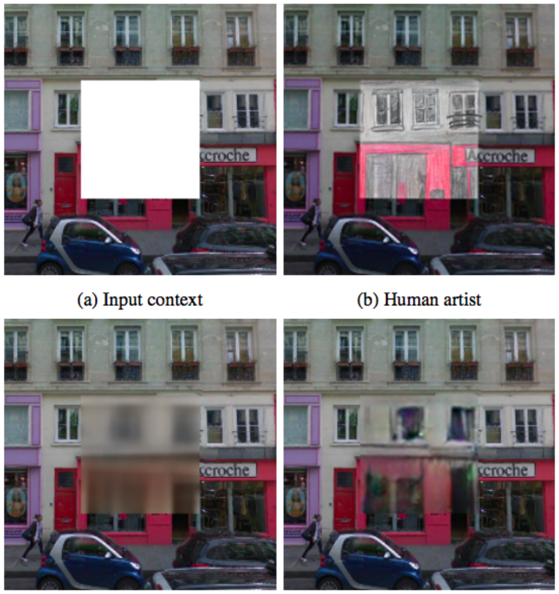
Trained to minimize L_2 image distortion between $y_\Phi(x)$ and y.

Evaluation

Self-supervised representations (features) are tested by training a linear classifier for ImageNet.

Self-Supervised Pretraining in Vision

Feature Learning by Inpainting, Pathak et al., 2016



(a) Input context

(b) Human artist

(c) Context Encoder
($L2$ loss)

(d) Context Encoder
($L2$ + Adversarial loss)

Trained to minimize distortion between $y_\Phi(x)$ and y and to maximize discriminator loss for distinguishing $y_\Phi(x)$ from y.

24

Contrastive Predictive Coding (CPC)

2 frame of video

We consider a population distribution on pairs $\langle x, y \rangle$.

For example x and y might be video frames separated by 10 seconds in a video. ⤳ find in common ⇒ Mutual Information

For simplicity we will assume that the marginal distributions on x and y are the same — the probability that an image occurs as a first frame is the same as the probability that image occurs as a second frame.

In CPC we draw a pair $\langle x, y \rangle$ and minimize a discriminator loss for distinguishing $z_\Phi(y)$ from $z_\Phi(\tilde{y})$ for $\tilde{y} \sim \mathrm{Pop}(y)$. The discriminator gets to see x.

discriminator needs to pick (predict) what happen in los

$$\min_\varphi (Z_\varphi(y) - Z_\varphi(\tilde{y}))$$

Contrastive Predictive Coding (CPC)

For $N \geq 2$ let $\tilde{P}^{(N)}$ be the distribution on tuples $\langle i, y_1, \ldots, y_N, x\rangle$ defined by the following process.

- draw a pair $\langle x, y\rangle$ from the population.
- drawn a sequence of $N-1$ "distractor values" from the marginal distribution $\mathrm{Pop}(y)$. These are unrelated to x.
- insert y at a random position among the distractors to get the sequence $y_1, \ldots, y_N$.
- return the tuple $\langle i, y_1, \ldots, y_N, x\rangle$ where i is the index of y among the distractors.

Contrastive Predictive Coding (CPC)

$$\Phi^* = \operatorname*{argmin}_{\Phi} \mathcal{L}_{\mathrm{CPC}}(\Phi)$$

$$\mathcal{L}_{\mathrm{CPC}}(\Phi) = E_{\langle i, y_1, \ldots, y_N, x\rangle \sim \tilde{P}^{(N)}} - \ln P_{\mathrm{CPC}}(i | z_\Phi(y_1), \ldots, z_\Phi(y_N), z_\Phi(x))$$

giving many videos
pick y_i corresponding
to x

$$P_{\mathrm{CPC}}(i | z_1, \ldots, z_N, z_x) = \operatorname*{softmax}_{i} \; z_i^\top z_x$$

trying to predict i

Contrastive Predictive Coding (CPC)

$$\Phi^* = \underset{\Phi}{\mathrm{argmin}} \; \mathcal{L}_{\mathrm{CPC}}(\Phi)$$

$$P_\Phi(i|z_1, \ldots, z_n, z_x) = \underset{i}{\mathrm{softmax}} \; z_i^\top z_x$$

As N gets larger the contrastive discrimination task gets harder.

The task is also made difficult by the requirement that the score is defined to be an inner product of feature vectors.

Contrastive Predictive Coding (CPC)

(SimCLR:) A Simple Framework for Contrastive Learning of Visual Representations, Chen et al., Feb. 2020 (self-supervised leader as of February, 2020).

They use a distribution on pairs $\langle x, y\rangle$ defined by drawing an image s from ImageNet and then drawing x and y as random "augmentations" (modifications) of the image s — either a random translation, rotation, color jitter, masking, edge image, or a composition of these modifications.

data augmentation: change the data a little bit

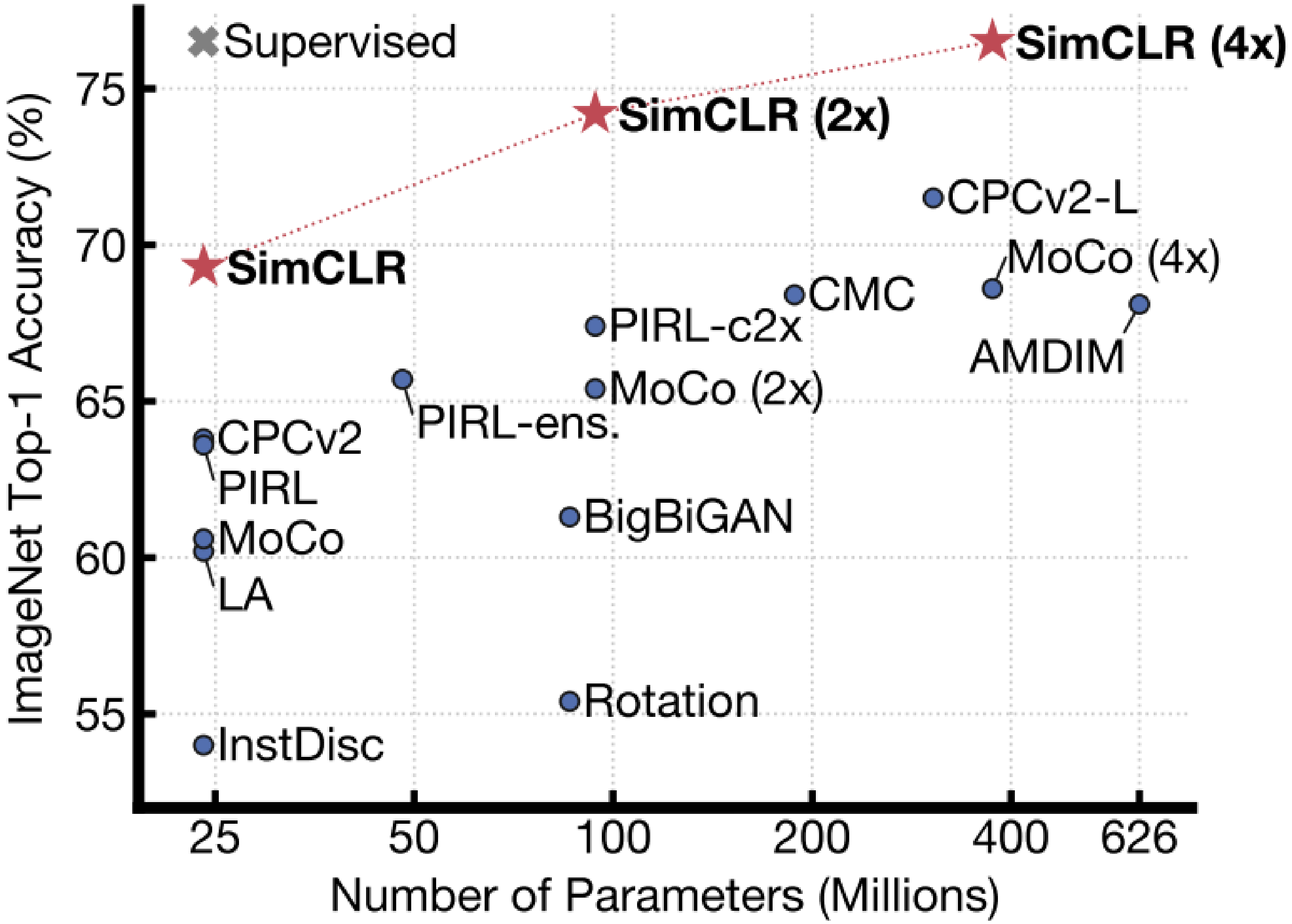
↳ add as new

Contrastive Predictive Coding (CPC)

The feature map z_Φ can then be applied to the images of ImageNet.

The feature map z_Φ is then tested by using a linear classifier for ImageNet based on these features.

SimCLR


Supervised
SimCLR (4x)
SimCLR (2x)
SimCLR
CPCv2-L
MoCo (4x)
CMC
PIRL-c2x
AMDIM
MoCo (2x)
PIRL-ens.
CPCv2
PIRL
BigBiGAN
MoCo
LA
Rotation
InstDisc
ImageNet Top-1 Accuracy (%)
75
70
65
60
55
25
50
100
200
400
626
Number of Parameters (Millions)

Mutual Information Objectives

CPC represents a fundamental shift in the self-supervised training objective.

GANs and VAEs are motivated by modeling $\mathrm{Pop}(y)$.

But in CPC there is no attempt to model $\mathrm{Pop}(y)$.

CPC can be viewed as training a feature map z_Φ so as to maximize the mutual information $I(z_\Phi(x), z_\Phi(y))$ while, at the same time, making $z_\Phi(x)$ useful for linear classifiers.

Relationship to Noise Contrastive Estimation

CPC is noise contrastive estimation (NCE) with "noise" generated by drawing y unrelated to x. By the NCE theorems, universality implies

$$P_{\Phi^*}(i|z_1, \ldots, z_N, z_x) = \operatorname*{softmax}_i \; \ln \frac{\mathrm{Pop}(z_i|z_x)}{\mathrm{Pop}(z_i)}$$

and also

$$\begin{aligned} \mathcal{L}_{\mathrm{CPC}} &\geq \ln N - \frac{N-1}{N}(KL(\mathrm{Pop}(z_y|z_x), \mathrm{Pop}(z_y)) + KL(\mathrm{Pop}(z_y), \mathrm{Pop}(z_y|z_x))) \\ &= \ln N - \frac{N-1}{N}(I(z_x, z_y) + KL(\mathrm{Pop}(z_y), \mathrm{Pop}(z_y|z_x))) \end{aligned}$$

Deep Co-Training

For a population on $\langle x, y\rangle$ and a "feature map" z_Φ we optimize Φ by

keep what we need remember to be low

$$\Phi^* = \operatorname*{argmax}_{\Phi}\; \underbrace{I(z_\Phi(\overset{\text{now}}{x}), z_\Phi(\overset{\text{future}}{y}))}_{\text{max}} - \beta \underbrace{H(z_\Phi(x))}_{\text{while low}}$$

Here we can think of $z_\Phi(x)$ as what we remember about a past x to carry information about a future y while maintaining low memory requirements.

want small

Deep Co-Training

$$\Phi^* = \underset{\Phi}{\mathrm{argmax}}\ (1-\beta)\hat{H}_\Phi(z_\Phi(x)) - \hat{H}_\Phi(z_\Phi(x)|z_\Phi(y))$$

$$\hat{H}_\Phi(z_\Phi(x)) = E_x\ P_{\Psi*}(z_\Phi(x))$$

min cross entropy

$$\Psi^* = \underset{\Psi}{\mathrm{argmin}}\ \ E_x\ \ -\ln P_\Psi(z_\Phi(x))$$

$$\hat{H}_\Phi(z_\Phi(x)|z_\Phi(y)) = E_{x,y}\ \ -\ln P_\Phi(z_\Phi(x)|z_\Phi(y))$$

Here, as in CPC, we only model distributions on z. There is no attempt to model distributions on x or y.

becomes Adversarial

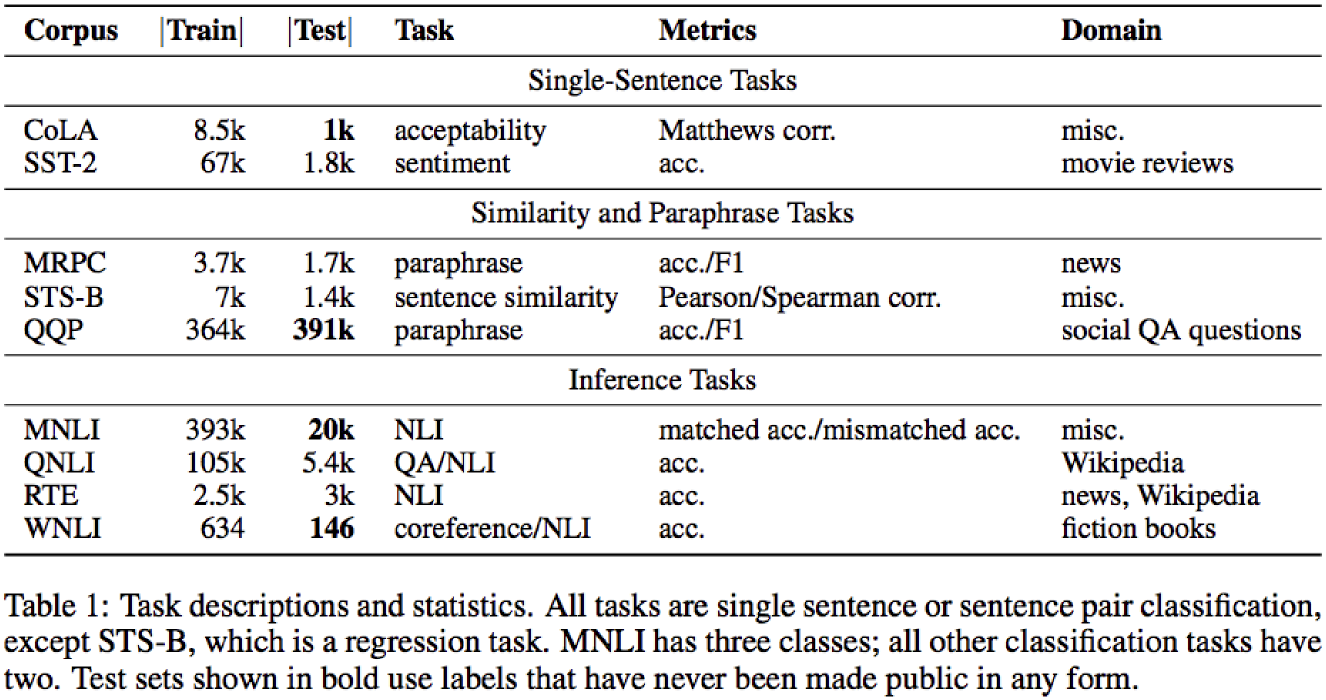
Pretraining for NLP

Unlike vision, in NLP self-supervised pretraining is now required for strong benchmark performance.

Moore’s Law of AI: Natural Language Understanding

GLUE: General Language Understanding Evaluation

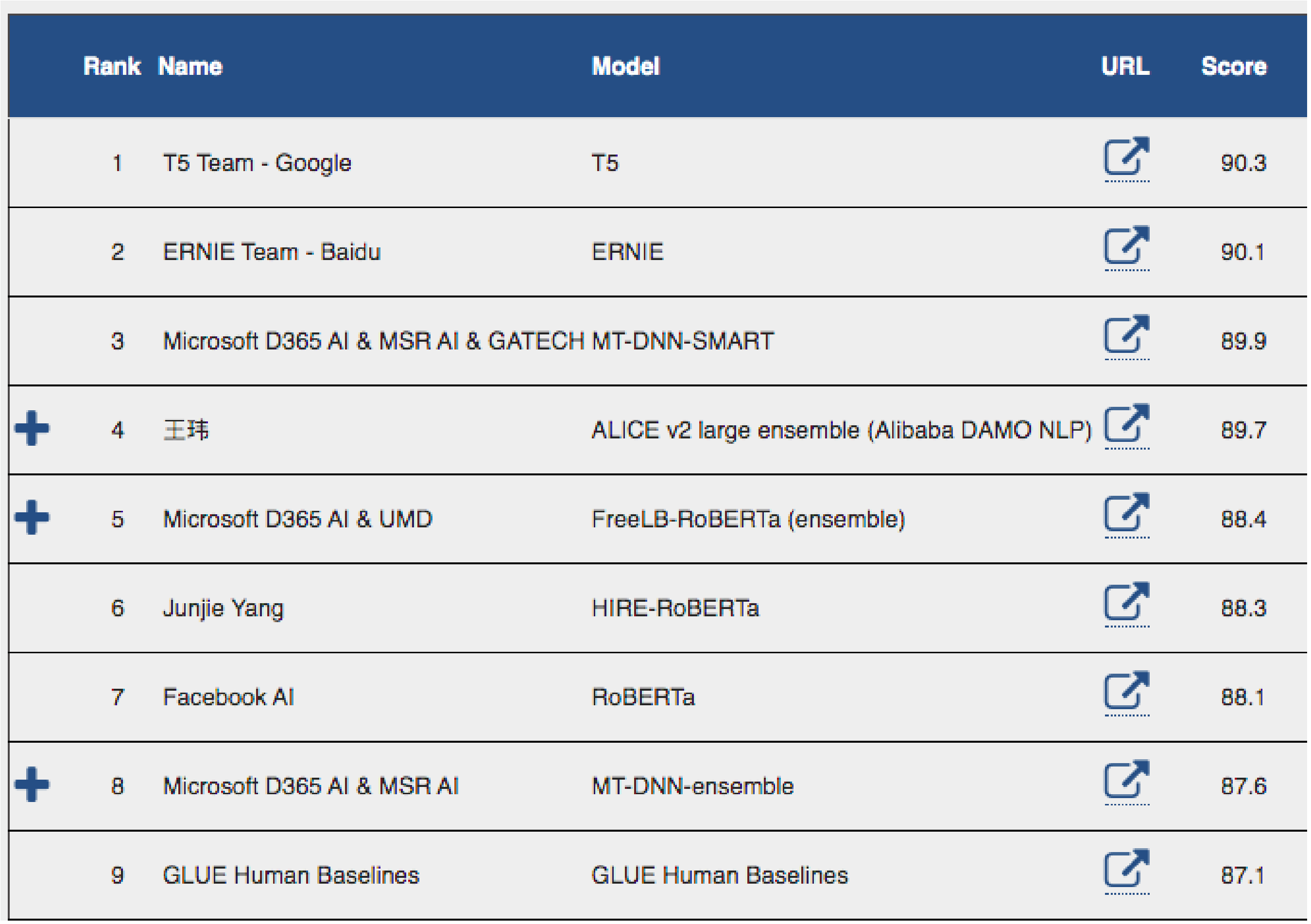
ArXiv 1804.07461



Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

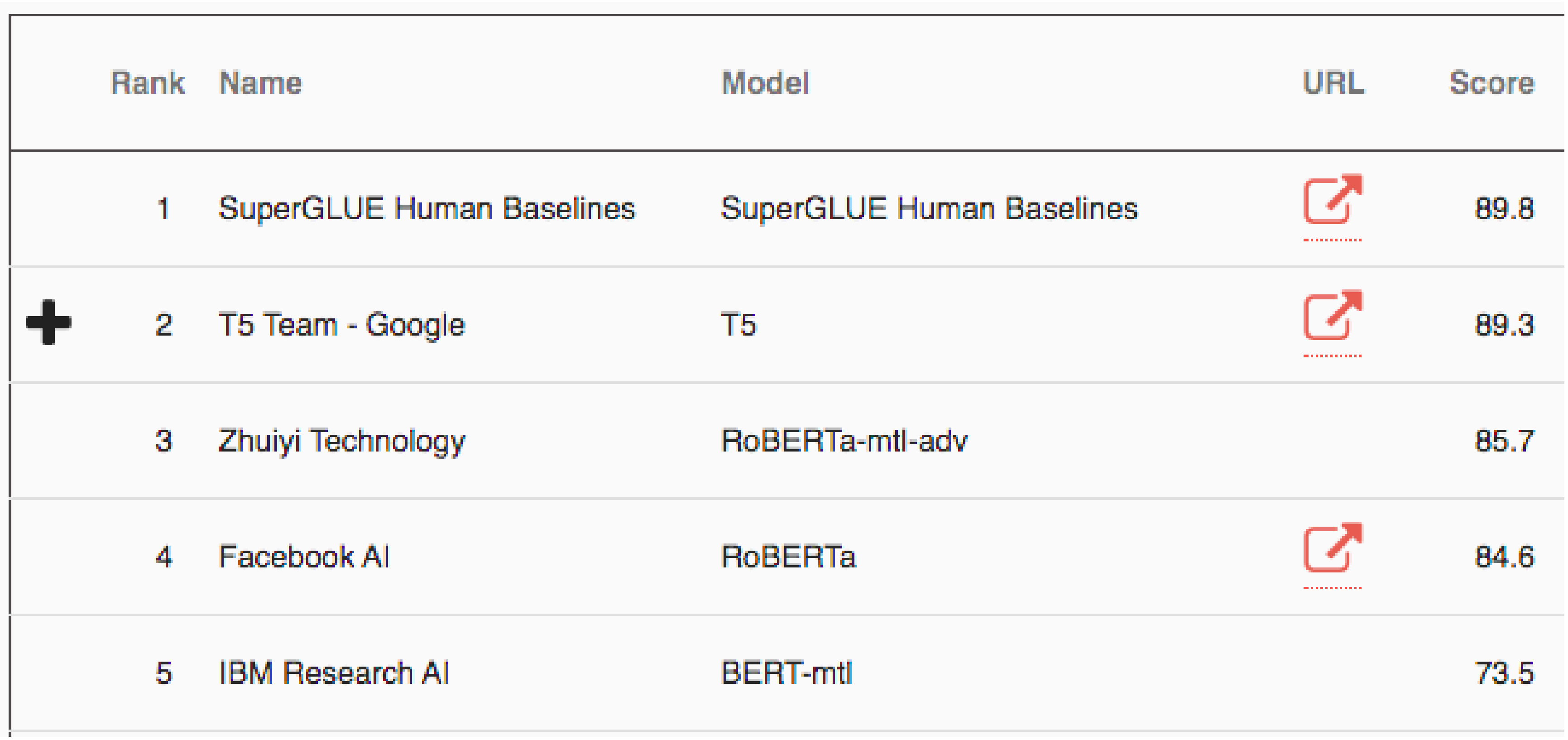
Table 1: Task descriptions and statistics. All tasks are single sentence or sentence pair classification, except STS-B, which is a regression task. MNLI has three classes; all other classification tasks have two. Test sets shown in bold use labels that have never been made public in any form.

GLUE Leader Board as of February 27, 2020



	Rank	Name	Model	URL	Score
	1	T5 Team - Google	T5	↗	90.3
	2	ERNIE Team - Baidu	ERNIE	↗	90.1
	3	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART	↗	89.9
✚	4	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)	↗	89.7
✚	5	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)	↗	88.4
	6	Junjie Yang	HIRE-RoBERTa	↗	88.3
	7	Facebook AI	RoBERTa	↗	88.1
✚	8	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	↗	87.6
	9	GLUE Human Baselines	GLUE Human Baselines	↗	87.1

SuperGLUE Leader Board as of February 27, 2020



	Rank	Name	Model	URL	Score
	1	SuperGLUE Human Baselines	SuperGLUE Human Baselines	↗	89.8
✚	2	T5 Team - Google	T5	↗	89.3
	3	Zhuiyi Technology	RoBERTa-mtl-adv		85.7
	4	Facebook AI	RoBERTa	↗	84.6
	5	IBM Research AI	BERT-mtl		73.5

Pretrained Word Embeddings

Advances in Pre-Training Distributed Word Representations, Mikolov et al., 2017

We want a mapping from a word w to a vector $e(w)$ — a word embedding.

fastText from Facebook is currently popular.

It provides both contextual bag of words (cbow) and byte pair encoding (BPE) word vectors.

cbow word vectors

We construct a population distribution on pairs (c, w) here c is a bag of word context and w is a word.

$$\Phi^* = \operatorname*{argmin}_{\Phi} E_{c,w} \; - \ln P(w|c)$$

Φ consists of a matrix $e[w, i]$ where $e[w, I]$ is the word embedding of w, and a matrix $e'[w, i]$ giving the embedding of the word w when it appears in a context.

A score $s(w|c)$ is defined by

$$s(w|c) = \frac{1}{|c|} \sum_{w' \in c} \underbrace{e(w)}_{\text{word embedding}}{}^{\top} \underbrace{e'(w')}_{\text{word in context embedding}}$$

Negative Sampling in cbow

Rather than define $P_\Phi(w|c)$ by a softmax over w, one uses restricted negative sampling.

We construct a training set of triples (w, c, N_C)

$$\Phi^* = \underset{\Phi}{\mathrm{argmin}}\ E_{w,c,N_c}\ \ln\left(1 + e^{-s(w,c)}\right) + \sum_{n \in N_C} \ln\ \left(1 + e^{s(n,c)}\right)$$

Byte Pair Encoding (BPE)

BPE constructs a set of character n-grams by starting with the unigrams and then greedily merging most common bigrams of n-grams.

Given a set of character n-grams each word is treated as a bag of character n-grams.

?

$$e[w] = \frac{1}{N} \sum_{n \in w} e(n)$$

Current systems use byte pairs but train the byte pair embeddings as part of transformer training.

The Transformer

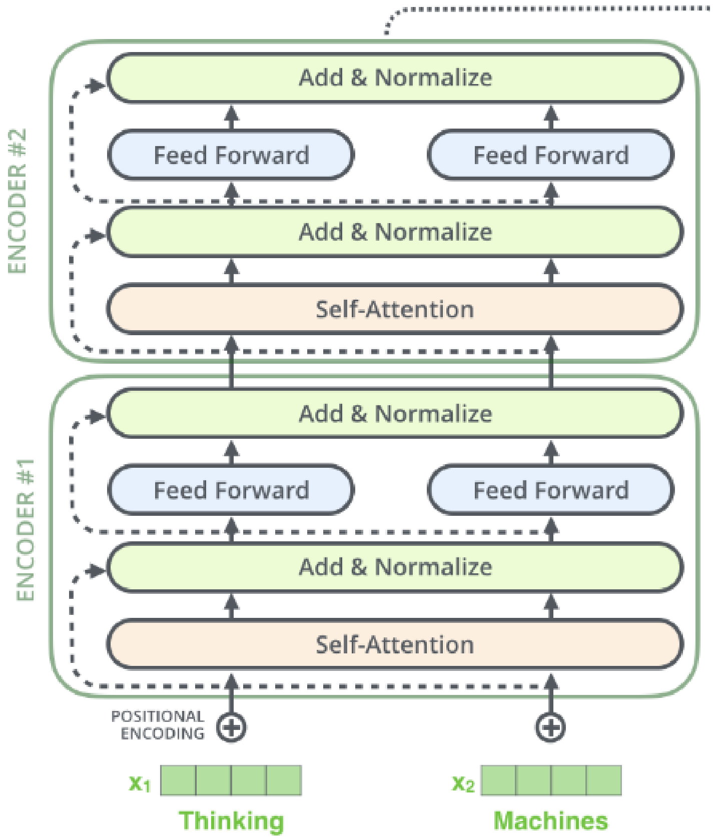
Attention is All You Need, Vaswani et al., June 2017

The transformer is like an RNN in that it takes a sequence of words and converts it to a sequence of vectors.

The output vectors are sometimes called contextual word embeddings.

But unlike RNNs, transformers run in parallel time in proportion to the layering depth independent of the length of the block.

The Transformer



Jay Alammar's blog

All layers run in $O(1)$ time independent of block length.

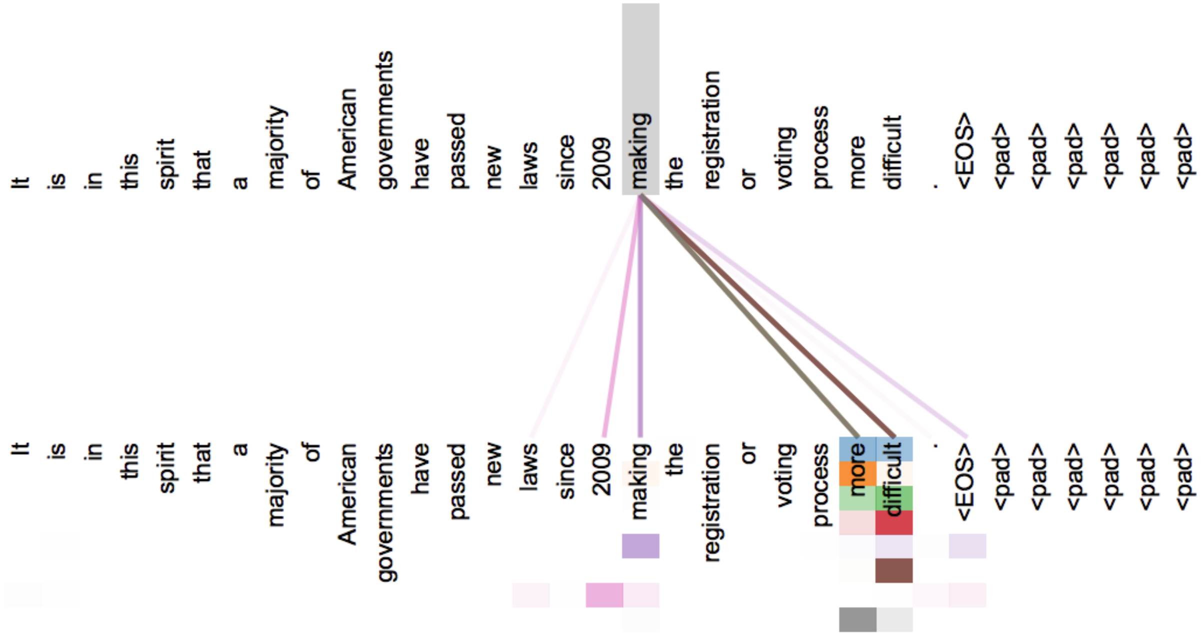
head: ask what is the subjective of the word?

attention for head k

A Self-Attention Layer

Given an $h_{\mathrm{in}}[T, J]$ we will construct $h_{\mathrm{out}}[T, J]$

We first construct a head-specific self-attention $\alpha[k, t_1, t_2]$ — the attention position t_1 is giving to position t_2 for head k.



naturally find the connection of words

Computing the Self Attention

For each head k and position t we compute a key vector and a query vector with dimension U typically smaller than dimension J.

head

i.e "512"

$$\mathrm{Query}[k,t,U] = W^Q[k,U,J]h_{\mathrm{in}}[t,J]$$

$$\mathrm{Key}[k,t,U] = W^K[k,U,J]h_{\mathrm{in}}[t,J]$$

$$\alpha[k,t_1,t_2] = \operatorname{softmax}_{t_2} \; \mathrm{Query}[k,t_1,U]\mathrm{Key}[k,t_2,U]$$

ask the attention from t_2 ⇒ after training we will get attention

Computing the Output

total dimension

head position

We require $I = J/K$.

attention

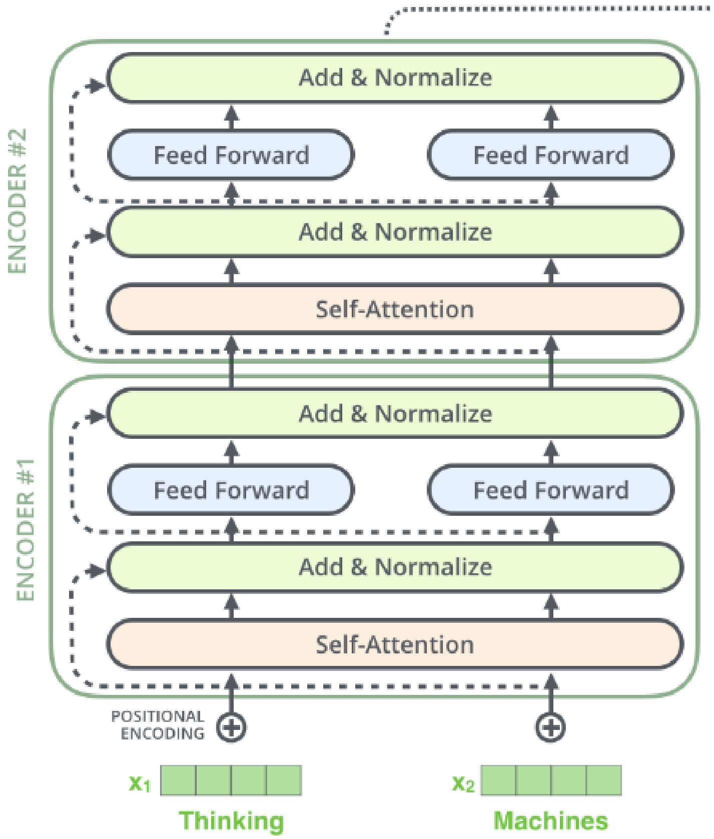
$$\mathrm{Value}[k,t,I] = W^V[k,I,J]h_{\mathrm{in}}[k,t,J]$$

$$\mathrm{Out}[k,t,I] = \sum_{t'} \alpha[k,t,t']\mathrm{Value}[k,t',I]$$

$$h_{\mathrm{out}}[t,J] = \mathrm{Out}[1,t,I];\cdots;\mathrm{Out}[K,t,I]$$

head position

The Transformer



Jay Alammar's blog

Position encodings are inserted at the bottom.

Encoding Positional Information

At the input layer we augment the word embeddings with position information. For example:

$$h[0,t,J] = e[w[t],I]; e^{i\omega t}\,; e^{i2\omega t}\,; e^{i4\omega t} \cdots ; e^{i2^k\omega t}$$

In modern versions there is a position encodings trained for each position in the block.

ELMO: Language Modeling

To do language modeling we fix $\alpha[k, t_1, t_2] = 0$ for $t_2 > t_1$.

We can then predict the word $w[t+1]$ as

top vector

$$P(w_{t+1}|w_1, \ldots, w_t) = \operatorname*{softmax}_{w}\; e[w, I]h_{\mathrm{top}}[t, I]$$

Machine Translation

Translation is just a conditional language model.

We take the input English sentence followed by a special token and then generate from the transformer language model.

Continuing from a Prompt

GPT-2 from Open AI.

Continue from:

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.

The Predicted Continuation

Continuation excerpted from a single response, the response selected from 10 tries.

The scientist named the population, after their distinctive horn, Ovids 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 Prez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when ... Prez and his friends were astonished to see the unicorn herd. ... While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Prez stated, We can see, for example,

that they have a common language, something like a dialect or dialectic.

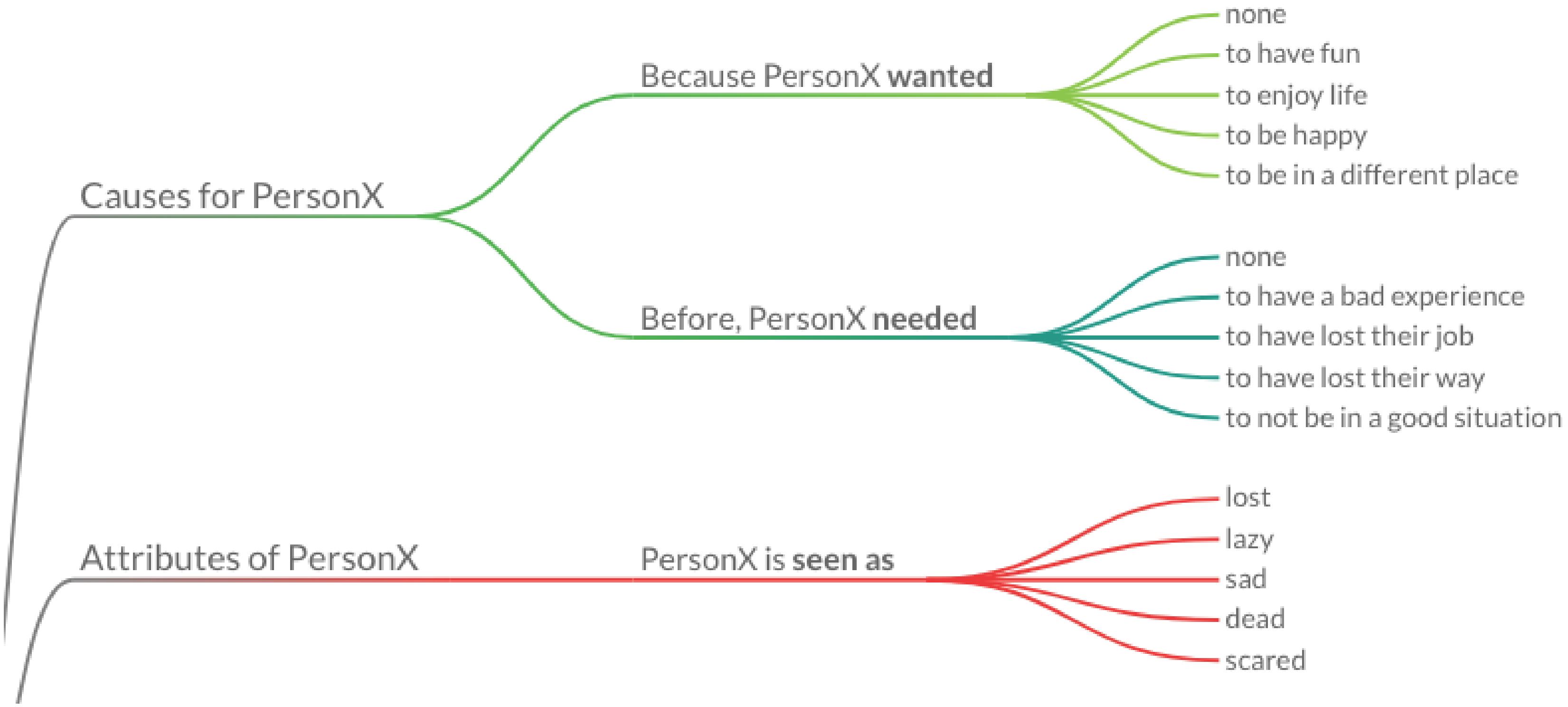
Dr. Prez believes that the unicorns may have originated in Argentina ... some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. ... However, Prez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA.

...

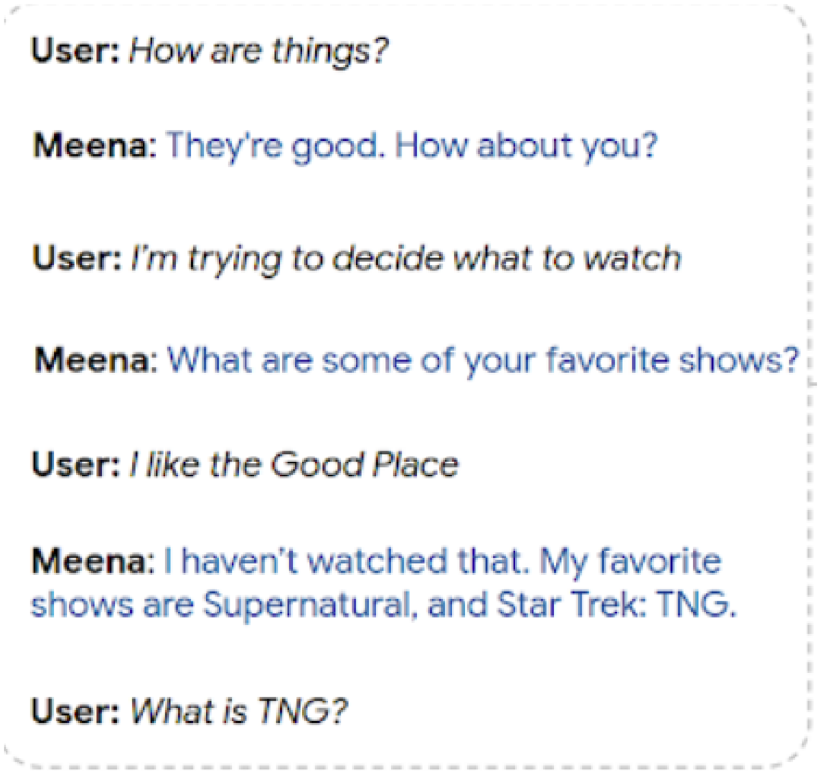
Fine Tuning on Question Answering

COMET: Busselut et al, June 2019.

Charlie is drifting though life:



The Chatbot Meena



User: *How are things?*

Meena: They're good. How about you?

User: *I'm trying to decide what to watch*

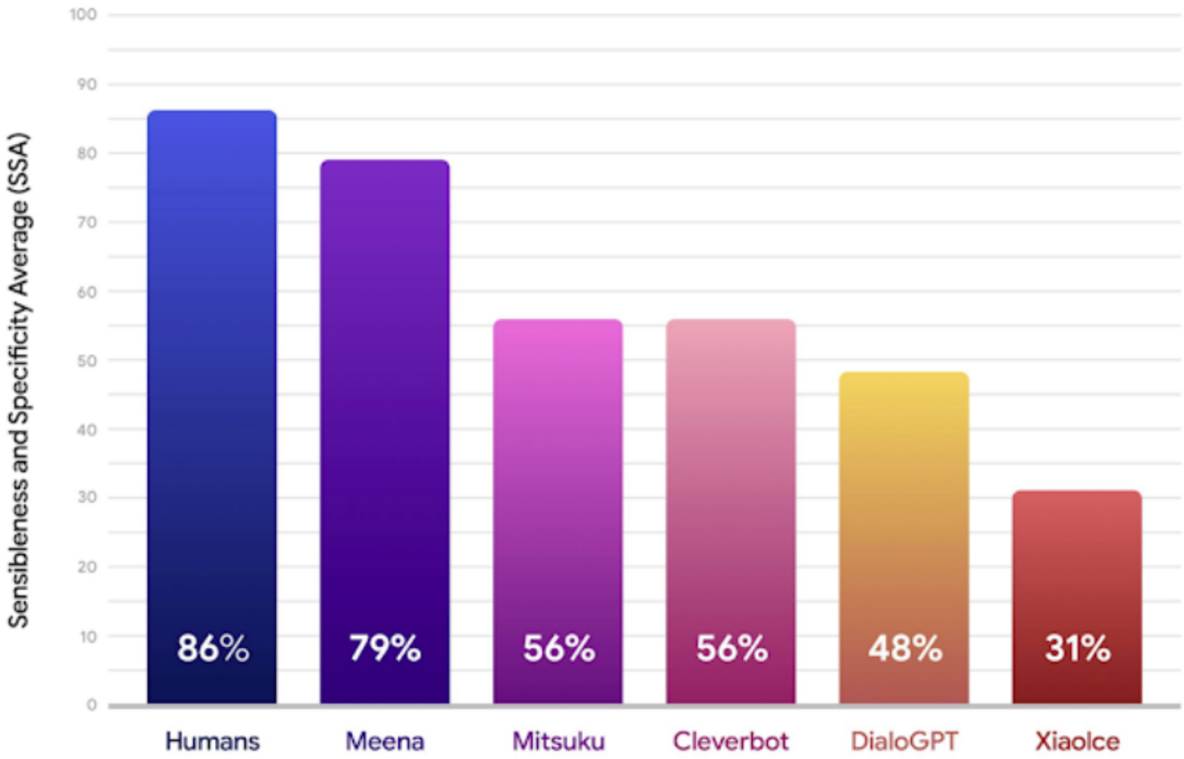
Meena: What are some of your favorite shows?

User: *I like the Good Place*

Meena: I haven't watched that. My favorite shows are Supernatural, and Star Trek: TNG.

User: *What is TNG?*

The Chatbot Meena



BERT: Blank Languagage Modeling

We replace a random subset of the words with a blank token.

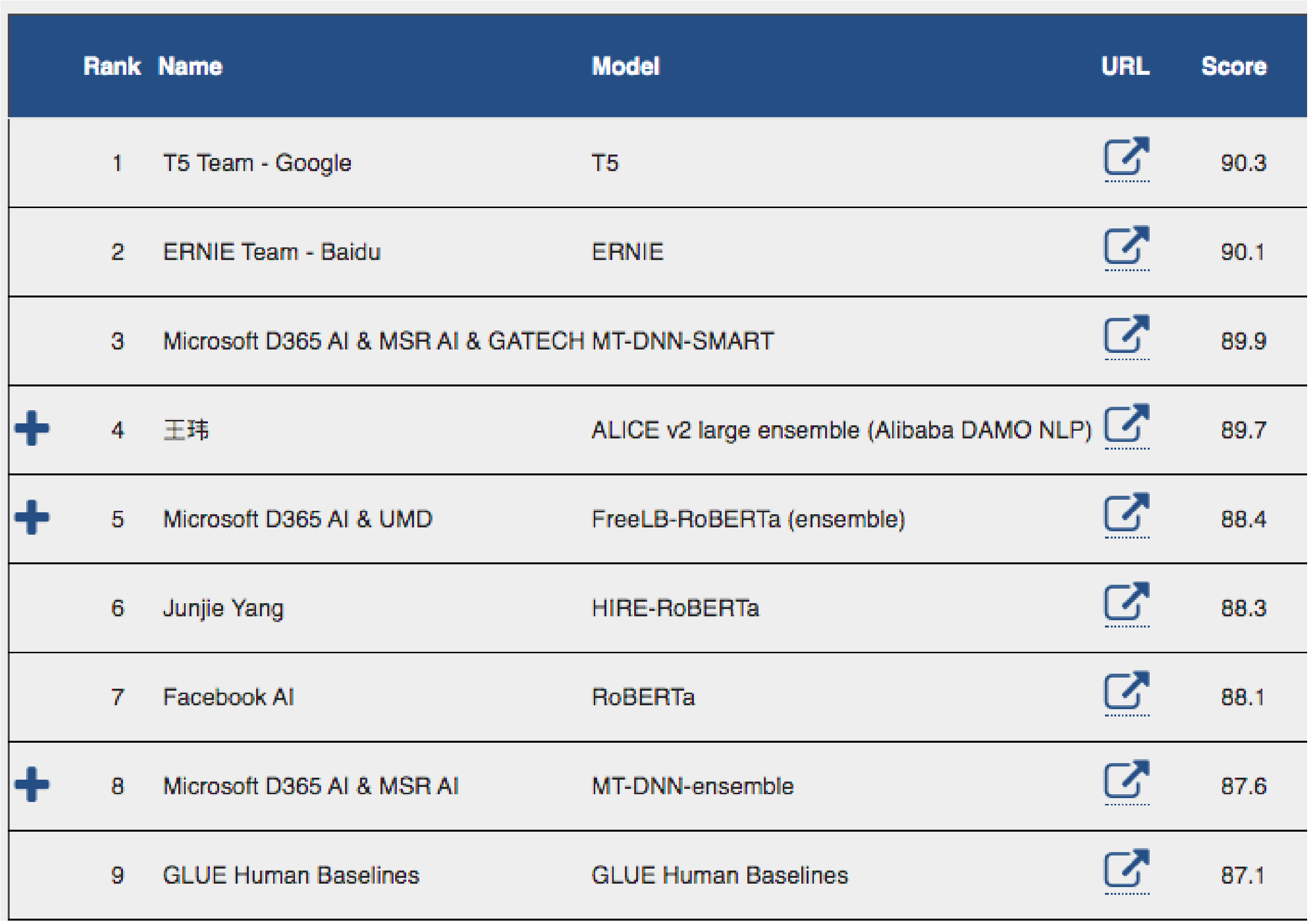
We run a transformer on a block of text containing some blanks.

For a blank occurring at position t we predict the word at position t:

$$P(w) = \operatorname*{softmax}_w \; h[t,J]e[w,J]$$

Blank language modeling outperforms language modeling when used for pretraining in classification tasks such as the GLUE tasks.

GLUE Leader Board as of February 27, 2020



	Rank	Name	Model	URL	Score
	1	T5 Team - Google	T5	↗	90.3
	2	ERNIE Team - Baidu	ERNIE	↗	90.1
	3	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART	↗	89.9
+	4	王玮	ALICE v2 large ensemble (Alibaba DAMO NLP)	↗	89.7
+	5	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)	↗	88.4
	6	Junjie Yang	HIRE-RoBERTa	↗	88.3
	7	Facebook AI	RoBERTa	↗	88.1
+	8	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	↗	87.6
	9	GLUE Human Baselines	GLUE Human Baselines	↗	87.1

END