DSC291: Advanced Statistical Natural Language Processing

Self-supervised Learning

Zhiting Hu

Lecture 5, April 12, 2022

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Outline

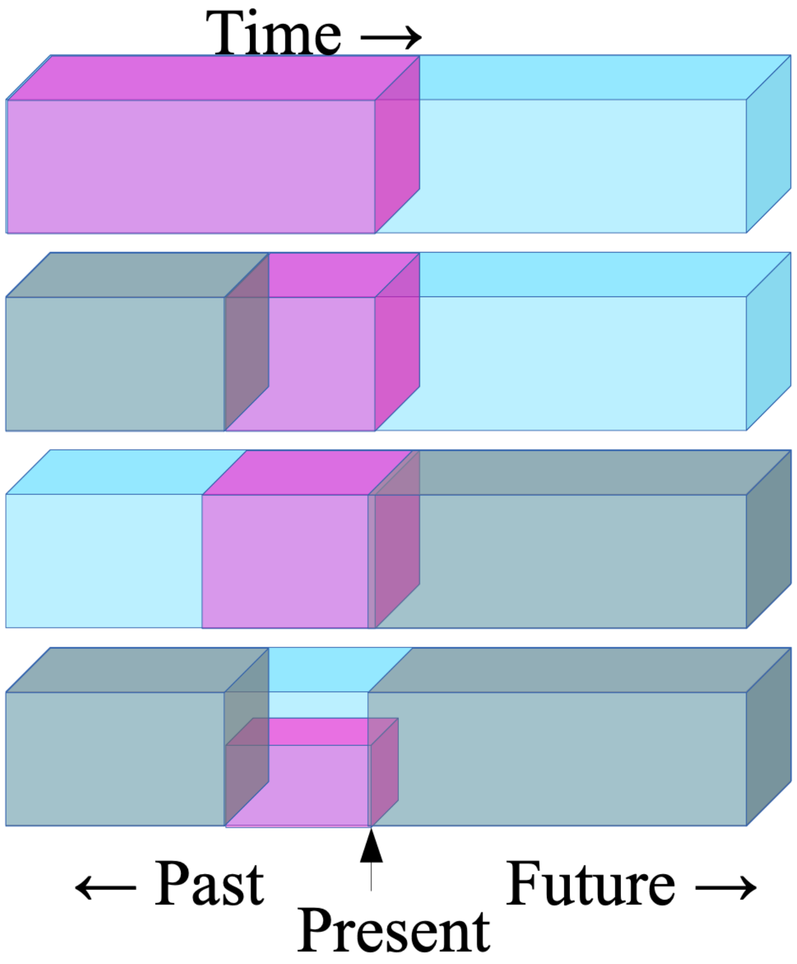
- Self supervised learning
- Contrastive learning (a special self-supervised learning)

Self-Supervised Learning

- Given an observed data instance $\boldsymbol{t}$
- One could derive various supervision signals based on the structure of the data
- By applying a "split" function that artificially partition $\boldsymbol{t}$ into two parts
  - $(\boldsymbol{x}, \boldsymbol{y}) = split(\boldsymbol{t})$
  - sometimes split in a stochastic way
- Treat $\boldsymbol{x}$ as the input and $\boldsymbol{y}$ as the output
- Train a model $p_\theta(\boldsymbol{y}|\boldsymbol{x})$

Self-Supervised Learning: Examples

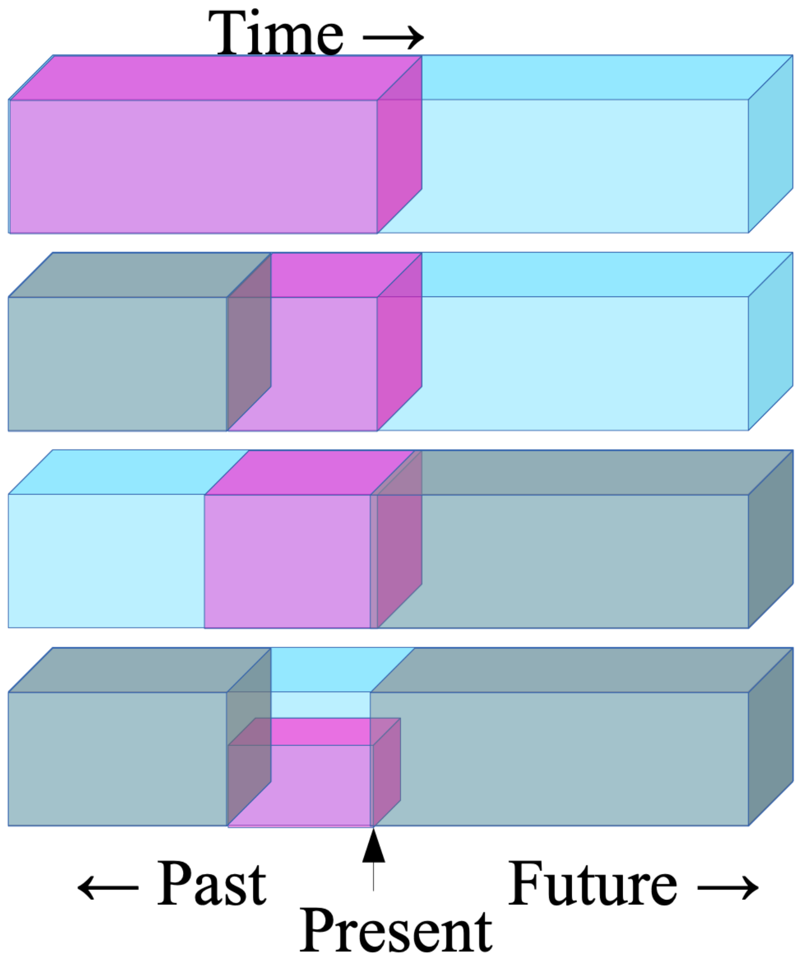
- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the future from the recent past.**
- **Predict the past from the present.**
- **Predict the top from the bottom.**



[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Examples

- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the future from the recent past.**
- **Predict the past from the present.**
- **Predict the top from the bottom.**
- **Predict the occluded from the visible**
- **Pretend there is a part of the input you don't know and predict that.**



[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Motivation (I)

- **Our brains do this all the time**

- **Filling in the visual field at the retinal blind spot**
- **Filling in occluded images, missing segments in speech**
- **Predicting the state of the world from partial (textual) descriptions**
- **Predicting the consequences of our actions**
- **Predicting the sequence of actions leading to a result**

- **Predicting any part of the past, present or future percepts from whatever information is available.**

[Courtesy: Lecun "Self-supervised Learning"]

Self-Supervised Learning: Motivation (I)

- Successfully learning to predict everything from everything else would result in the accumulation of lots of background knowledge about how the world works

- The model is forced to learn what we really care about, e.g. a semantic representation, in order to solve the prediction problem

[Courtesy: Lecun “Self-supervised Learning”]

[Courtesy: Zisserman “Self-supervised Learning”]

Self-Supervised Learning: Motivation (II)

- The machine predicts any part of its input from any observed part
  - A lot of supervision signals in each data instance

- Untapped/availability of vast numbers of unlabeled text/images/videos..
  - Facebook: one billion images uploaded per day
  - 300 hours of video are uploaded to YouTube every minute

[Courtesy: Zisserman "Self-supervised Learning"]

Self-Supervised Learning from Text

Examples:

- Language models
- Learning text representations

Language Models: Training

- Given data example $\boldsymbol{y}^*$
- Minimizes negative log-likelihood of the data

$$\min_\theta \mathcal{L}_{\text{MLE}} = -\log p_\theta(\boldsymbol{y}^*) = -\prod_{t=1}^{T} p_\theta(y_t^* \mid \boldsymbol{y}_{1:t-1}^*)$$

  - Next word prediction

Self-Supervised Learning from Text

Examples:

- Language models
- Learning text representations

Word Embedding

- A pre-trained **matrix**, each row is an embedding vector of a word

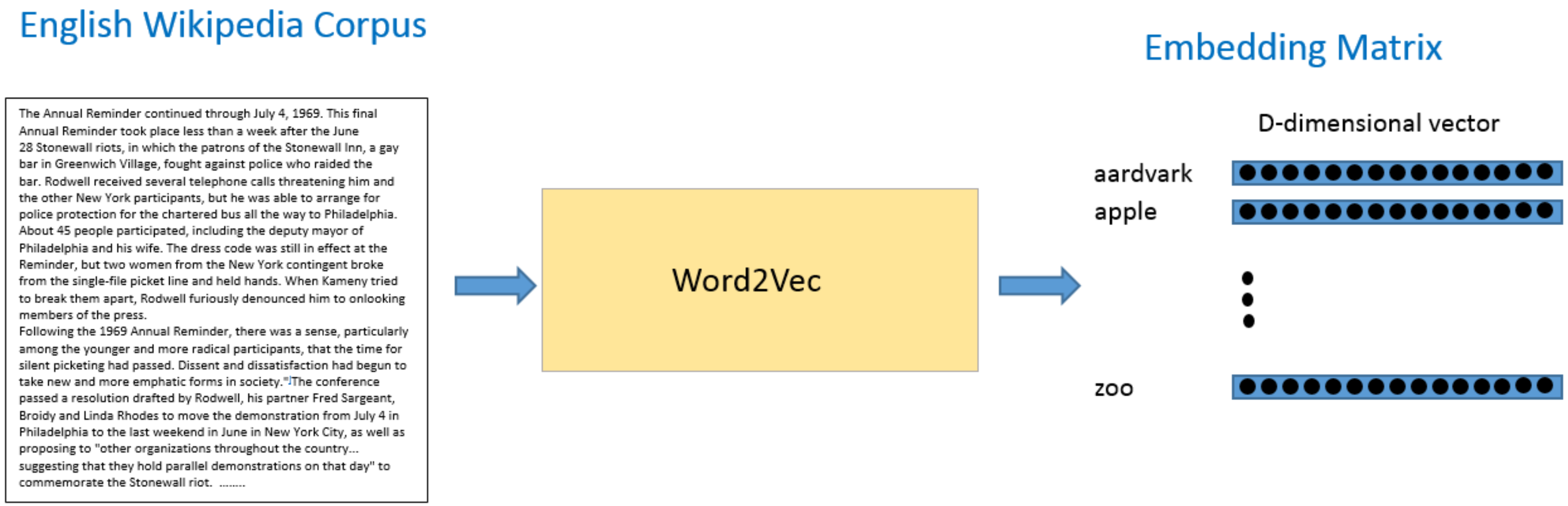
	0	1	2	3	4	5	6	7	8	9	...
fox	-0.348680	-0.077720	0.177750	-0.094953	-0.452890	0.237790	0.209440	0.037886	0.035064	0.899010	...
ham	-0.773320	-0.282540	0.580760	0.841480	0.258540	0.585210	-0.021890	-0.463680	0.139070	0.658720	...
brown	-0.374120	-0.076264	0.109260	0.186620	0.029943	0.182700	-0.631980	0.133060	-0.128980	0.603430	...
beautiful	0.171200	0.534390	-0.348540	-0.097234	0.101800	-0.170860	0.295650	-0.041816	-0.516550	2.117200	...
jumps	-0.334840	0.215990	-0.350440	-0.260020	0.411070	0.154010	-0.386110	0.206380	0.386700	1.460500	...
eggs	-0.417810	-0.035192	-0.126150	-0.215930	-0.669740	0.513250	-0.797090	-0.068611	0.634660	1.256300	...
beans	-0.423290	-0.264500	0.200870	0.082187	0.066944	1.027600	-0.989140	-0.259950	0.145960	0.766450	...
sky	0.312550	-0.303080	0.019587	-0.354940	0.100180	-0.141530	-0.514270	0.886110	-0.530540	1.556600	...
bacon	-0.430730	-0.016025	0.484620	0.101390	-0.299200	0.761820	-0.353130	-0.325290	0.156730	0.873210	...
breakfast	0.073378	0.227670	0.208420	-0.456790	-0.078219	0.601960	-0.024494	-0.467980	0.054627	2.283700	...
toast	0.130740	-0.193730	0.253270	0.090102	-0.272580	-0.030571	0.096945	-0.115060	0.484000	0.848380	...
today	-0.156570	0.594890	-0.031445	-0.077586	0.278630	-0.509210	-0.066350	-0.081890	-0.047986	2.803600	...
blue	0.129450	0.036518	0.032298	-0.060034	0.399840	-0.103020	-0.507880	0.076630	-0.422920	0.815730	...
green	-0.072368	0.233200	0.137260	-0.156630	0.248440	0.349870	-0.241700	-0.091426	-0.530150	1.341300	...
kings	0.259230	-0.854690	0.360010	-0.642000	0.568530	-0.321420	0.173250	0.133030	-0.089720	1.528600	...
dog	-0.057120	0.052685	0.003026	-0.048517	0.007043	0.041856	-0.024704	-0.039783	0.009614	0.308416	...
sausages	-0.174290	-0.064869	-0.046976	0.287420	-0.128150	0.647630	0.056315	-0.240440	-0.025094	0.502220	...
lazy	-0.353320	-0.299710	-0.176230	-0.321940	-0.385640	0.586110	0.411160	-0.418680	0.073093	1.486500	...
love	0.139490	0.534530	-0.252470	-0.125650	0.048748	0.152440	0.199060	-0.065970	0.128830	2.055900	...
quick	-0.445630	0.191510	-0.249210	0.465900	0.161950	0.212780	-0.046480	0.021170	0.417660	1.686900	...

[Courtesy: Vaswani, et al., 2017]

Word Embedding

- A pre-trained **matrix**, each row is an embedding vector of a word

	0	1	2	3	4	5	6	7	8	9	...
fox	-0.348680	-0.077720	0.177750	-0.094953	-0.452890	0.237790	0.209440	0.037886	0.035064	0.899010	...
ham	-0.773320	-0.282540	0.580760	0.841480	0.258540	0.585210	-0.021890	-0.463680	0.139070	0.658720	...
brown	-0.374120	-0.076264	0.109260	0.186620	0.029943	0.182700	-0.631980	0.133060	-0.128980	0.603430	...
beautiful	0.171200	0.534390	-0.348540	-0.097234	0.101800	-0.170860	0.295650	-0.041816	-0.516550	2.117200	...
jumps	-0.334840	0.215990	-0.350440	-0.260020	0.411070	0.154010	-0.386110	0.206380	0.386700	1.460500	...
eggs	-0.417810	-0.035192	-0.126150	-0.215930	-0.669740	0.513250	-0.797090	-0.068611	0.634660	1.256300	...
beans	-0.423290	-0.264500	0.200870	0.082187	0.066944	1.027600	-0.989140	-0.259950	0.145960	0.766450	...
sky	0.312550	-0.303080	0.019587	-0.354940	0.100180	-0.141530	-0.514270	0.886110	-0.530540	1.556600	...
bacon	-0.430730	-0.016025	0.484620	0.101390	-0.299200	0.761820	-0.353130	-0.325290	0.156730	0.873210	...
breakfast	0.073378	0.227670	0.208420	-0.456790	-0.078219	0.601960	-0.024494	-0.467980	0.054627	2.283700	...
							945	-0.115060	0.484000	0.848380	...
							350	-0.081890	-0.047986	2.803600	...
							880	0.076630	-0.422920	0.815730	...
							700	-0.091426	-0.530150	1.341300	...
							250	0.133030	-0.089720	1.528600	...
							704	-0.039783	0.009614	0.308416	...
							315	-0.240440	-0.025094	0.502220	...
							160	-0.418680	0.073093	1.486500	...
							060	-0.065970	0.128830	2.055900	...
							480	0.021170	0.417660	1.686900	...



The Annual Reminder continued through July 4, 1969. This final Annual Reminder took place less than a week after the June 28 Stonewall riots, in which the patrons of the Stonewall Inn, a gay bar in Greenwich Village, fought against police who raided the bar. Rodwell received several telephone calls threatening him and the other New York participants, but he was able to arrange for police protection for the chartered bus all the way to Philadelphia. About 45 people participated, including the deputy mayor of Philadelphia and his wife. The dress code was still in effect at the Reminder, but two women from the New York contingent broke from the single-file picket line and held hands. When Kameny tried to break them apart, Rodwell furiously denounced him to onlooking members of the press.
Following the 1969 Annual Reminder, there was a sense, particularly among the younger and more radical participants, that the time for silent picketing had passed. Dissent and dissatisfaction had begun to take new and more emphatic forms in society.[1] The conference passed a resolution drafted by Rodwell, his partner Fred Sargeant, Broidy and Linda Rhodes to move the demonstration from July 4 in Philadelphia to the last weekend in June in New York City, as well as proposing to "other organizations throughout the country... suggesting that they hold parallel demonstrations on that day" to commemorate the Stonewall riot. ....

[Image source: Va

Word2vec: Skip-Gram Model

- (Mikolov et al., 2013a,b)

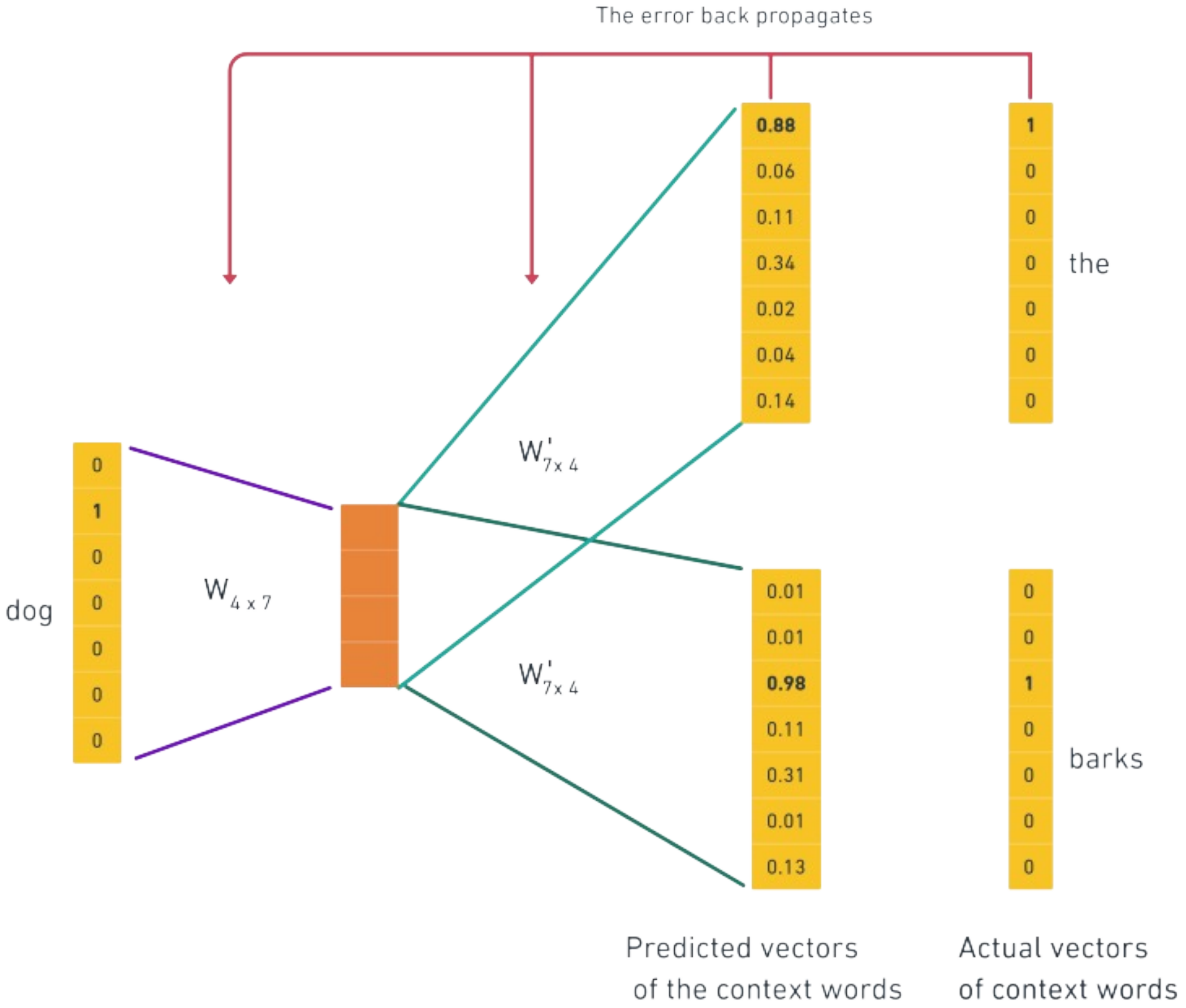
$$p(C = c \mid X = v) = \frac{1}{Z_v} \exp \mathbf{c}_c^\top \mathbf{v}_v$$

  - Two different vectors for each element of $\mathcal{V}$: one when it is "v" ($\mathbf{v}$) and one when it is "c" ($\mathbf{c}$).
  - This should remind you of a neural network; SGD on the likelihood function is the conventional approach to estimating the vectors.
  - Normalization term Z_v is expensive, so approximations are required for efficiency.
  - Can expand this to be over the whole sentence or document, or otherwise choose which words "count" as context.

[Courtesy: UW CSE 447 by Noah Smith]

Word2vec: Skip-Gram Model

“the dog barks”



[Figure courtesy: Maryam Fallah]

Word Embedding Evaluation

Several popular methods for *intrinsic* evaluations:

- Do (cosine) similarities of pairs of words' vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., *rug* $\stackrel{?}{\rightarrow}$ {*sofa*, *ottoman*, *carpet*, *hallway*}
- Syntactic analogies, e.g., "*walking* is to *walked* as *eating* is to what?" Solved via:

$$\max_{v \in \mathcal{V}} \cos\left(\mathbf{v}_v, -\mathbf{v}_{\textit{walking}} + \mathbf{v}_{\textit{walked}} + \mathbf{v}_{\textit{eating}}\right)$$

[Courtesy: UW CSE 447 by Noah Smith]

Word Embedding Evaluation

Extrinsic evaluation:

1. Use large unannotated corpus to get your word vectors (sometimes called **pretraining**).
2. Use them in a text classifier (or some other NLP system). Two options:
   - Plug in word vectors as "frozen" features, and estimate the other parameters of your model.
   - Treat them as parameters of the text classifier; pretraining gives initial values, but they get updated, or "finetuned" during supervised learning.
3. Does that system's performance improve?

[Courtesy: UW CSE 447 by Noah Smith]

Word Embedding

- **Problem**: word embeddings are applied in a context free manner

```
open a bank account          on the river bank

                  [0.3, 0.2, -0.8, …]
```

Word Embedding

- **Problem**: word embeddings are applied in a context free manner

`open a bank account` `on the river bank`

`[0.3, 0.2, -0.8, …]`

- **Solution**: Train contextual representations on text corpus

`[0.9, -0.2, 1.6, …]` `[-1.9, -0.4, 0.1, …]`

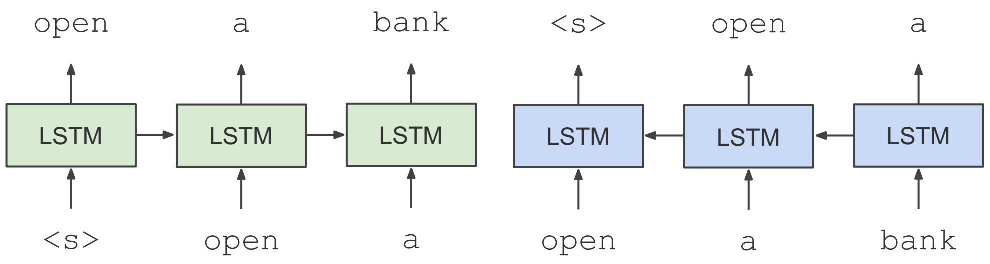
`open a bank account` `on the river bank`

Courtesy: Devlin 2019

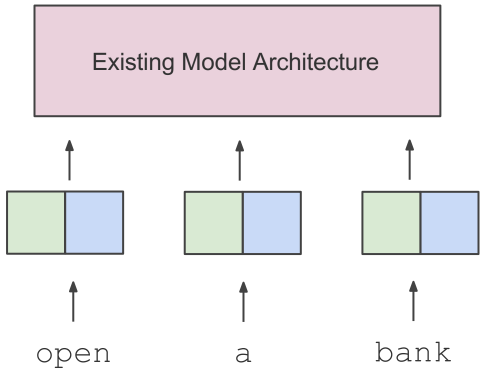
Contextual Representations

- *ELMo: Deep Contextual Word Embeddings*, AI2 & University of Washington, 2017

Train Separate Left-to-Right and Right-to-Left LMs



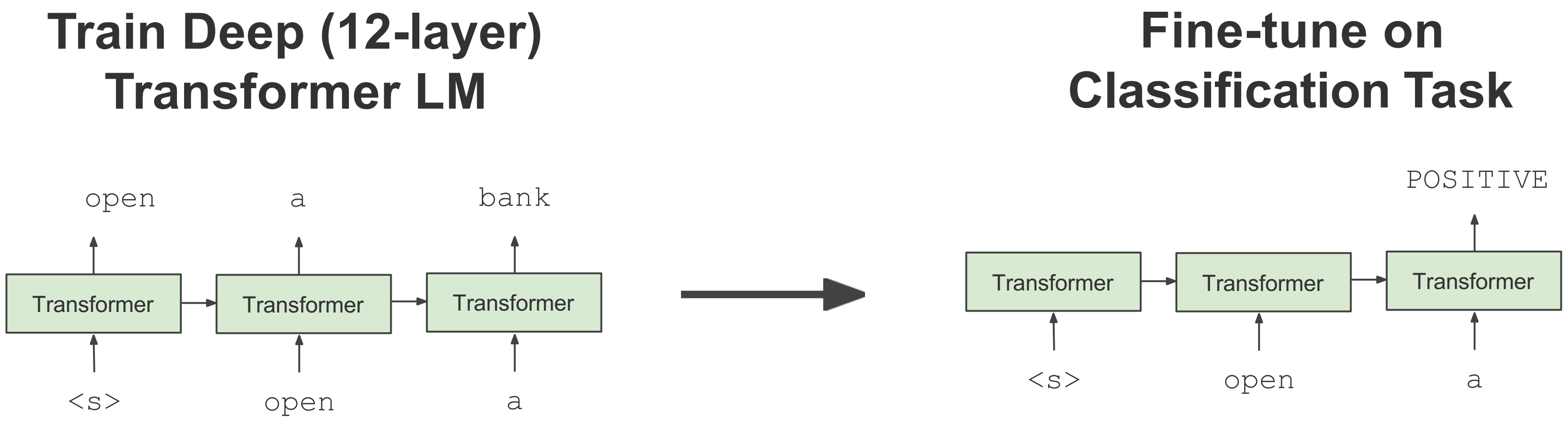
Apply as “Pre-trained Embeddings”



Courtesy: Devlin 2019

Contextual Representations

- *Improving Language Understanding by Generative Pre-Training*, OpenAI, 2018



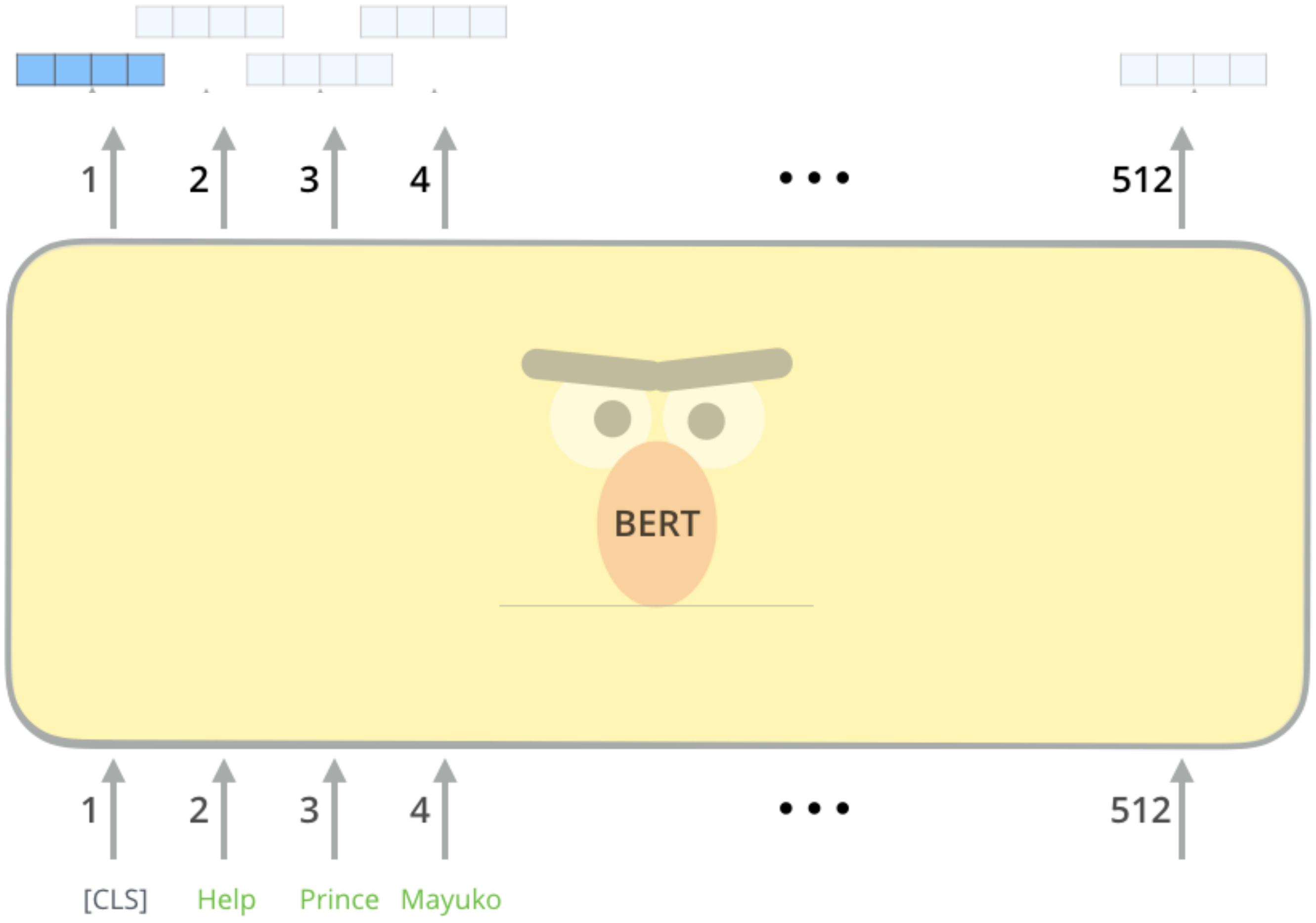
Courtesy: Devlin 2019

Problem with Previous Methods

- **Problem**: Language models only use left context *or* right context, but language understanding is bidirectional.

BERT

- BERT: A bidirectional model to extract contextual word embedding



BERT: Pre-training Procedure

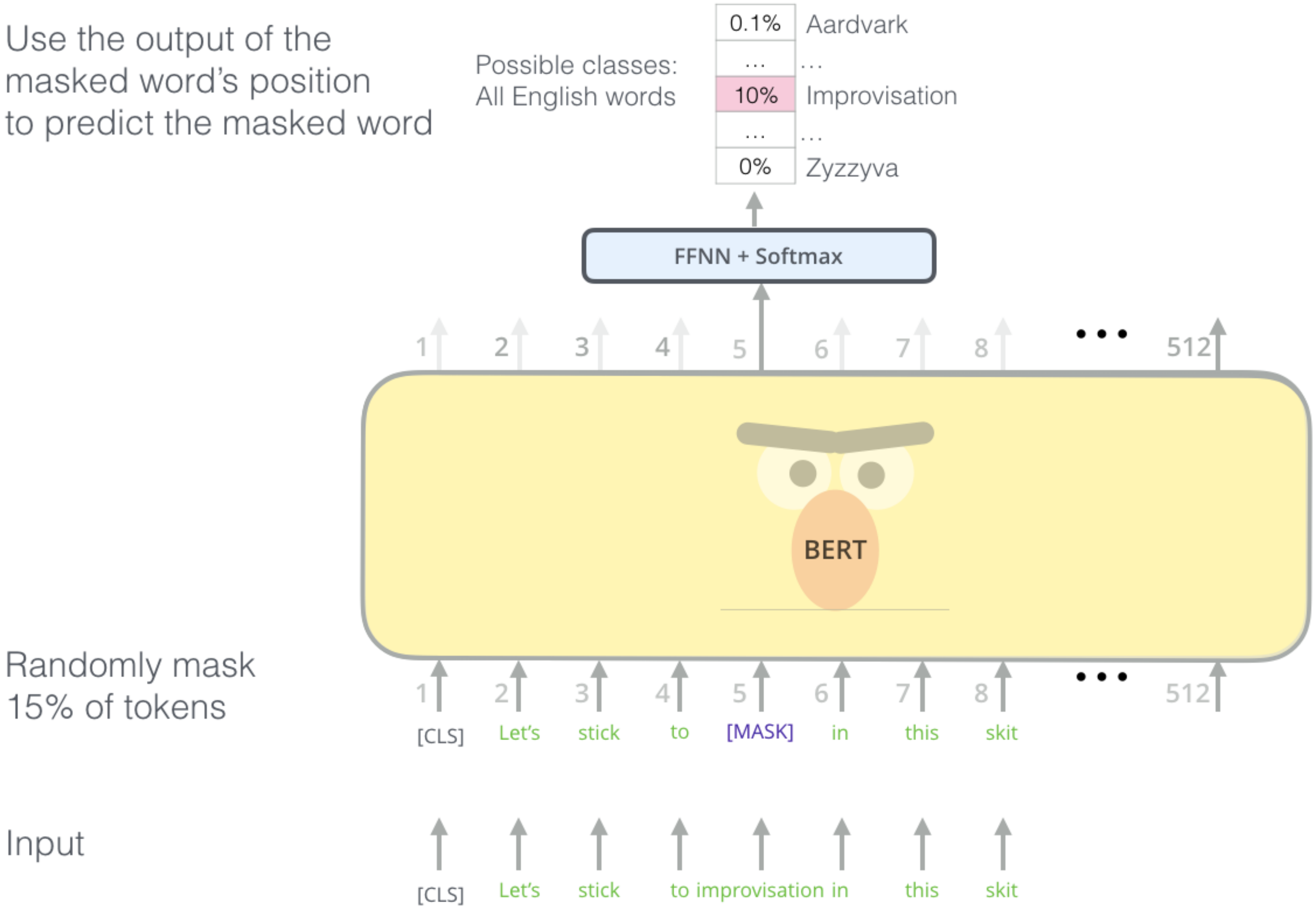
- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)

BERT: Pre-training Procedure

- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)
- Training procedure
  - **masked language model** (masked LM)
    - Masks some percent of words from the input and has to reconstruct those words from context

BERT: Pre-training Procedure

- Masked LM



BERT: Pre-training Procedure

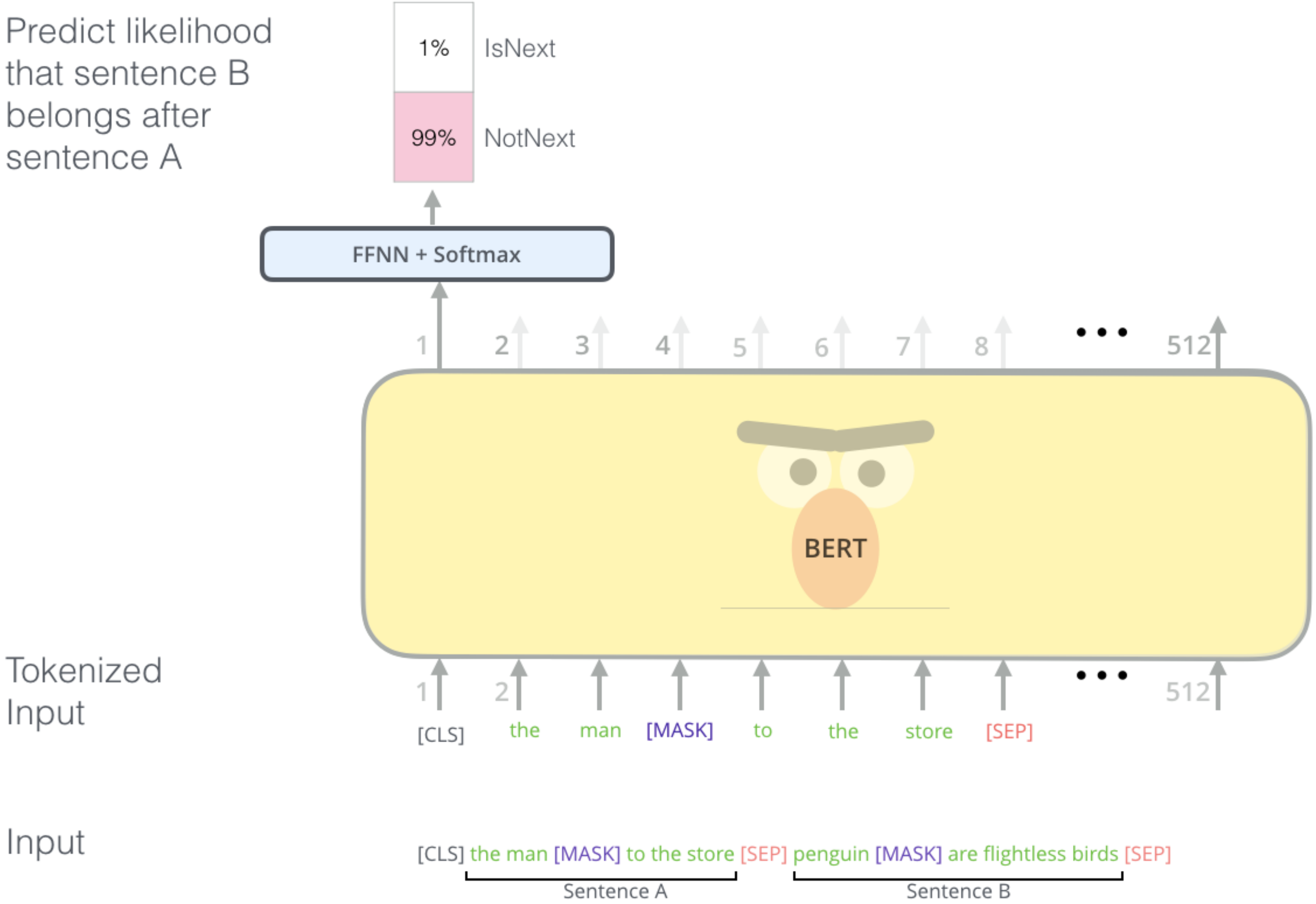
- Masked LM
- 15% masking:
    - Too little masking: Too expensive to train (few supervision signals per example)
    - Too much masking: Not enough context
- Problem: Mask token never seen at fine-tuning
- Solution: don't replace with `[MASK]` 100% of the time. Instead:
- 80% of the time, replace with `[MASK]`
    - `went to the store → went to the [MASK]`
- 10% of the time, replace random word
    - `went to the store → went to the running`
- 10% of the time, keep same
    - `went to the store → went to the store`

BERT: Pre-training Procedure

- Dataset:
  - Wikipedia (2.5B words) + a collection of free ebooks (800M words)
- Training procedure
  - **masked language model** (masked LM)
    - Masks some percent of words from the input and has to reconstruct those words from context
  - **Two-sentence task**
    - To understand relationships between sentences
    - Concatenate two sentences A and B and predict whether B actually comes after A in the original text

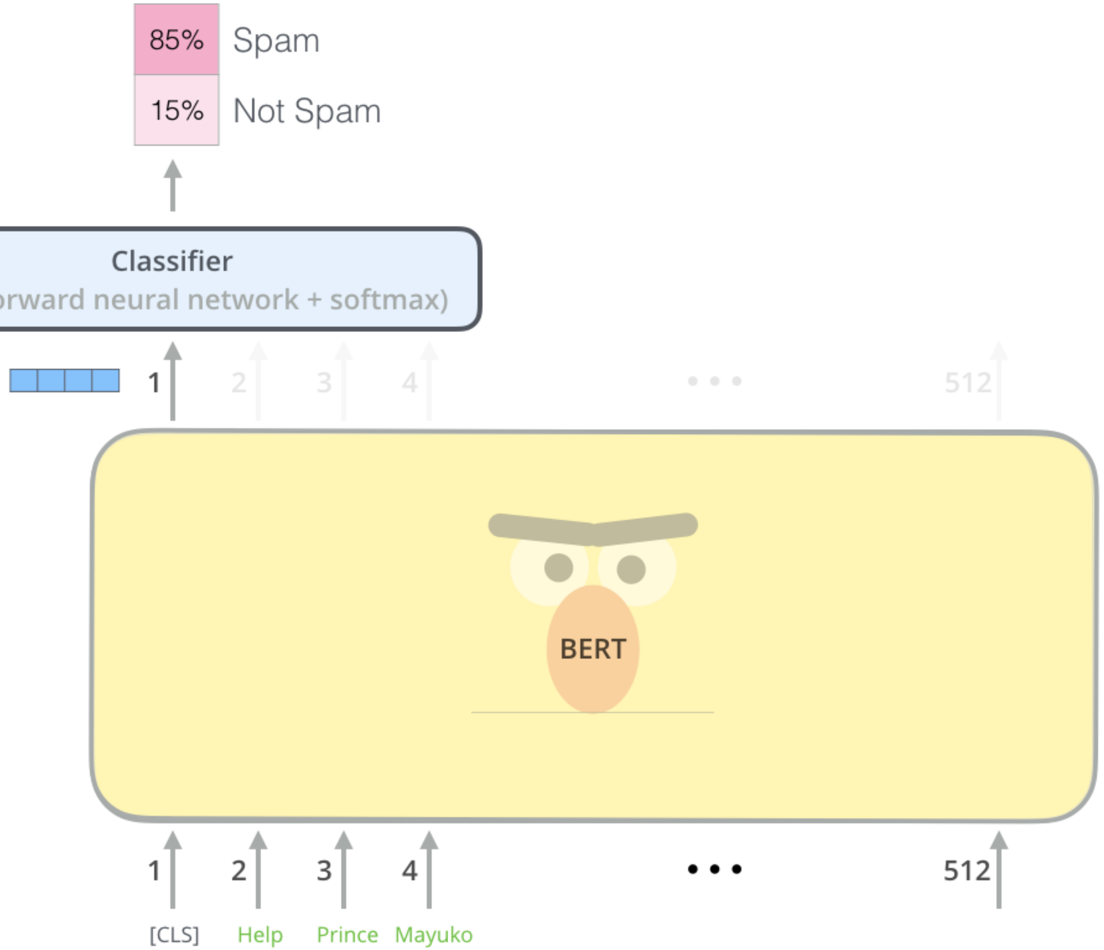
BERT: Pre-training Procedure

- Two sentence task

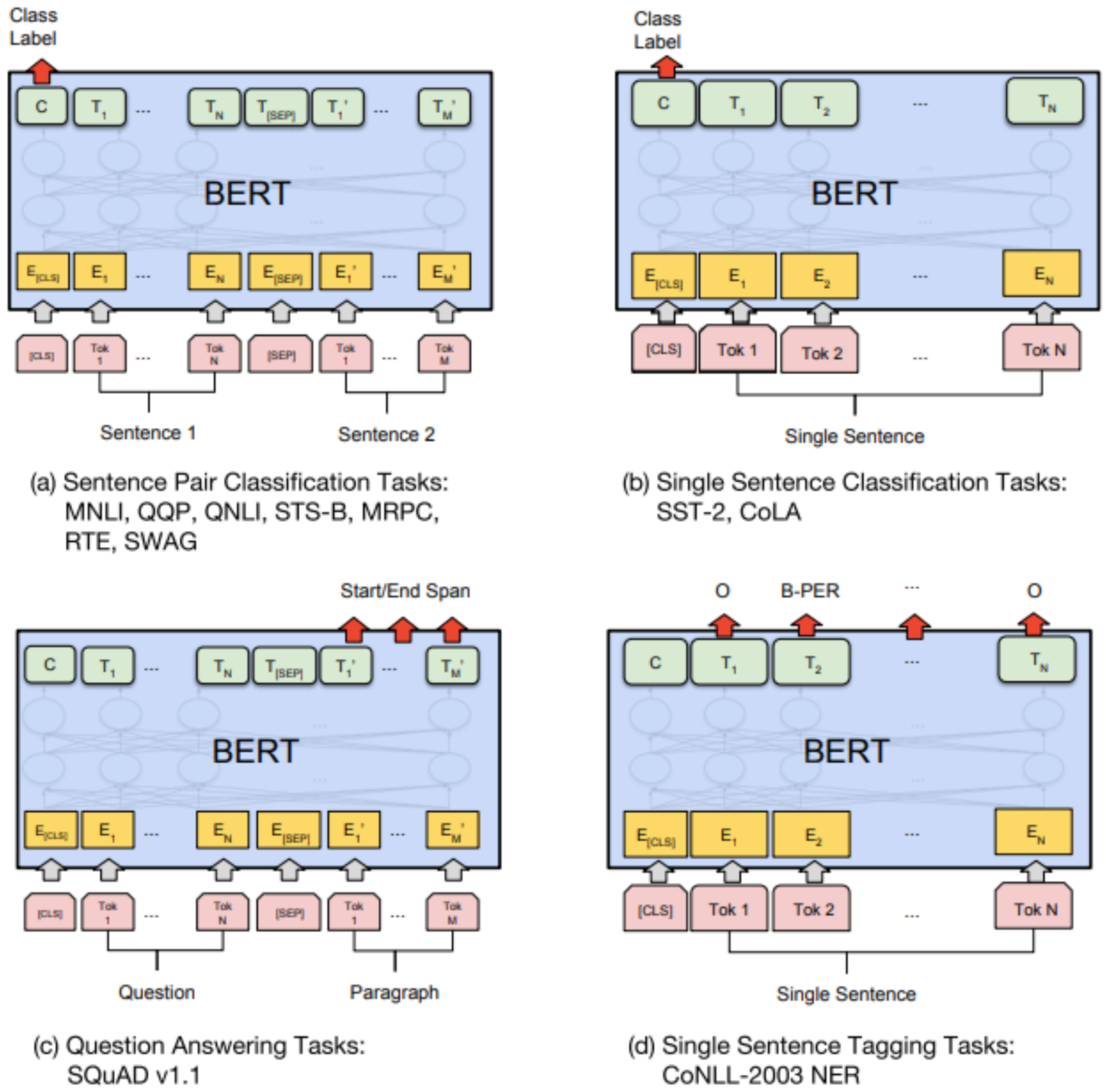


BERT: Downstream Fine-tuning

- Use BERT for sentence classification



BERT: Downstream Fine-tuning



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

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

(c) Question Answering Tasks: SQuAD v1.1

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

BERT Results

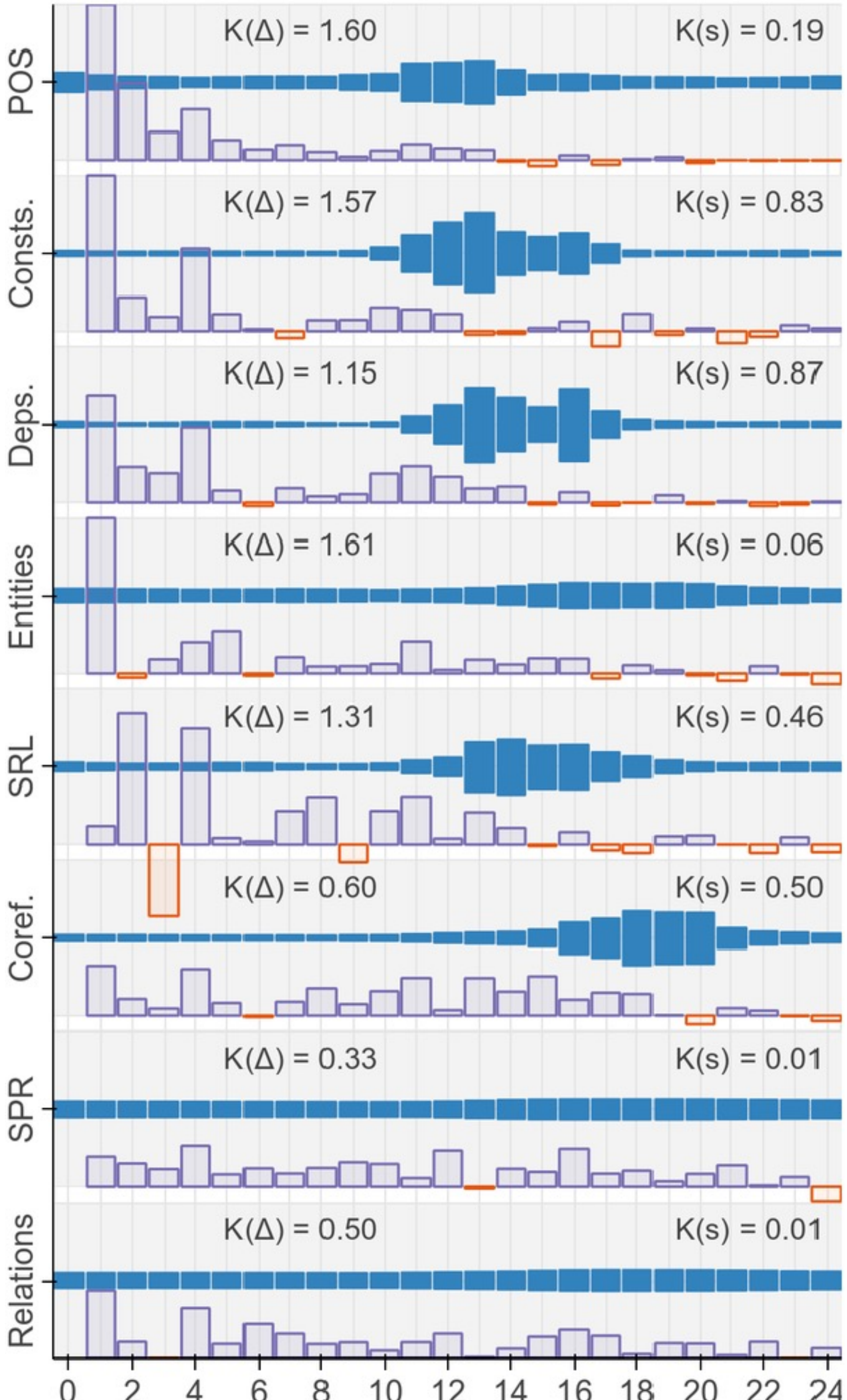
- Huge improvements over SOTA on 12 NLP task

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/.

Analysis

- *BERT Rediscovers the Classical NLP Pipeline.* Tenney et al., 2019

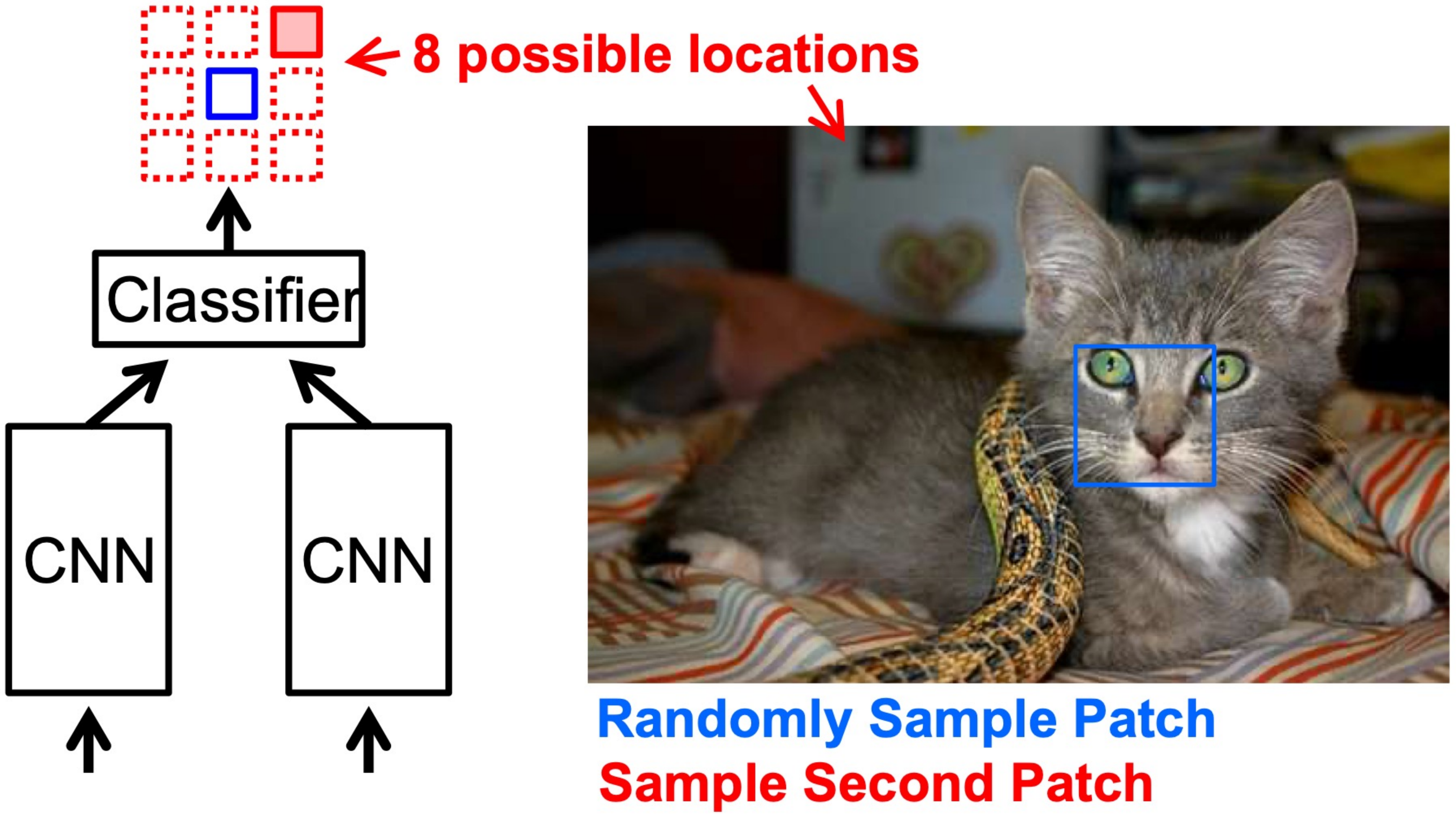


Self-supervised learning for other modalities: quick overview

- SSL on images
- SSL on videos

SSL from Images, EX (I): relative positioning

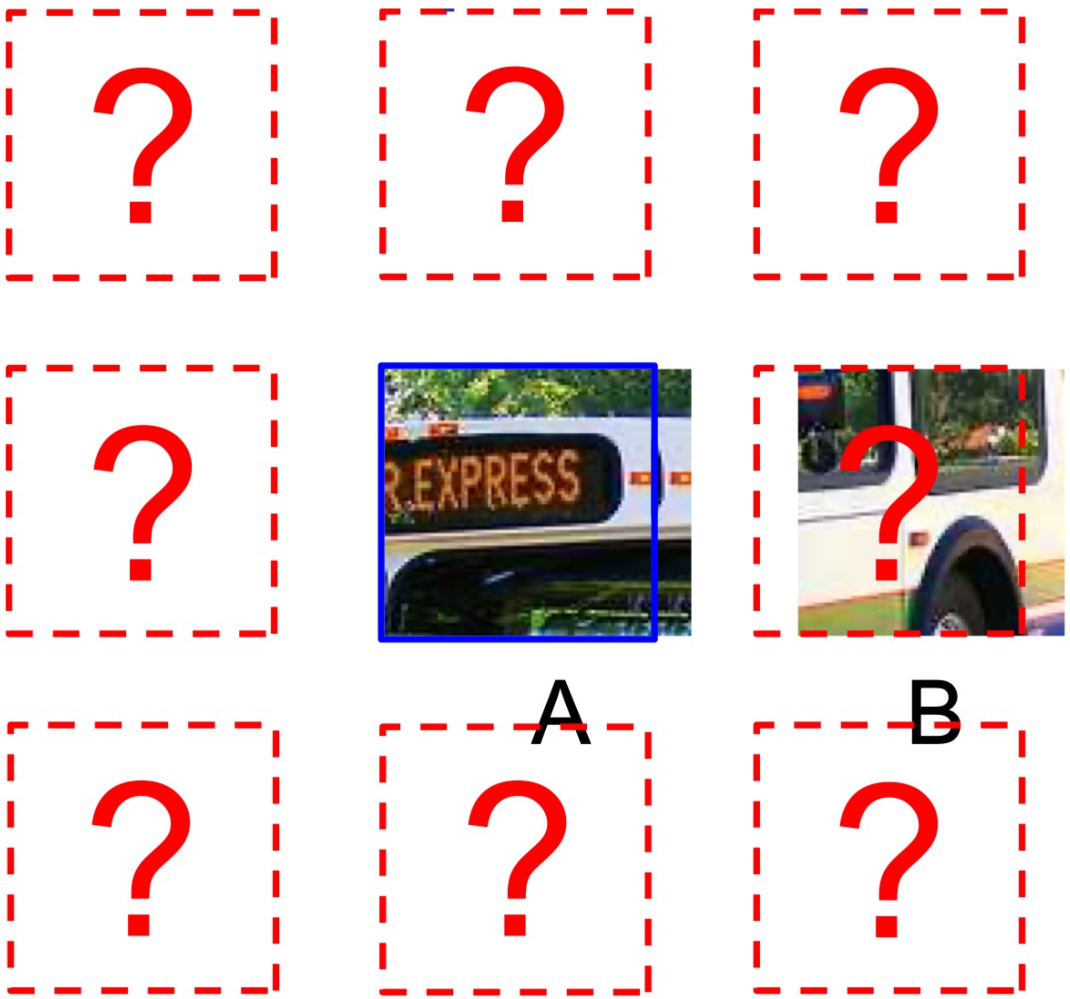
Train network to predict relative position of two regions in the same image



Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Images, EX (I): relative positioning



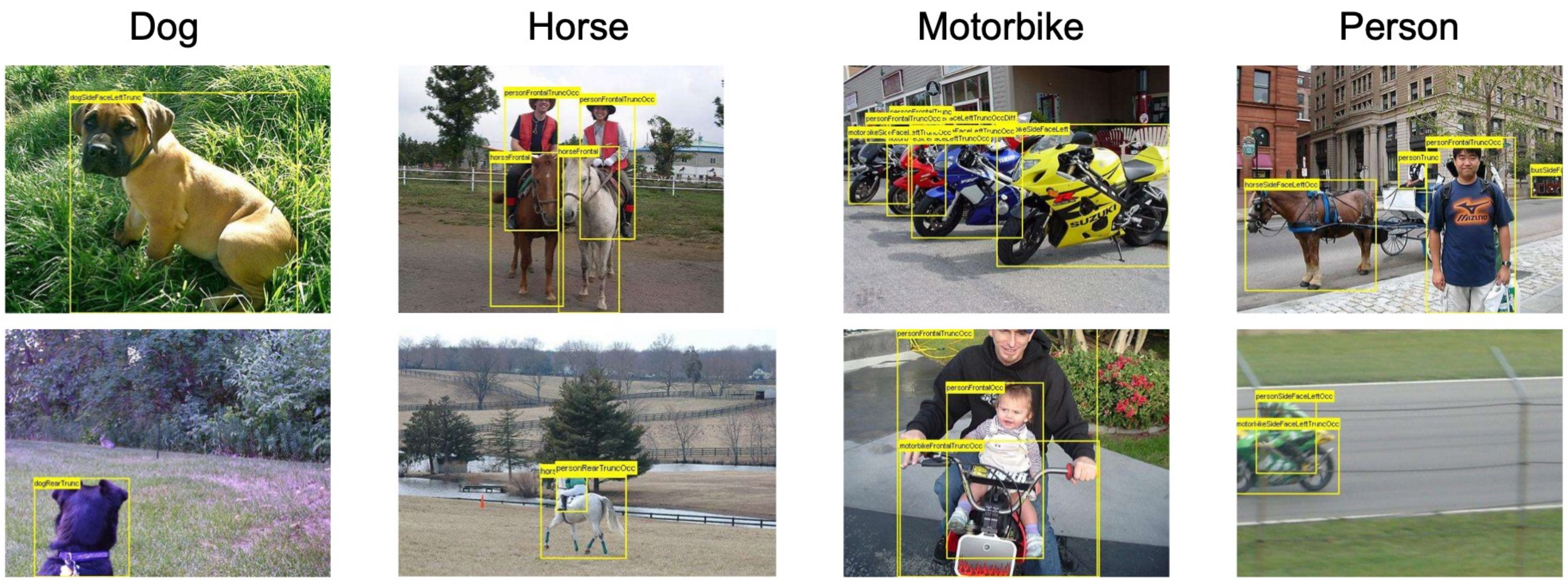
Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (I): relative positioning

Evaluation: PASCAL VOC Detection

- 20 object classes (car, bicycle, person, horse …)
- Predict the bounding boxes of all objects of a given class in an image (if any)



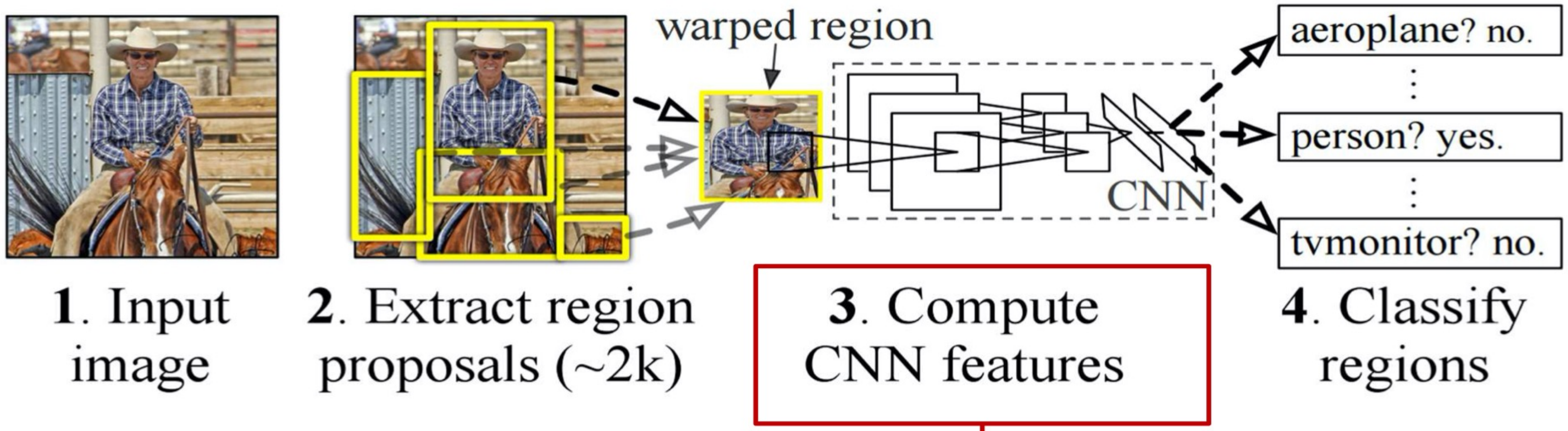
[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Images, EX (I): relative positioning

Evaluation: PASCAL VOC Detection

- Pre-train CNN using self-supervision (no labels)
- Train CNN for detection in R-CNN object category detection pipeline

R-CNN



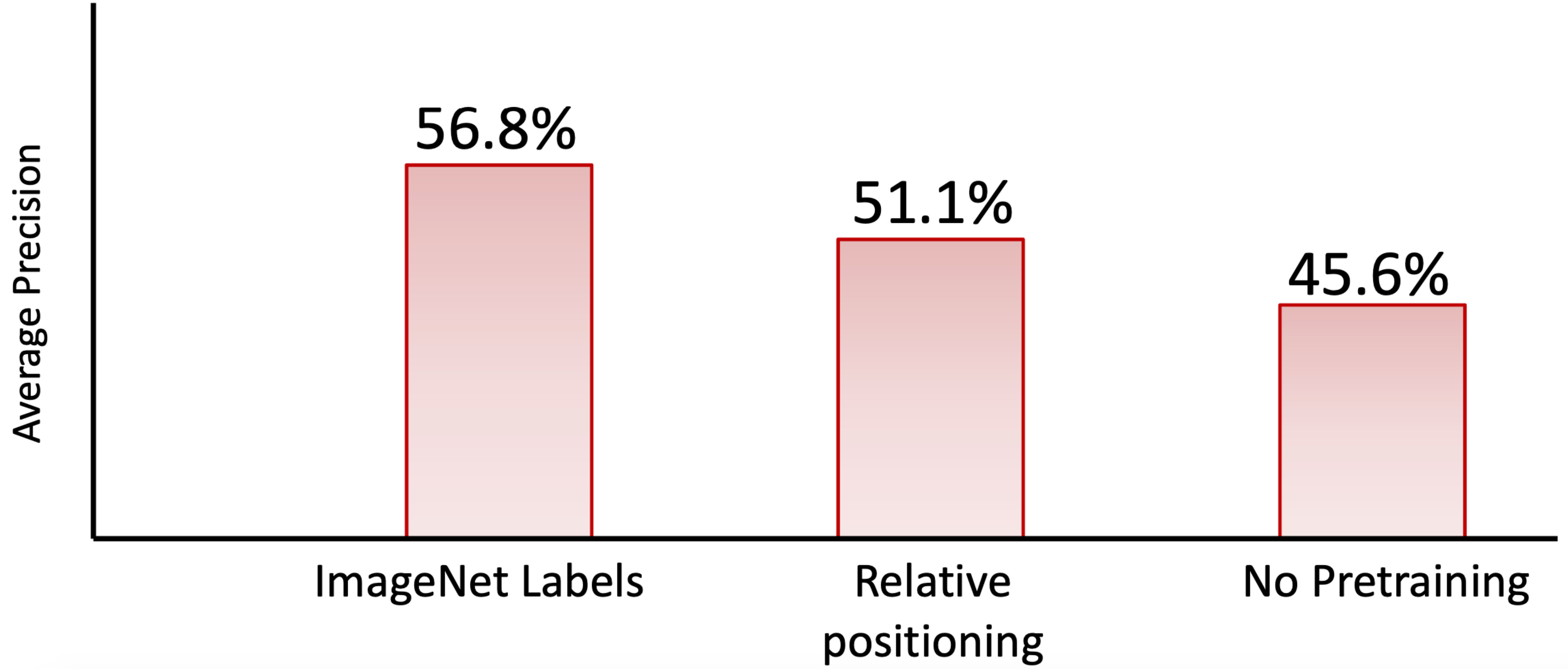
Pre-train on relative-position task, w/o labels

[Girshick et al. 2014]

[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Images, EX (I): relative positioning

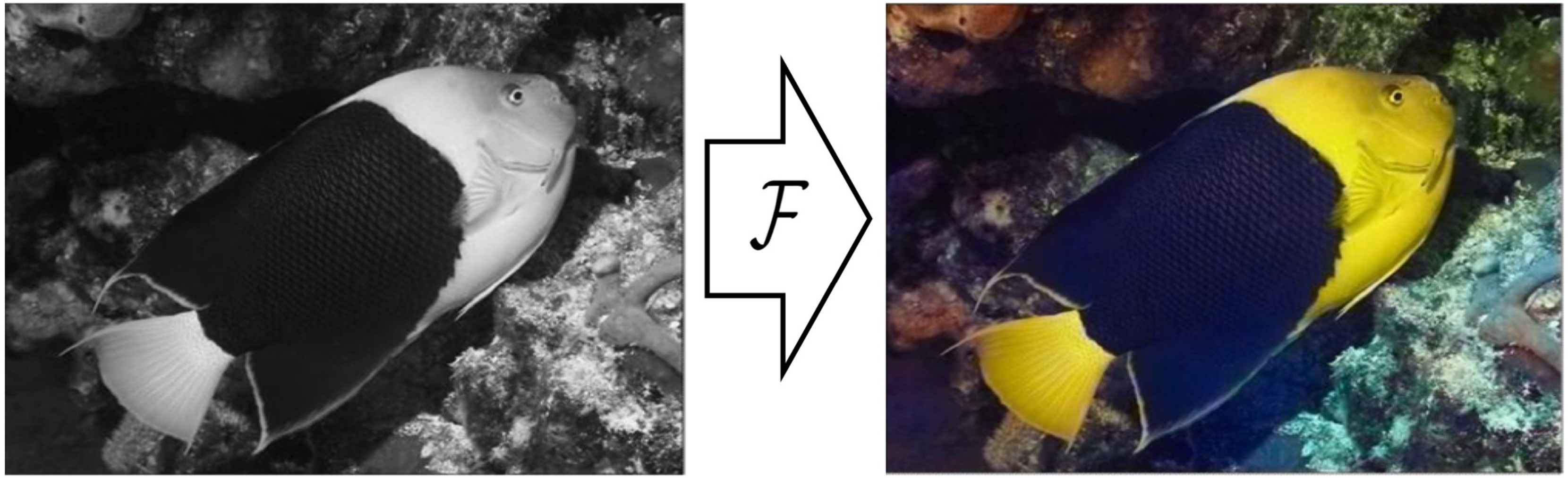
Evaluation: PASCAL VOC Detection



[Courtesy: Zisserman “Self-supervised Learning”]

SSL from Images, EX (II): colorization

Train network to predict pixel colour from a monochrome input

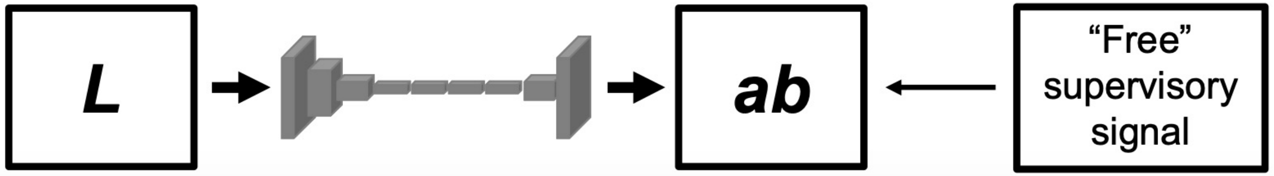


Grayscale image: *L* channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

Concatenate (L,ab)

$$(\mathbf{X}, \widehat{\mathbf{Y}})$$



[Courtesy: Zisserman "Self-supervised Learning"]

Colorful Image Colorization, Zhang et al., ECCV 2016

SSL from Images, EX (II): colorization

Train network to predict pixel colour from a monochrome input

[Courtesy: Zisserman “Self-supervised Learning”]

Colorful Image Colorization, Zhang et al., ECCV 2016

SSL from Images, EX (III): exemplar networks

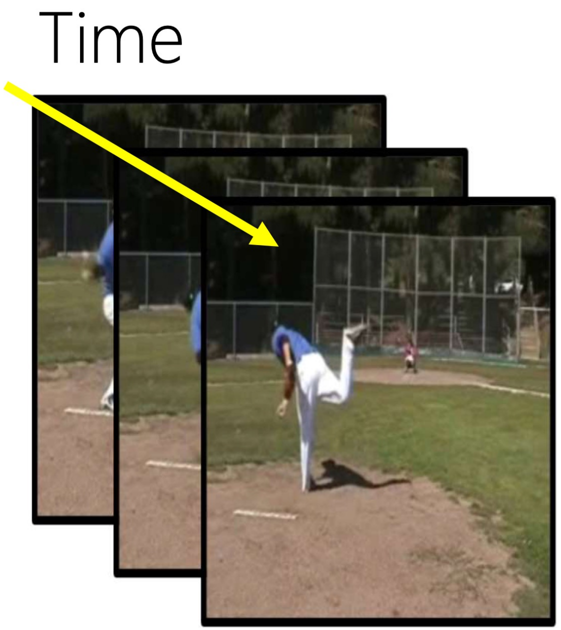
- Exemplar Networks (Dosovitskiy et al., 2014)
- Perturb/distort image patches, e.g. by cropping and affine transformations
- Train to classify these exemplars as same class

[Courtesy: Zisserman "Self-supervised Learning"]

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?



“Sequence” of data

[Courtesy: Zisserman “Self-supervised Learning”]

Wei et al., 2018 Arrow of Time

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?
- Video direction
  - Predict if video playing forwards or backwards

[Courtesy: Zisserman “Self-supervised Learning”]

Wei et al., 2018 Arrow of Time

SSL from Videos

Three example tasks:

- Video sequence order
  - Sequential Verification: Is this a valid sequence?
- Video direction
  - Predict if video playing forwards or backwards
- Video tracking
  - Given a color video, colorize all frames of a gray scale version using a reference frame

[Courtesy: Zisserman "Self-supervised Learning"]

Vondric et al., 2018

Key Takeaways

- Self supervision learning
  - Predicting any part of the observations given any available information
  - The prediction task forces models to learn semantic representations
  - Massive/unlimited data supervisions
- SSL for text:
  - Language models: next word prediction
  - Word embedding: skip-gram
  - BERT text representations: masked language model (MLM)
- SSL for images/videos:
  - Various ways of defining the prediction task

Contrastive Learning

Contrastive learning

- Take a data example x, sample a "positive" sample x_{pos} and "negative" samples x_{neg} in some way
- Then try fit a scoring model such that

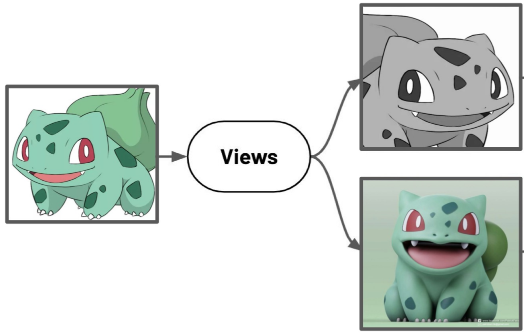
$$score(x, x_{pos}) > score(x, x_{neg})$$

Credit: [CVPR 2021 Tutorial] Leave Those Nets Alone: Advances in Self-Supervised Learning

Contrastive learning

- Take a data example x, sample a "positive" sample x_{pos} and "negative" samples x_{neg} in some way

"positive" sample:
- Data of the same labels
- Data of the same pseudo-labels
- Augmented/distorted version of x
- Data that captures the same target from different views



"negative" sample:
- Randomly sampled data
- Hard negative sample mining

Contrastive learning

- Take a data example x, sample a "positive" sample x_{pos} and "negative" samples x_{neg} in some way
- Then try fit a scoring model such that

$$score(x, x_{pos}) > score(x, x_{neg})$$

Credit: [CVPR 2021 Tutorial] Leave Those Nets Alone: Advances in Self-Supervised Learning

Contrastive learning: Ex 1

Learning a similarity metric discriminatively

Sample a pair of images and compute their distance:

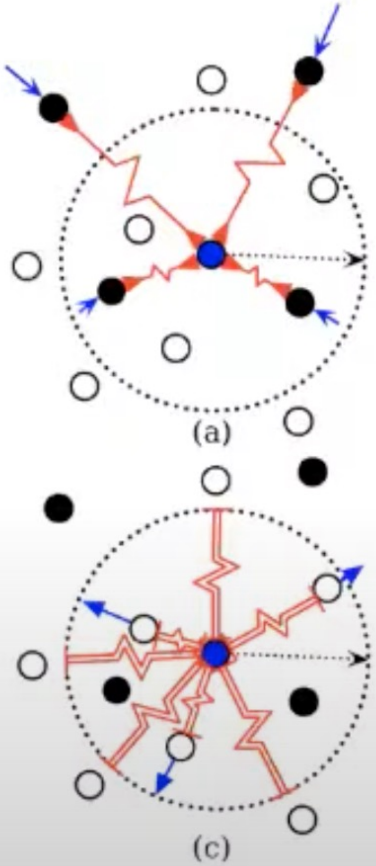
$$D_i = \|x, x_i\|_2$$

If **positive** sample:

$$L_i = D_i^2$$

If **negative** sample:

$$L_i = \max\left(0, \epsilon - D_i\right)^2$$



[Chopra et al., 2005; Hadsell et al., 2006]

Credit: [CVPR 2021 Tutorial] Leave Those Nets Alone: Advances in Self-Supervised Learning

Common contrastive learning functions

- Contrastive loss (Chopra et al. 2005)
- Triplet loss (Schroff et al. 2015; FaceNet)
- Lifted structured loss (Song et al. 2015)
- Multi-class n-pair loss (Sohn 2016)
- Noise contrastive estimation (“NCE”; Gutmann & Hyvarinen 2010)
- InfoNCE (van den Oord, et al. 2018)
- Soft-nearest neighbors loss (Salakhutdinov & Hinton 2007, Frosst et al. 2019)

[Courtesy: Weng & Kim, NeurIPS 2021 tutorial]

Contrastive learning: Ex 2

- SimCSE ("Simple Contrastive learning of Sentence Embeddings"; Gao et al. 2021)
  - Predict a sentence from itself with only dropout noise
  - One sentence gets two different versions of dropout augmentations

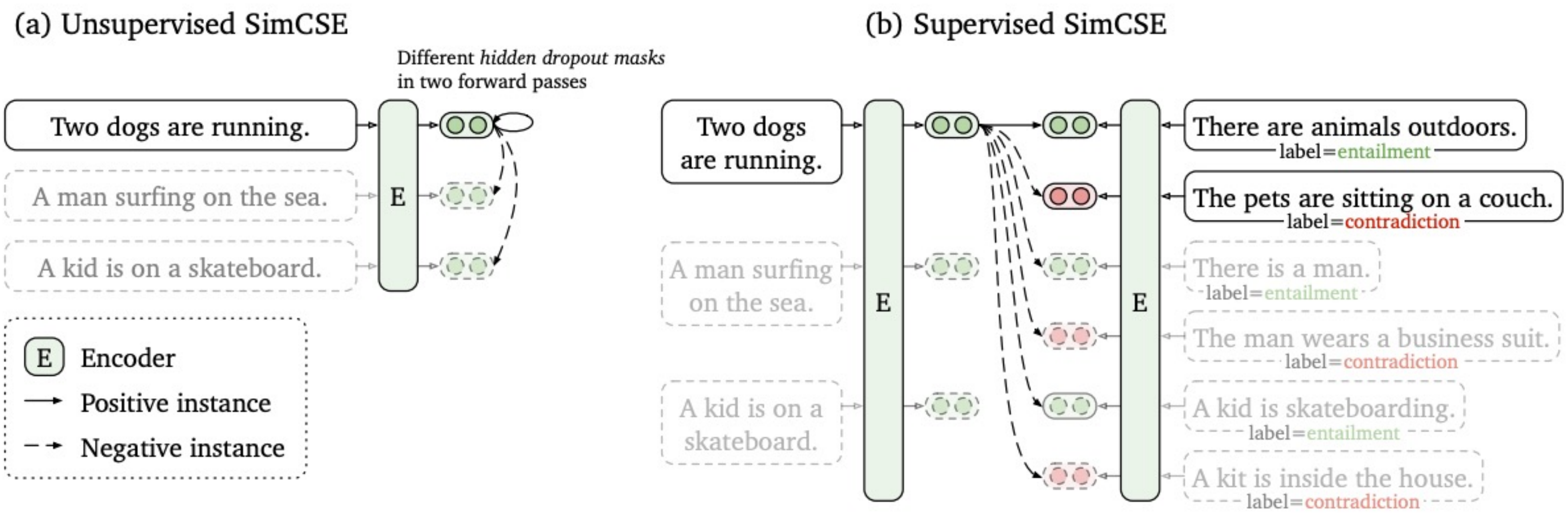
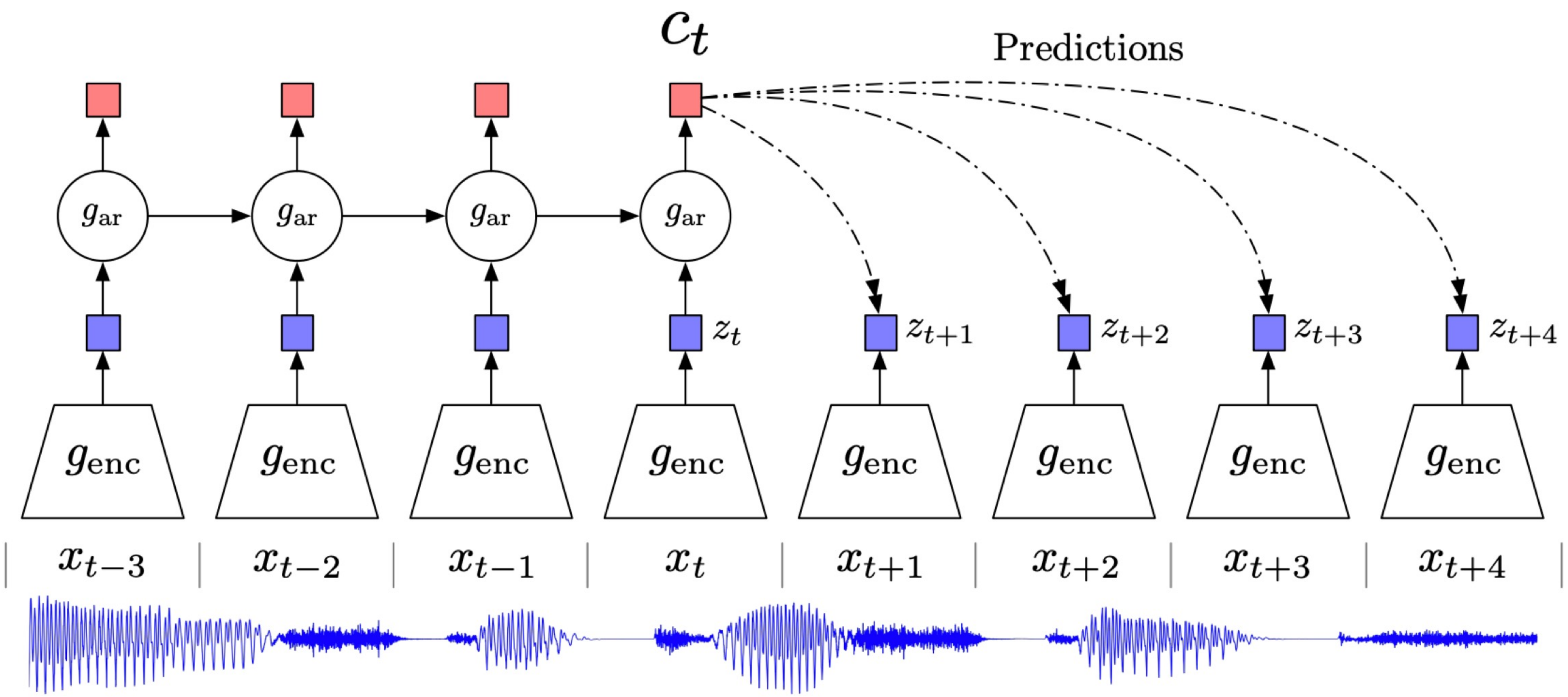


Figure 1: (a) Unsupervised SimCSE predicts the input sentence itself from in-batch negatives, with different hidden dropout masks applied. (b) Supervised SimCSE leverages the NLI datasets and takes the entailment (premise-hypothesis) pairs as positives, and contradiction pairs as well as other in-batch instances as negatives.

Contrastive learning: Ex 3 - InfoNCE

- The CPC model
  - c_t: context representation from history
  - x_{t+k} (or z_{t+k}): future target



[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

InfoNCE loss

- Define scoring function $f_k > 0$
- The InfoNCE loss:
  - Given $X = \{$ one positive sample from $p(x_{t+k}|\, c_t)$, $N-1$ negative samples from the negative sampling distribution $p(x_{t+k})$ $\}$

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

- InfoNCE is interesting because it's effectively maximizing the mutual information between c_t and x_{t+k}

[van den Oord et al., "Representation Learning with Contrastive Predictive Coding"]

Mutual Information (MI)

- How much is our uncertainty about x reduced by knowing c ?

$$I(x;c) = \sum_{x,c} p(x,c) \log \frac{p(x,c)}{p(x)p(c)} = \sum_{x,c} p(x,c) \log \frac{p(x|c)}{p(x)}$$

$$= H(x) + H(c) - H(x,c)$$

$$= H(x) - H(x|c)$$

$$= KL(p(x,c) \,||\, p(x)p(c))$$

[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

Minimizing InfoNCE <=> Maximzing MI

- InfoNCE loss

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

- The loss is optimized when

$$f_k(x_{t+k}, c_t) \propto \frac{p(x_{t+k}|c_t)}{p(x_{t+k})}$$

  - Proof:

$$p(sample\ i\ is\ positive|X, c_t) = \frac{p(x_i|c_t)\prod_{l\neq i} p(x_l)}{\sum_{j=1}^{N} p(x_j|c_t)\prod_{l\neq j} p(x_l)}$$

$$= \frac{\frac{p(x_i|c_t)}{p(x_i)}}{\sum_{j=1}^{N} \frac{p(x_j|c_t)}{p(x_j)}}.$$

[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

- How does this loss maximize the mutual information?

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

- How does this loss maximize the mutual information?

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

$$\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} = -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right]$$

Use proportionality condition

[van den Oord et al., "Representation Learning with Contrastive Predictive Coding"]

- How does this loss maximize the mutual information?

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

$$\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} = -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right]$$

$$= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}\right]$$

Take -ve inside log

[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

- How does this loss maximize the mutual information?

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

$$\begin{aligned}
\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right] \\
&= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}\right] \\
&\approx \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathop{\mathbb{E}}_{x_j} \frac{p(x_j|c_t)}{p(x_j)}\right]
\end{aligned}$$

This approximation becomes more accurate as N increases, so it is preferable to use large negative samples

[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

- How does this loss maximize the mutual information?

$$\mathcal{L}_{\mathrm{N}} = -\mathop{\mathbb{E}}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

$$\begin{aligned}
\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\mathop{\mathbb{E}}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}}\right] \\
&= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}\right] \\
&\approx \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathop{\mathbb{E}}_{x_j} \frac{p(x_j|c_t)}{p(x_j)}\right] \quad = 1 \\
&= \mathop{\mathbb{E}}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1)\right]
\end{aligned}$$

[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

- How does this loss maximize the mutual information?

$$\mathcal{L}_{\mathrm{N}} = -\mathbb{E}_{X}\left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right]$$

$$\begin{aligned}
\mathcal{L}_{\mathrm{N}}^{\mathrm{opt}} &= -\mathbb{E}_{X} \log \left[\frac{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})}}{\frac{p(x_{t+k}|c_t)}{p(x_{t+k})} + \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)}} \right] \\
&= \mathbb{E}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} \sum_{x_j \in X_{\mathrm{neg}}} \frac{p(x_j|c_t)}{p(x_j)} \right] \\
&\approx \mathbb{E}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \mathbb{E}_{x_j} \frac{p(x_j|c_t)}{p(x_j)} \right] \\
&= \mathbb{E}_{X} \log \left[1 + \frac{p(x_{t+k})}{p(x_{t+k}|c_t)} (N-1) \right] \\
&\geq \mathbb{E}_{X} \log \left[\frac{p(x_{t+k})}{p(x_{t+k}|c_t)} N \right] \\
&= -I(x_{t+k}, c_t) + \log(N),
\end{aligned}$$

[van den Oord et al., “Representation Learning with Contrastive Predictive Coding”]

- How does this loss maximize the mutual information?

$$\mathcal{L}_{\mathrm{N}} = - \mathop{\mathbb{E}}_{X} \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)} \right]$$

$$I(x_{t+k}, c_t) \geq \log(N) - \mathcal{L}_{\mathrm{N}}$$

[van den Oord et al., "Representation Learning with Contrastive Predictive Coding"]

Key Takeaways: Contrastive learning

- Contrastive learning is a way of doing self-supervised learning
- Positive samples, negative samples
- Mutual information

$$I(x;c) = \sum_{x,c} p(x,c) \log \frac{p(x,c)}{p(x)p(c)} = \sum_{x,c} p(x,c) \log \frac{p(x|c)}{p(x)}$$

$$= H(x) + H(c) - H(x,c)$$

$$= H(x) + H(x|c)$$

$$= KL(p(x,c) \,||\, p(x)p(c))$$

  - InfoNCE ⇔ MI

Questions?