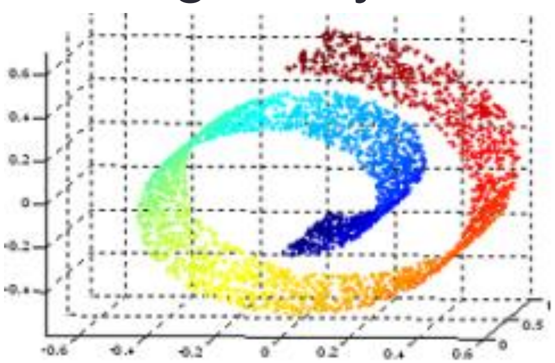
Text Representation

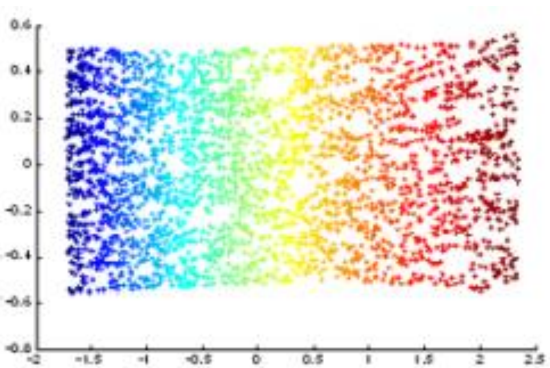
Representation learning

What is an embedding?

- "Latent space" or "embedding space" refers to a low-dimensional representation of high-dimensional data
  - In neural network, the mapping from original data to the embedding space is often linear.
    - Ex of linear mapping/projection: PCA
- Mapping of these embeddings are one of the key tricks in deep learning today



High dimension



Low dimension

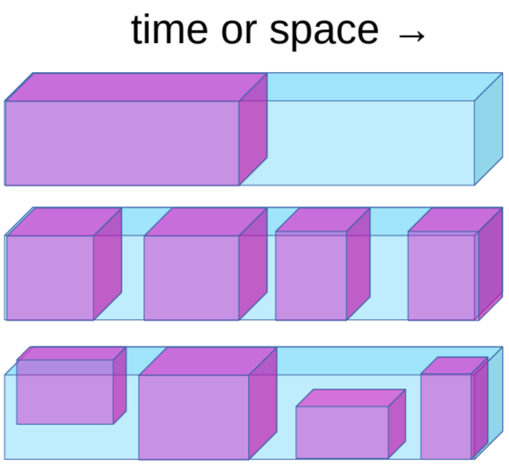
Embeddings

- Can be trained by supervised or self-supervised techniques

Y. LeCun

Self-Supervised Learning = Filling in the Blanks

- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the masked from the visible.**
- **Predict the any occluded part from all available parts.**



- **Pretend there is a part of the input you don't know and predict that.**
- **Reconstruction = SSL when any part could be known or unknown**

https://twitter.com/ylecun/status/1226838002787344391?lang=en

Outline

- Contrastive learning
- Sentence embeddings
  - MUSE
  - SimCSE
  - BGE
  - CLIP

Contrastive learning

- An important technique for self-supervised training is contrastive learning
  - Similar things should have similar embeddings
  - Different things should have different embeddings
- Example: negative sampling loss in word2vec

$$J_t(\theta) = \underbrace{\boxed{\log \sigma\left(u_o^T v_c\right)}}_{\text{Context word (positive, +1)}} + \underbrace{\boxed{\sum_{i=1}^{k} \mathbb{E}_{j \sim P(w)} \left[\log \sigma\left(-u_j^T v_c\right)\right]}}_{\text{Negative samples (negative, -1)}}$$

Types of contrastive learning

- Triplet loss
- InfoNCE loss

Triplet loss

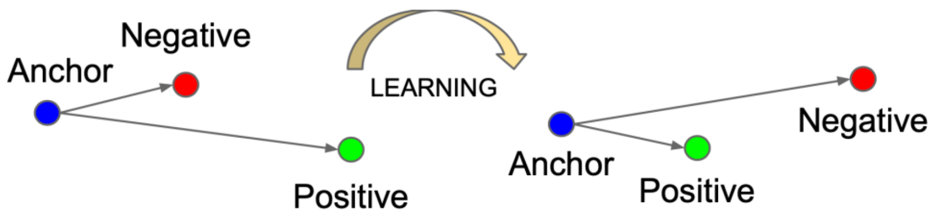
- Triplet loss considers an anchor, a positive, and a negative
- Requires mining of hard negative samples

margin

$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

Take positive only max(0,x)

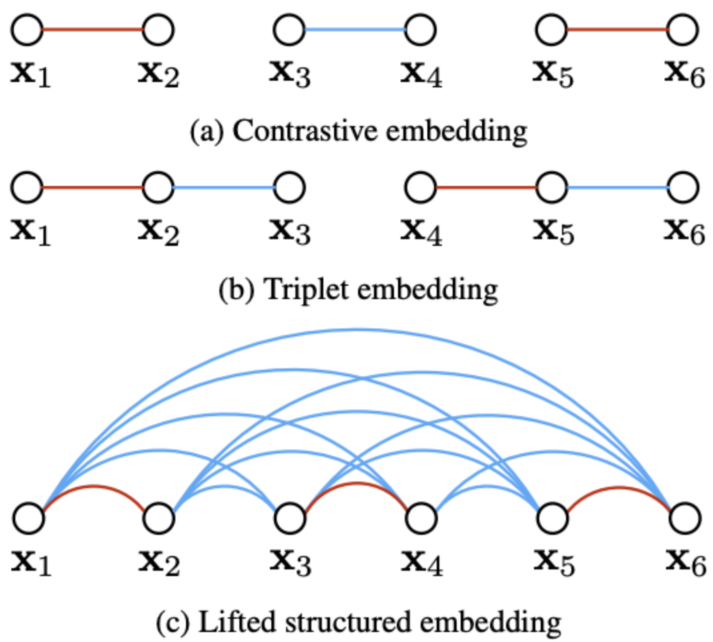
Example triplet loss



https://arxiv.org/abs/1503.03832

Dealing with minibatches

- Since we train in minibatches, most modern losses pair positive and negative samples within a minibatch for more efficient computation
  - Compute all pairwise distance within the minibatch



https://arxiv.org/pdf/1511.06452.pdf

NCE (Noise constrastive estimation) loss

- Maximize training data probability while reducing noise probability.
- Learn in a constrastive way to reduce overhead for normalization
  - Max LogP(data) – Log P(noise or negative samples)
  - Ex: used to train word embeddings such as W2V, too many classes in the softmax output

http://proceedings.mlr.press/v9/gutmann10a.html

InfoNCE

- Similar to NCE but just for categorical cross entropy (instead of binary cross entropy) https://arxiv.org/pdf/1807.03748.pdf

- Effectively maximize mutual information between c and positive x

$$L_{InfoNCE} = -E[log \frac{f(x,c)}{\sum_{x'} f(x',c)}]$$

$$f(x,c) = exp(\mathbf{z}^T W c)$$

z is encoded x

- f( ) can be any function that describes similarity
- Can be extended to have multiple positive examples in a batch (soft nearest neighbor loss) https://arxiv.org/abs/1902.01889

Soft nearest neighbor loss

- Multiple positive and negative
- Adds temperature (either hyperparameter, or learned)
  - Weights the gradient size, helps model learn form hard negatives

Definition. The *soft nearest neighbor loss* at temperature T, for a batch of b samples (x, y), is:

$$l_{sn}(x, y, T) = -\frac{1}{b} \sum_{i \in 1..b} \log \left(\frac{\sum\limits_{\substack{j \in 1..b \\ j \neq i \\ y_i = y_j}} e^{-\frac{||x_i - x_j||^2}{T}}}{\sum\limits_{\substack{k \in 1..b \\ k \neq i}} e^{-\frac{||x_i - x_k||^2}{T}}} \right) \quad (1)$$

https://arxiv.org/pdf/1902.01889.pdf

Contrastive summary

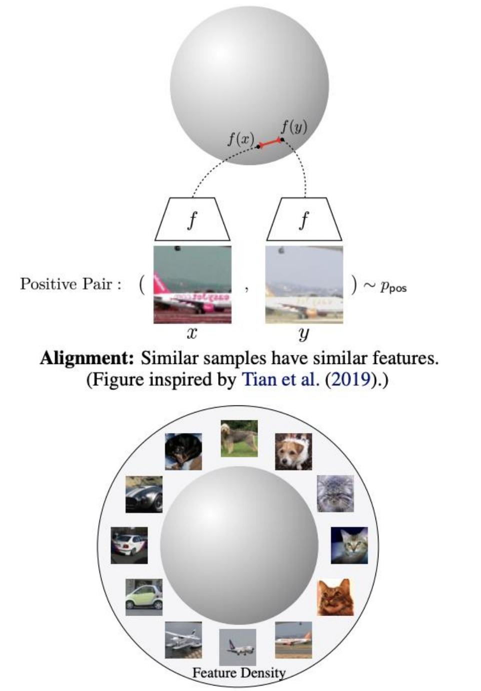
- The most common form you will see for contrastive learning is

$$\mathcal{L}^{\text{NT-Xent}} = -\frac{1}{n} \sum_{i,j \in \mathcal{MB}} \log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2n} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

$$\tau \mathcal{L}^{\text{NT-Xent}} = \underbrace{-\frac{1}{n} \sum_{i,j} \text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)}_{\mathcal{L}_{\text{alignment}}} + \underbrace{\frac{\tau}{n} \sum_{i} \log \sum_{k=1}^{2n} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}_{\mathcal{L}_{\text{distribution}}}$$

Encourage similar things to align

Encourage embeddings to spread uniformly in the hypersphere



Alignment: Similar samples have similar features. (Figure inspired by Tian et al. (2019).)

Uniformity: Preserve maximal information.

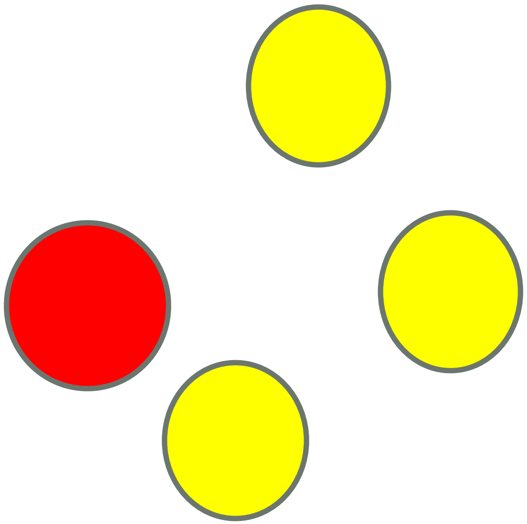
Figure 1: Illustration of alignment and uniformity of feature distributions on the output unit hypersphere. STL-10 (Coates et al., 2011) images are used for demonstration.

- People often refer to this as contrastive loss, InfoNCE loss, normalized temperature scaled CE loss, …

https://arxiv.org/abs/2005.10242 https://arxiv.org/abs/2011.02803 https://arxiv.org/abs/2002.05709

Key details to contrastive loss works

- Large batch
- Hard/semi-hard negative mining
- Augmentation on the anchor and postive
- Other tricks includes - adding classification/supervised loss (CE/softmax loss)



Outline

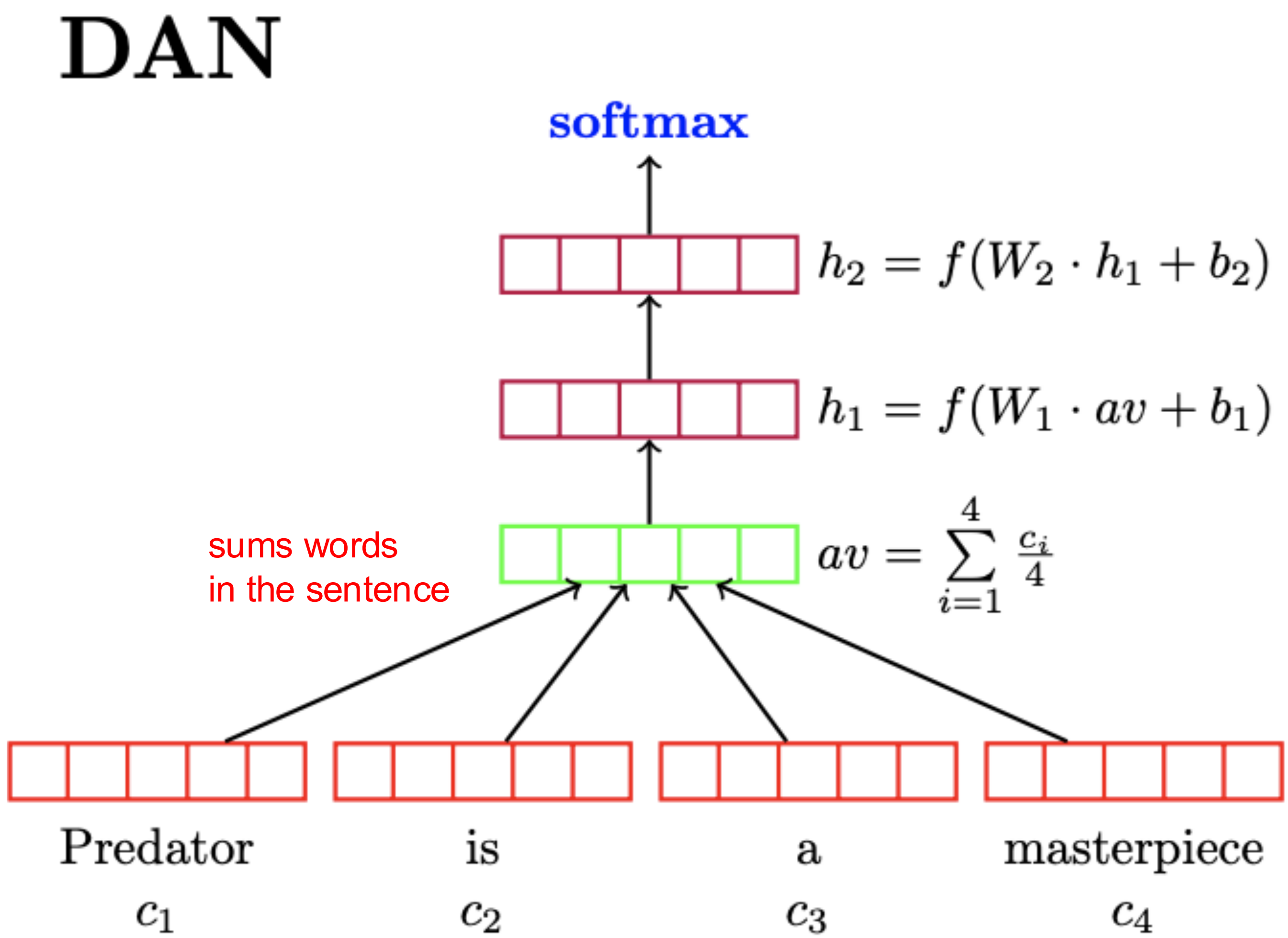
- Contrastive learning
- Sentence embeddings
  - MUSE
  - SimCSE
  - BGE
  - CLIP

Sentence representation

- How would we create a sentence embedding?
- Compositionality from words/tokens!
  - Sum, max
  - Recurrence
  - Attention

MUSE

Deep Averaging Networks (DAN)



https://www.aclweb.org/anthology/P15-1162/ 2015

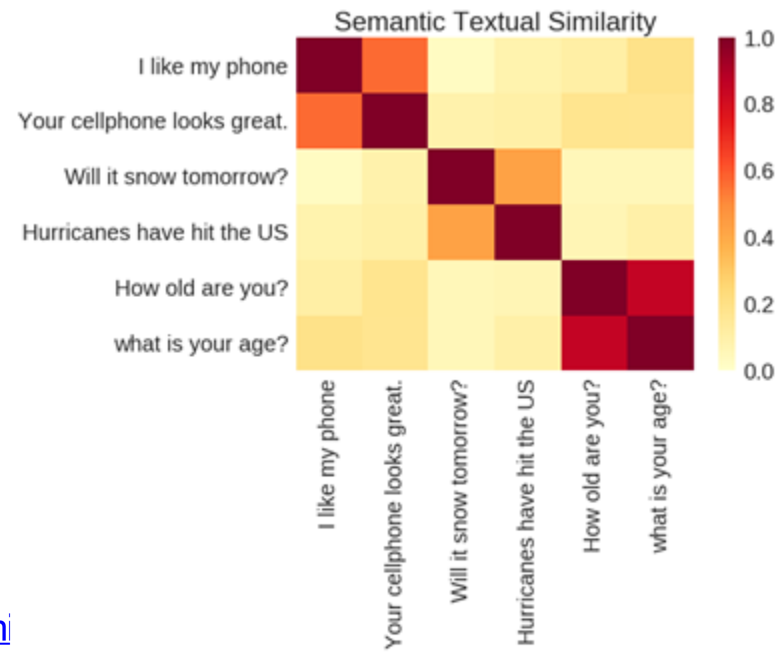
Universal Sentence Encoder (USE)

A model focusing on sentence representation

Use sentencepiece tokenization

Pre-trained then used anywhere

Based on (1) DAN (lite version) or (2) Transformer



Official implementation with pretrained weights

https://tfhub.dev/google/collections/universal-sentence-encoder/1

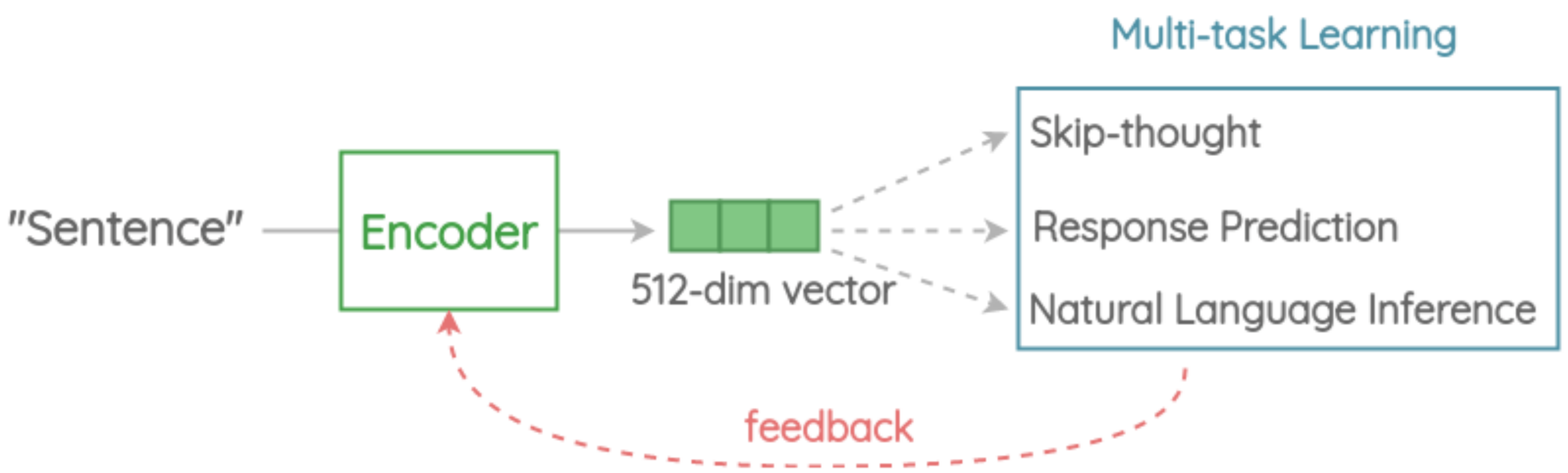
https://ai.googleblog.com/2018/05/advances-in-semantic-textual-simi

https://www.kaggle.com/models/google/universal-sentence-encoder

Pretraining USE

Training done using multi-task

1) Skip-thought
2) Response prediction
3) Natural language inference (NLI)

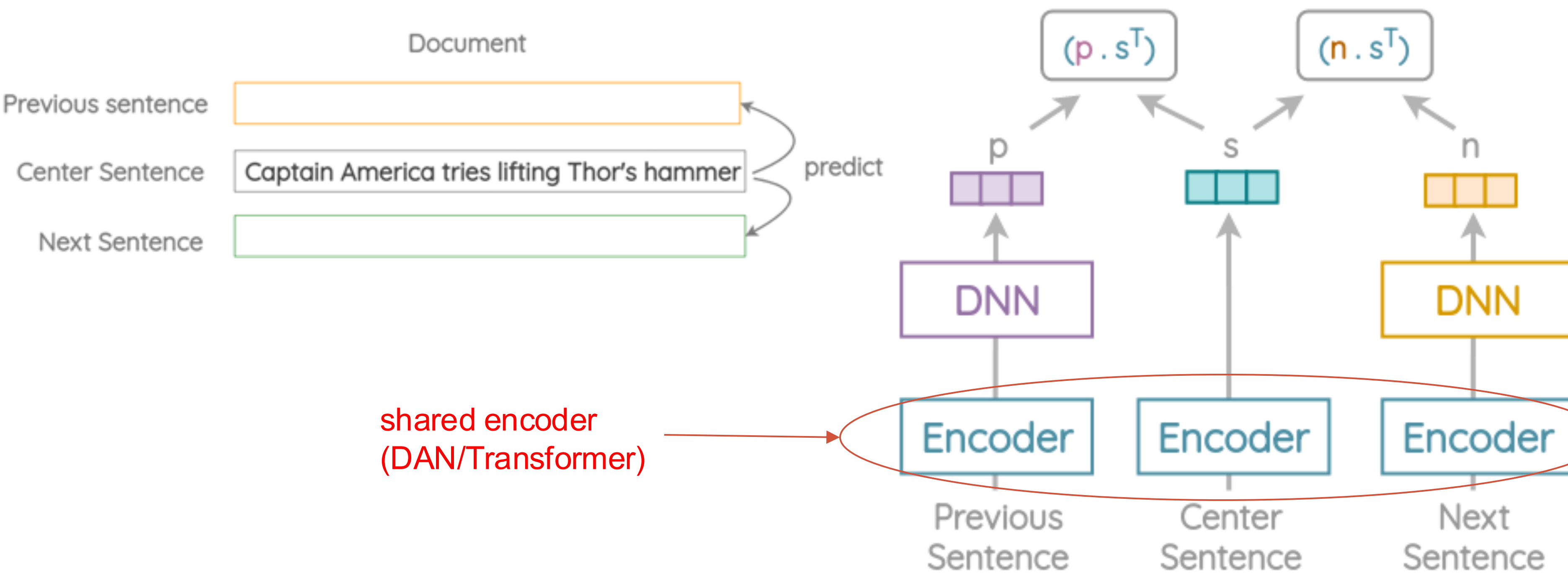


Picture credit: https://amitness.com/2020/06/universal-sentence-encoder/

Skip-thought task

Similar to skip-gram, use the middle to predict context

Unsupervised



Skip-thought Task Structure

Picture credit: https://amitness.com/2020/06/universal-sentence-encoder/

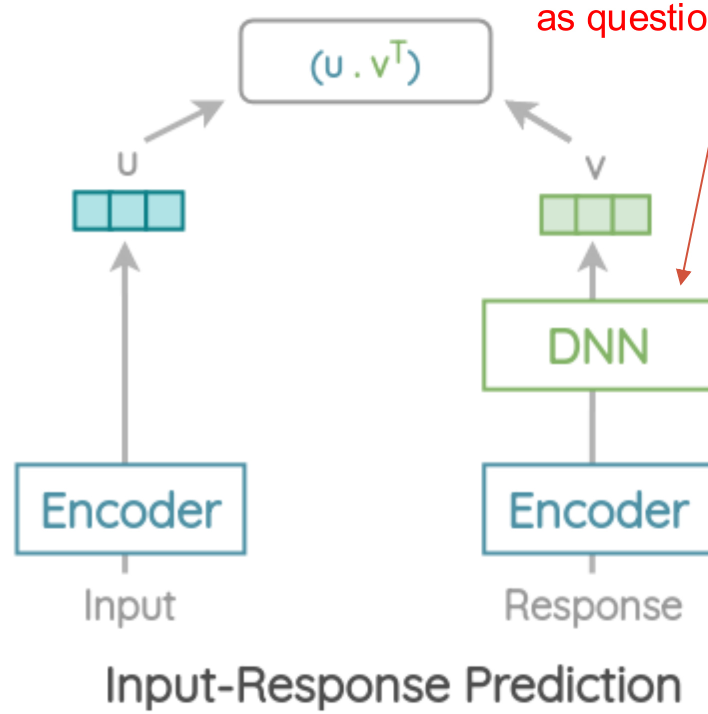
Response prediction

Match questions and answers in internet forum (scraped)

Supervised (free labels)

To map answer to the same space as question

similar sentence gives similar response



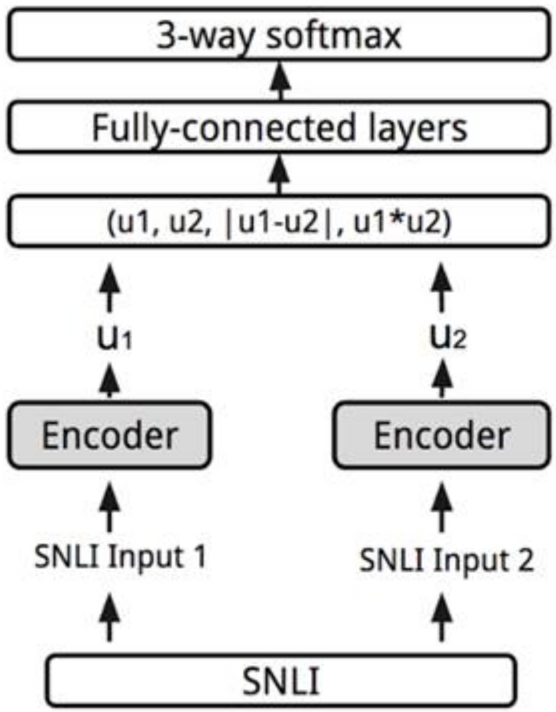
Picture credit: https://amitness.com/2020/06/universal-sentence-encoder/

Natural Language Inference

Predict relationship between sentence

Supervised

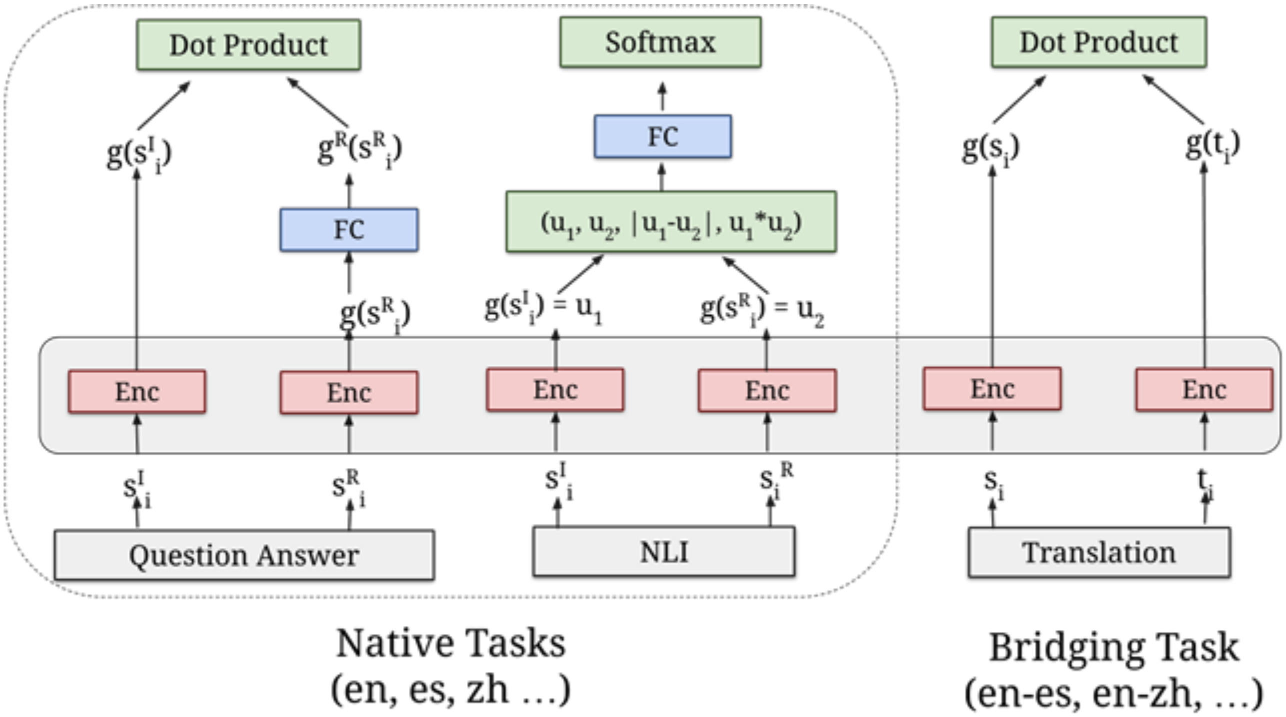
Premise	Hypothesis	Judgement
A soccer game with multiple males playing	Some men are playing a sport	entailment
I love Marvel movies	I hate Marvel movies	contradiction
I love Marvel movies	A ship arrived	neutral



Multilingual USE

Can train to map multiple languages to the same presentation.

Can handle code switching, has Thai!



https://ai.googleblog.com/2019/07/multilingual-universal-sentence-encoder.html

Download-ables

cmlm-en-base

cmlm-en-large

cmlm-multilingual-base

cmlm-multilingual-base-br

cmlm-multilingual-preprocess

large

multilingual

multilingual-large

multilingual-qa

qa

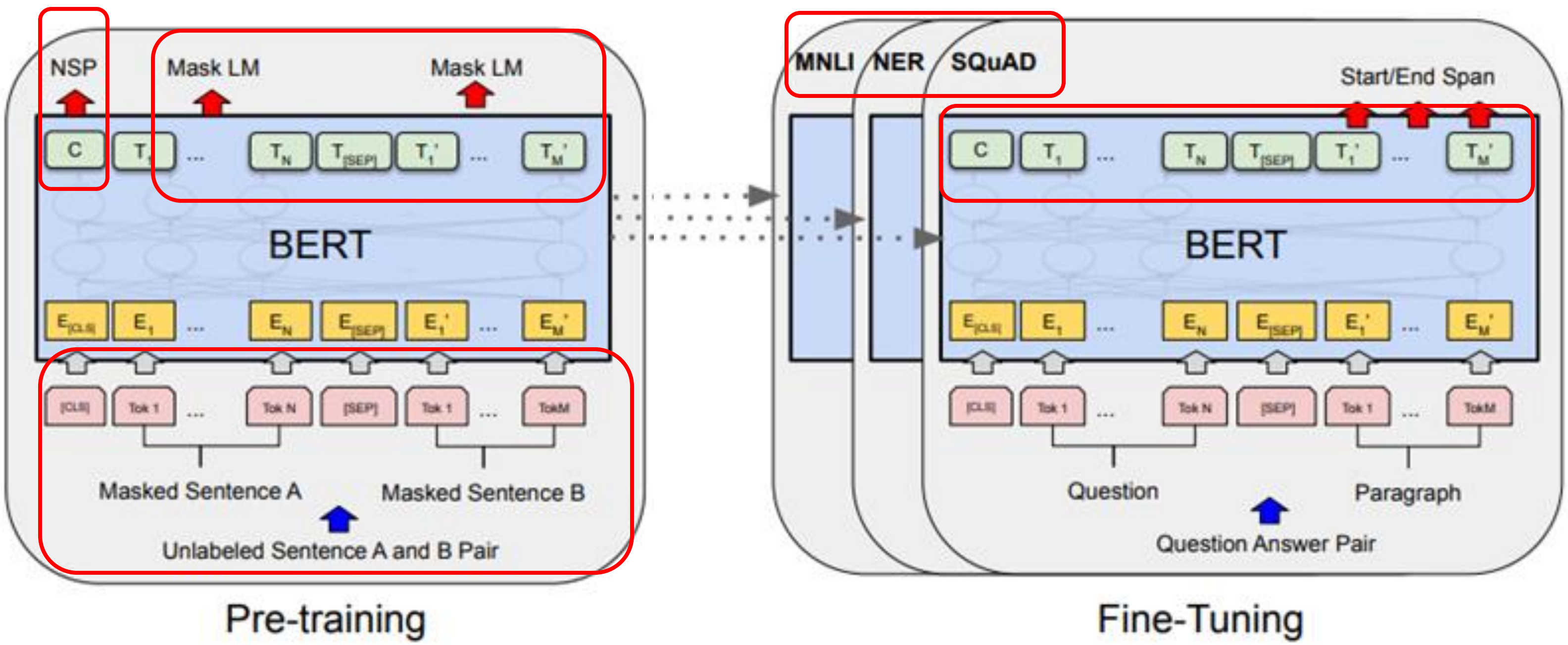
universal-sentence-encoder

Trained with Masked Language Model loss (BERT-like)

Pytorch conversion
https://huggingface.co/dayyass/universal-sentence-encoder-multilingual-large-3-pytorch

BERT-Based embeddings

Sentence representation with BERT

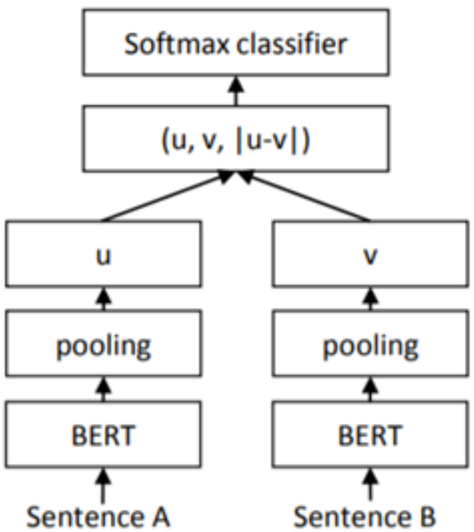


With BERT, we found that MLM training create good sentence representation too!

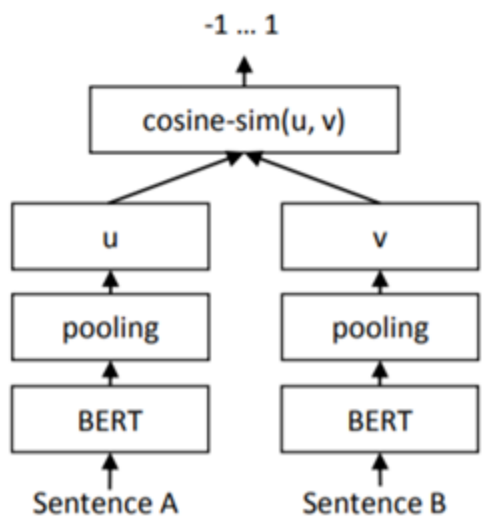
We can use NSP embedding or pool the token embeddings to create a sentence representation

SBERT

Language Understanding



Semantic Understanding



Model	Spearman
Not trained for STS	
Avg. GloVe embeddings	58.02
Avg. BERT embeddings	46.35
InferSent - GloVe	68.03
Universal Sentence Encoder	74.92
SBERT-NLI-base	77.03
SBERT-NLI-large	79.23
Trained on STS benchmark dataset	
BERT-STSb-base	84.30 $\pm$ 0.76
SBERT-STSb-base	84.67 $\pm$ 0.19
SRoBERTa-STSb-base	**84.92** $\pm$ 0.34
BERT-STSb-large	**85.64** $\pm$ 0.81
SBERT-STSb-large	84.45 $\pm$ 0.43
SRoBERTa-STSb-large	85.02 $\pm$ 0.76
Trained on NLI data + STS benchmark data	
BERT-NLI-STSb-base	**88.33** $\pm$ 0.19
SBERT-NLI-STSb-base	85.35 $\pm$ 0.17
SRoBERTa-NLI-STSb-base	84.79 $\pm$ 0.38
BERT-NLI-STSb-large	**88.77** $\pm$ 0.46
SBERT-NLI-STSb-large	86.10 $\pm$ 0.13
SRoBERTa-NLI-STSb-large	86.15 $\pm$ 0.35

https://arxiv.org/abs/1908.10084 2019

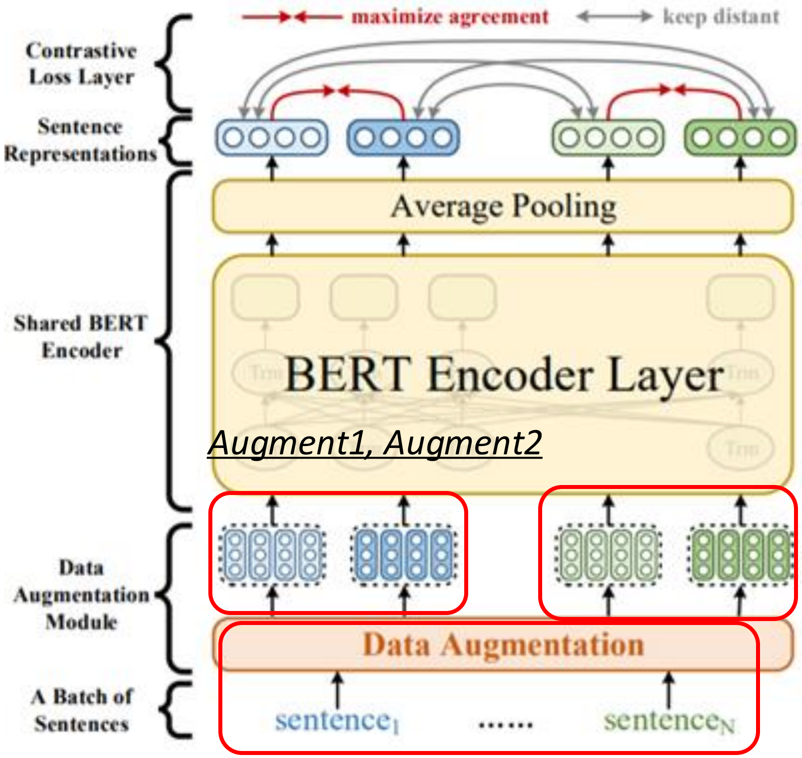
Sentence level contrastive learning

- We can learn better sentence representation with some additional supervised (or unsupervised) sentence level contrastive learning

<u>*Augment1, Augment2*</u>

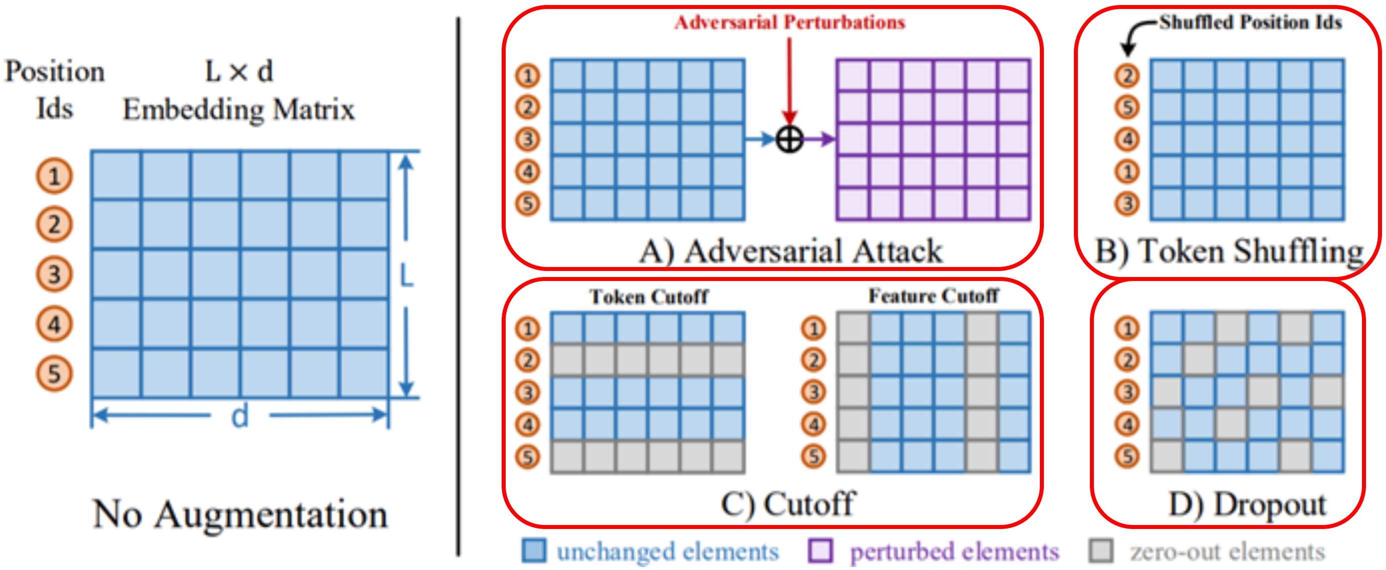
$$\mathcal{L}_{i,j} = -\log \frac{\exp(\mathrm{sim}(r_i, r_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(r_i, r_k)/\tau)}$$

<u>*Not augment2*</u>

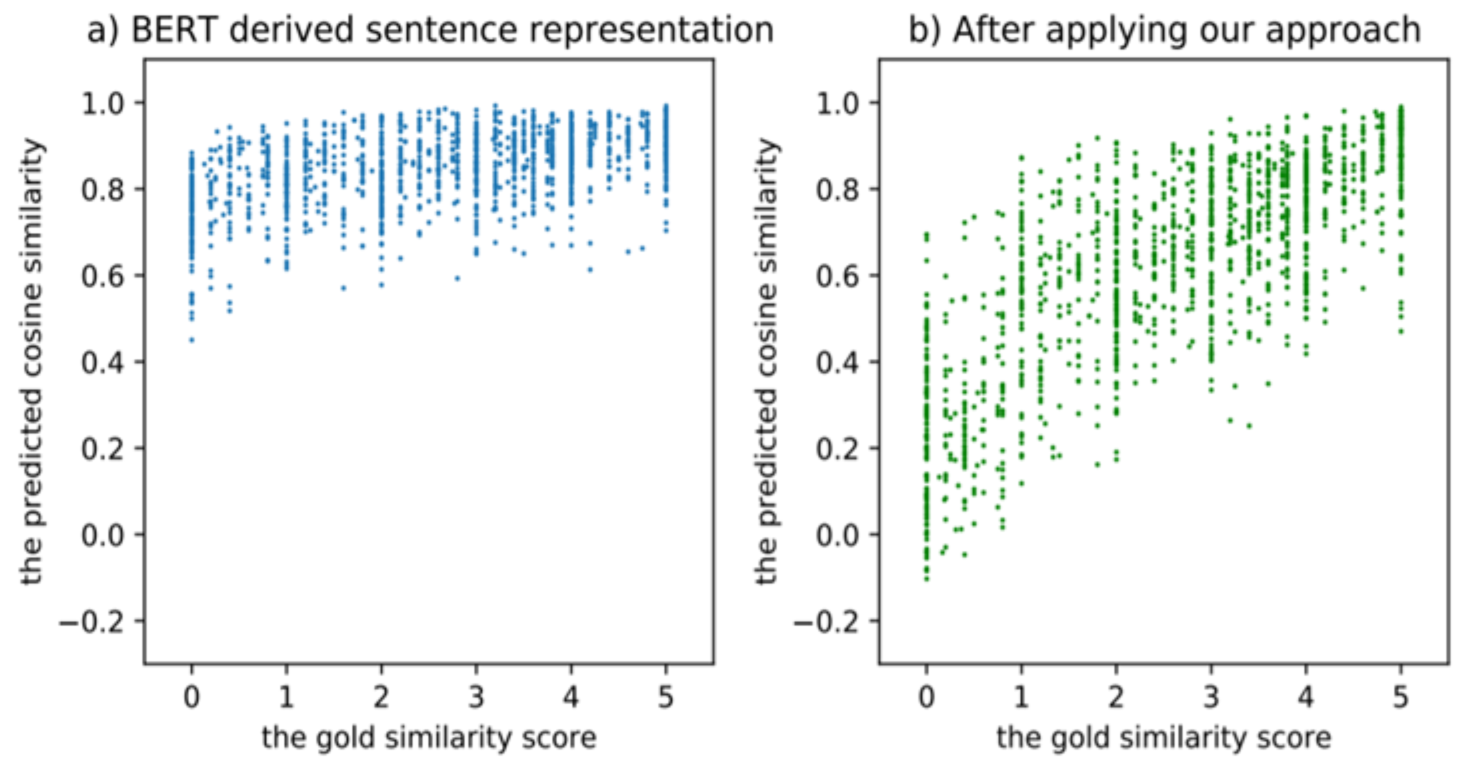


ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer (2021)

ConSERT augmentations

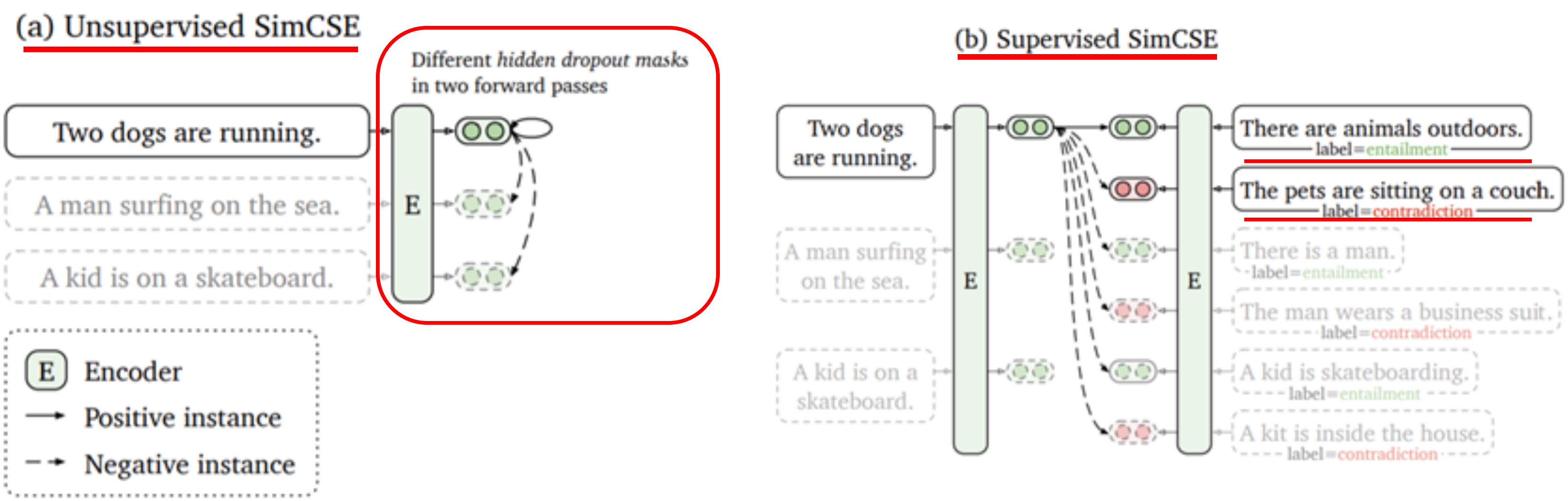


ConSERT alignment



SimCSE

- Use simple dropout in the model to create different versions of the same sentence



SimCSE

Rather than contrastive, predict next sentence, 1 of 3 next sentences

Data augmentation			STS-B
None (unsup. SimCSE)			**82.5**
Crop	*10%*	*20%*	*30%*
	77.8	71.4	63.6
Word deletion	*10%*	*20%*	*30%*
	75.9	72.2	68.2
Delete one word			75.9
w/o dropout			74.2
Synonym replacement			77.4
MLM 15%			62.2

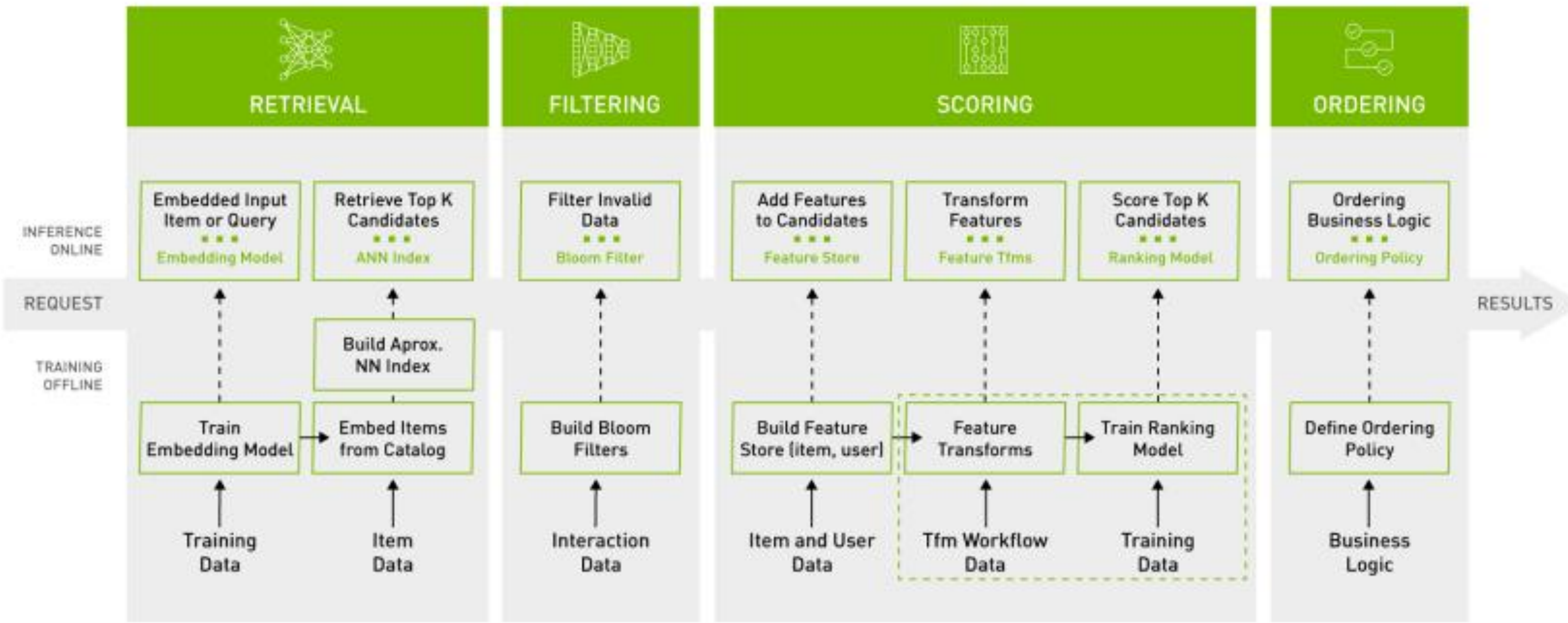
Other augmentations technique

Training objective	f_θ	$(f_{\theta_1}, f_{\theta_2})$
Next sentence	67.1	68.9
Next 3 sentences	67.4	68.8
Delete one word	75.9	73.1
Unsupervised SimCSE	**82.5**	80.7

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\mathrm{sim}(r_i, r_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(r_i, r_k)/\tau)}$$

What’s other use of embeddings?

- Retrieval and recommendation



https://developer.nvidia.com/merlin

What's other use of embeddings?

- Learn joint embeddings between different modalities

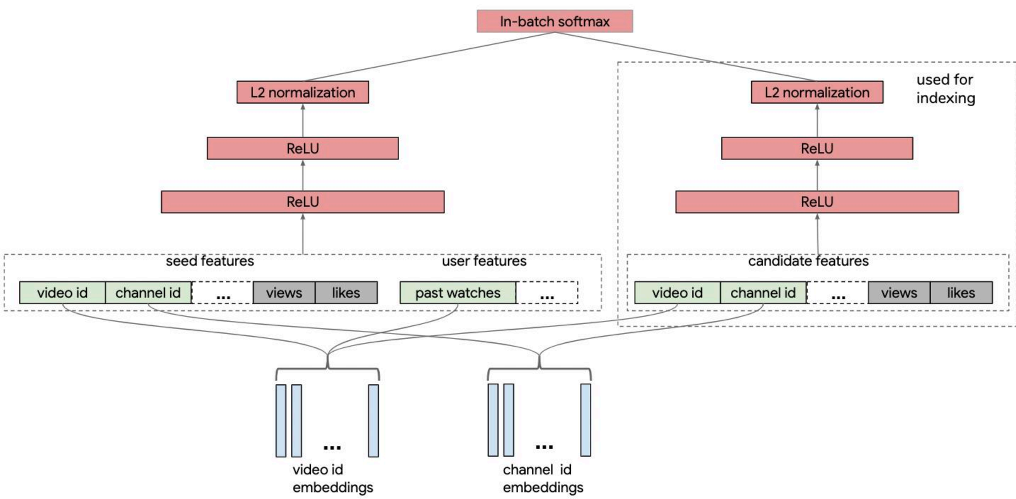
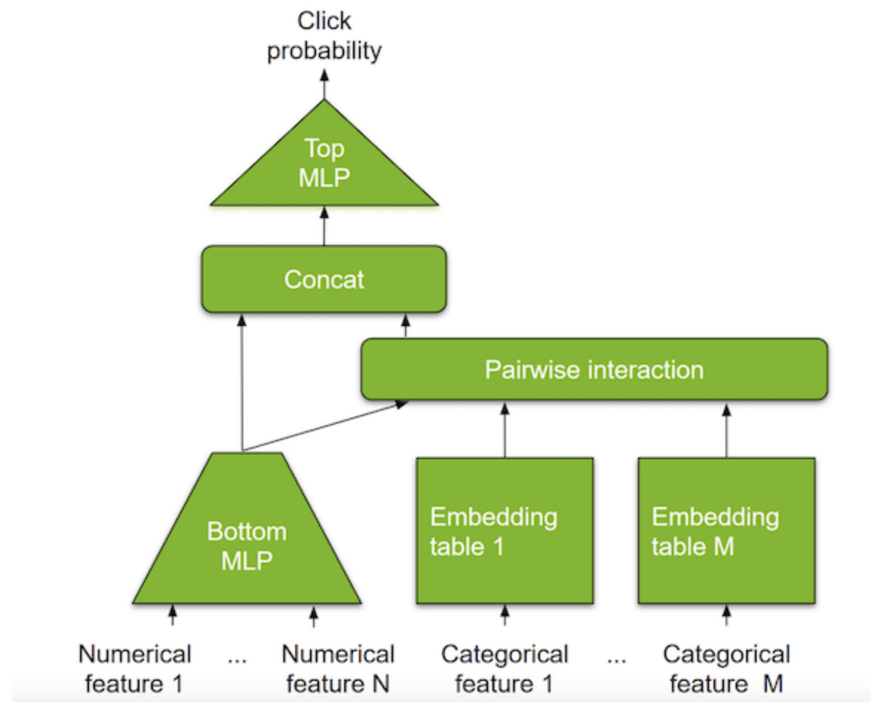


Figure 2: Illustration of the Neural Retrieval Model for YouTube.

Two tower model



Joint interaction model

https://nvidia-merlin.github.io/models/main/models_overview.html#deep-learning-recommender-model

BGE-M3

- A retreival model (Query -> Document)
- Built on top of BGE (Chinese embedding model)
  - BGE: Masked LM finetuned with contrastive and task specific losses
- BGE-M3 (multilingual, multifunction, multigranularity)
- Trained by multiple losses terms that utilizes different parts of the model embeddings

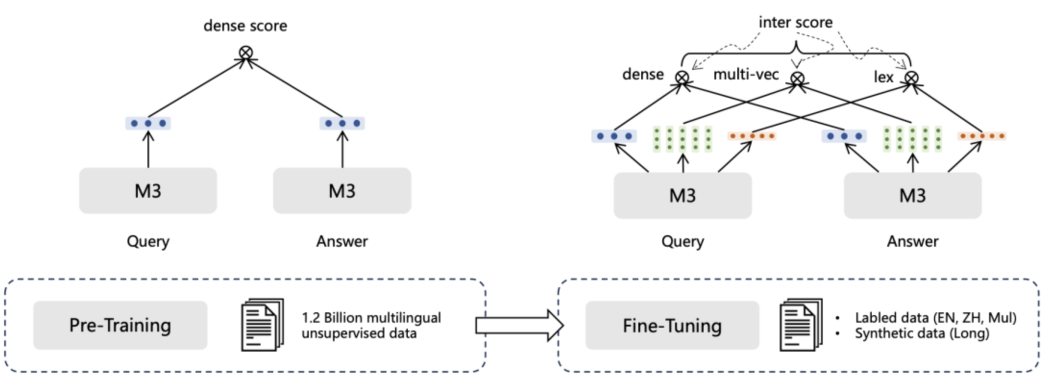
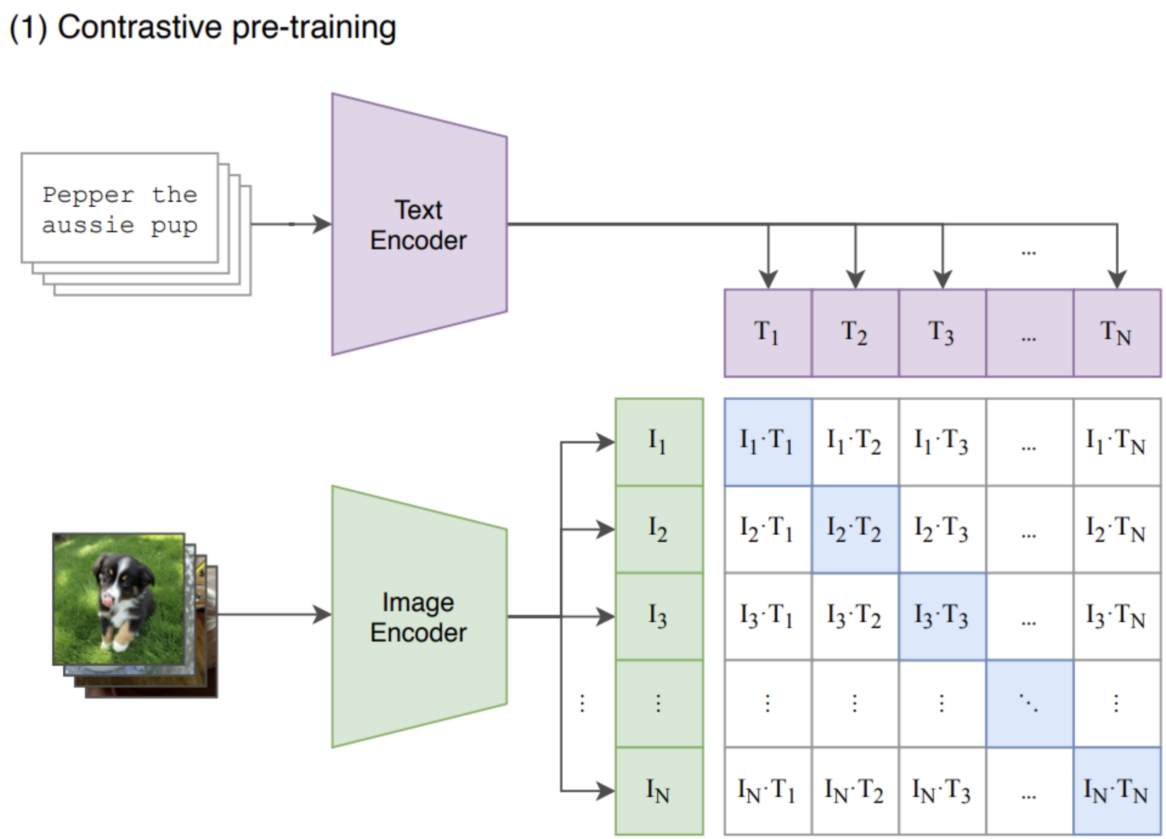


Figure 2: **Multi-stage training process of M3-Embedding with self-knowledge distillation.**

https://arxiv.org/abs/2402.03216

CLIP

- Contrastive learning on image-text pairs



https://arxiv.org/abs/2103.00020

Outline

- Contrastive learning
- Sentence embeddings
    - MUSE
    - SimCSE
    - BGE
    - CLIP