Representation learning

Self-supervised learning

Self-supervised learning

Unsupervised learning trained using supervised learning techniques

Cleverly exploit property of the data to create pseudo labels

Mostly used for representation learning

Need small supervised data to map to useful task

Yann LeCun
April 30, 2019 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.

Self-supervised learning has been enormously successful in natural language processing. For example, the BERT model and similar techniques produce excellent representations of text.
BERT is a prototypical example of self-supervised learning: show it a sequence of words on input, mask out 15% of the words, and ask the system to predict the missing words (or a distribution of words). This an example of masked auto-encoder, itself a special case of denoising auto-encoder, itself an example of self-supervised learning based on reconstruction or prediction. But text is a discrete space in which probability distributions are easy to represent.

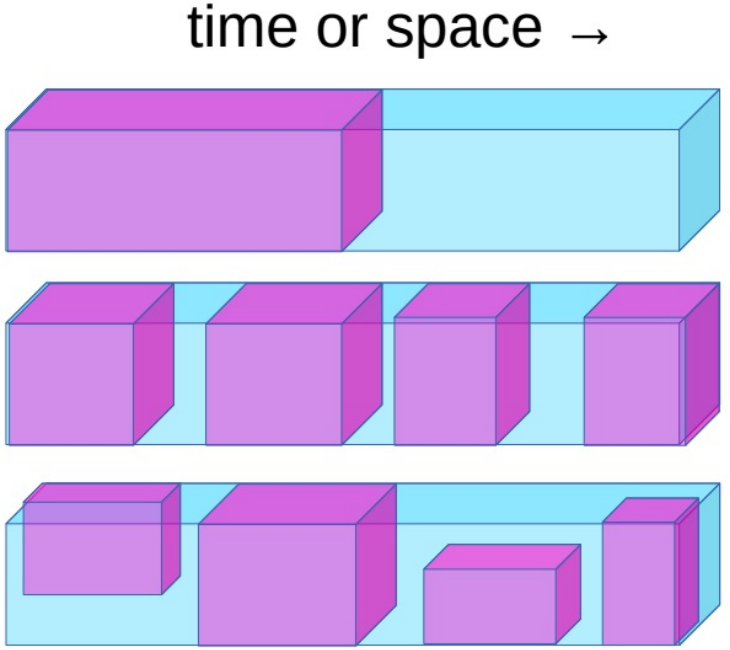
So far, similar approaches haven't worked quite as well for images or videos because of the difficulty of representing distributions over high-dimensional continuous spaces.

Doing this properly and reliably is the greatest challenge in ML and AI of the next few years in my opinion.

Self-Supervised Learning = Filling in the Blanks

Y. LeCun

- **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

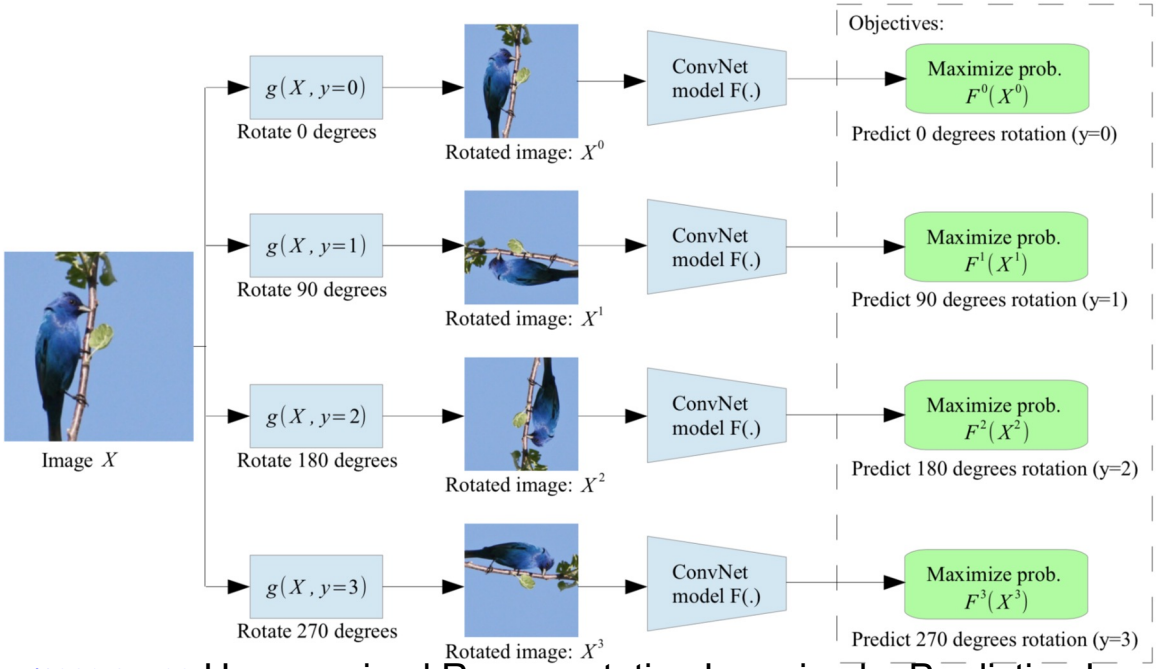
Examples

Text

Predict masked text - BERT

Images

Predict missing patches, predict orientation, etc.



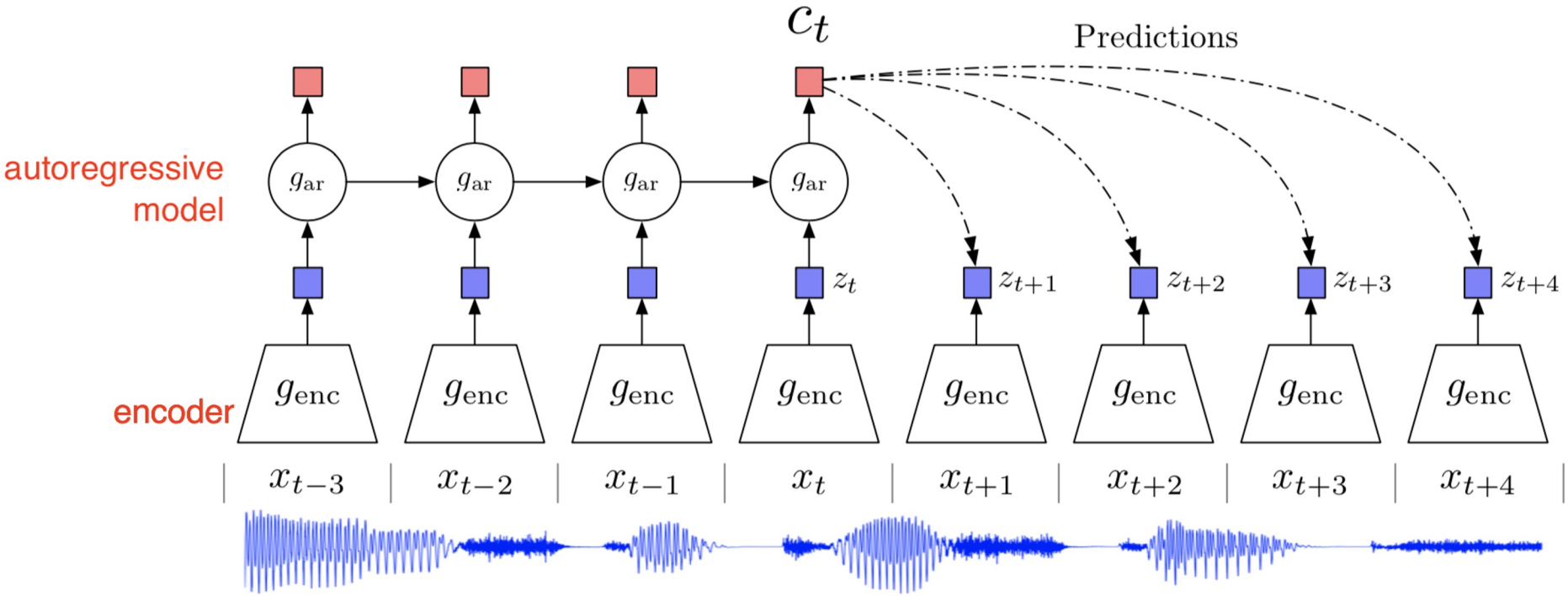
https://arxiv.org/abs/1803.07728 Unsupervised Representation Learning by Predicting Image Rotations

Examples

Sound

Cluster data into discrete classes then predict masked portions

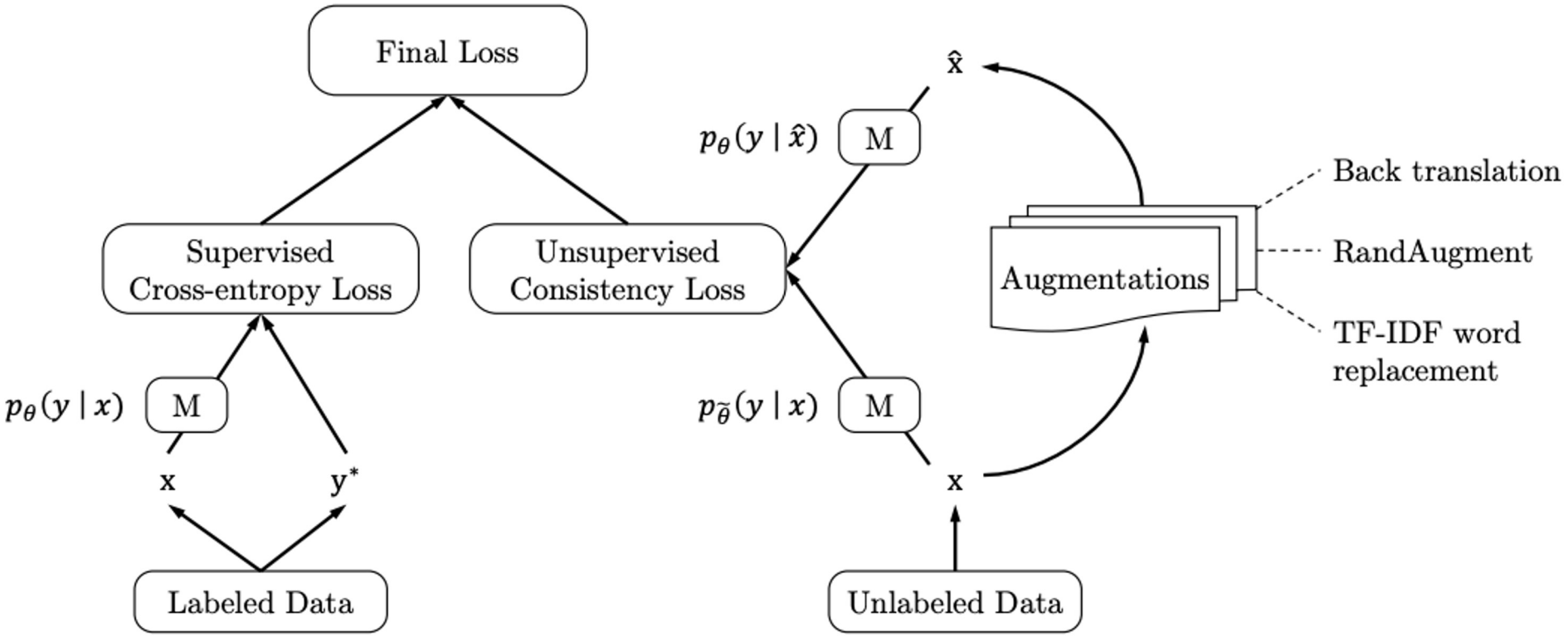
More examples here https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html



https://arxiv.org/abs/1807.03748 Representation Learning with Contrastive Predictive Coding

Consistency training

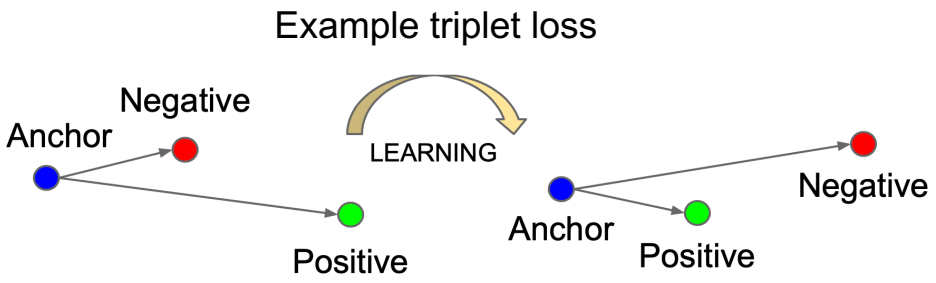
- Consistency loss can be considered as a self-supervised loss



https://arxiv.org/abs/1904.12848 Unsupervised Data Augmentation for Consistency Training

Contrastive training

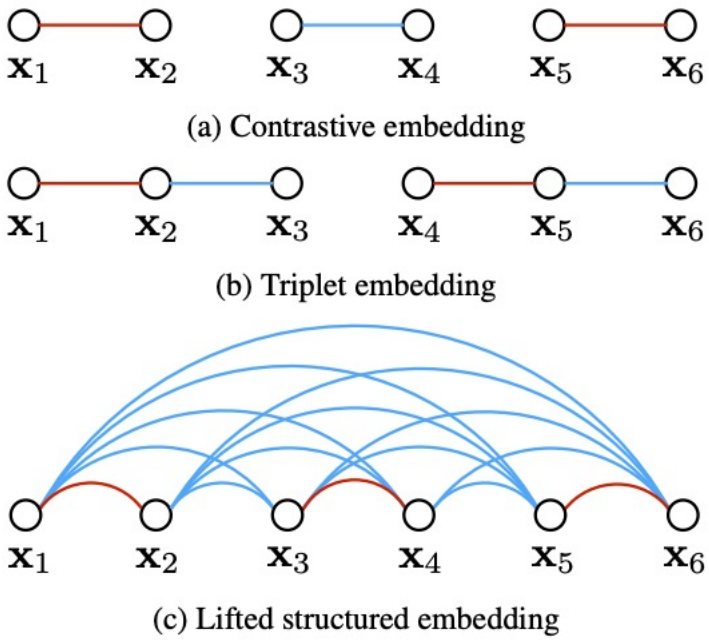
- Consistency training focus on pulling similar things together while ignoring noise
- Contrastive training focus on pushing different things away
- Contrastive loss are key in face verficiation task
- These two are often times used together, and the clear differentiation between the two are vague



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
  - LogP(data) – Log P(noise or negative samples)
  - This is pretty much negative sampling!
  - 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
- Given a context vector c, the positive x should be selected rather than the negative x
- Effectively maximize mutual information between c and positive x

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

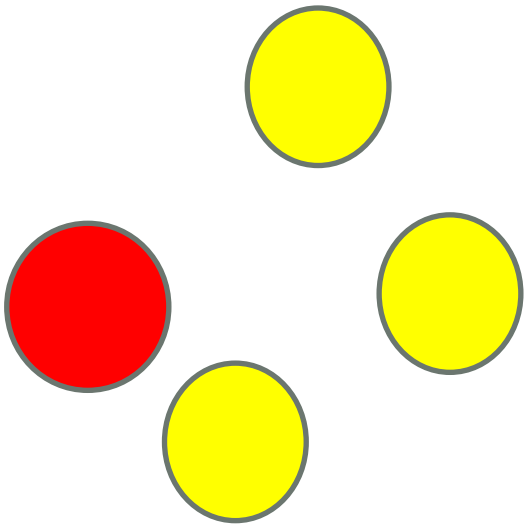
$$f(x,c) = exp(\mathbf{z}^T W c)$$

z is encoded x

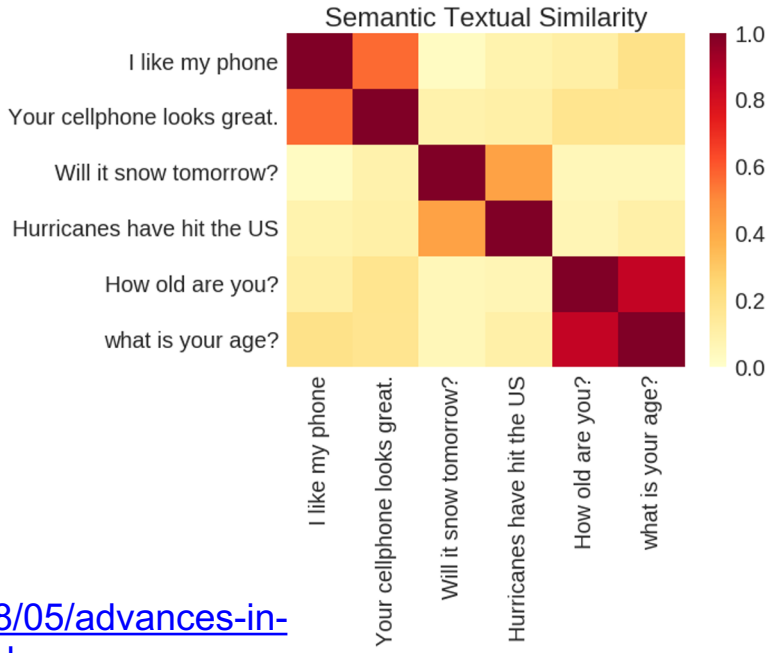
- f( ) can be any function that describes similarity
- Can be extended to have multiple positive examples in a batch (soft nearest neighbor loss) https://arxiv.org/abs/1902.01889

Key components to make constrastive learning work

- Large batch, large tranining data
- Hard/semi-hard negative mining
- Augmentation on the anchor and postive (consistency training)
- Other improvement includes - adding classification loss (CE/softmax loss)



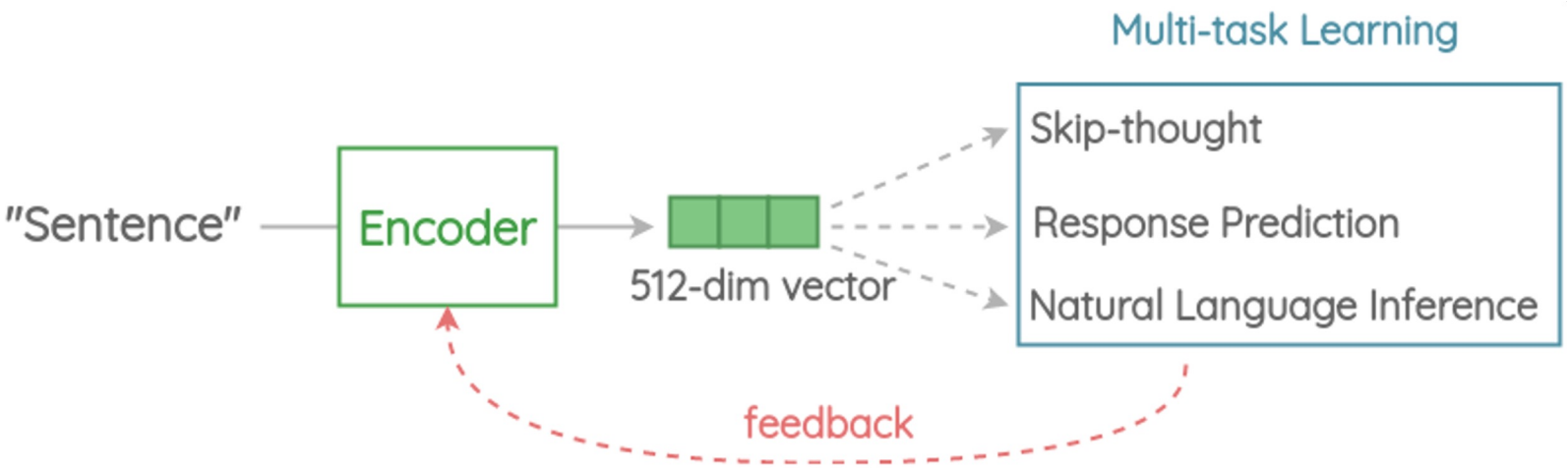
Sentence embeddings



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

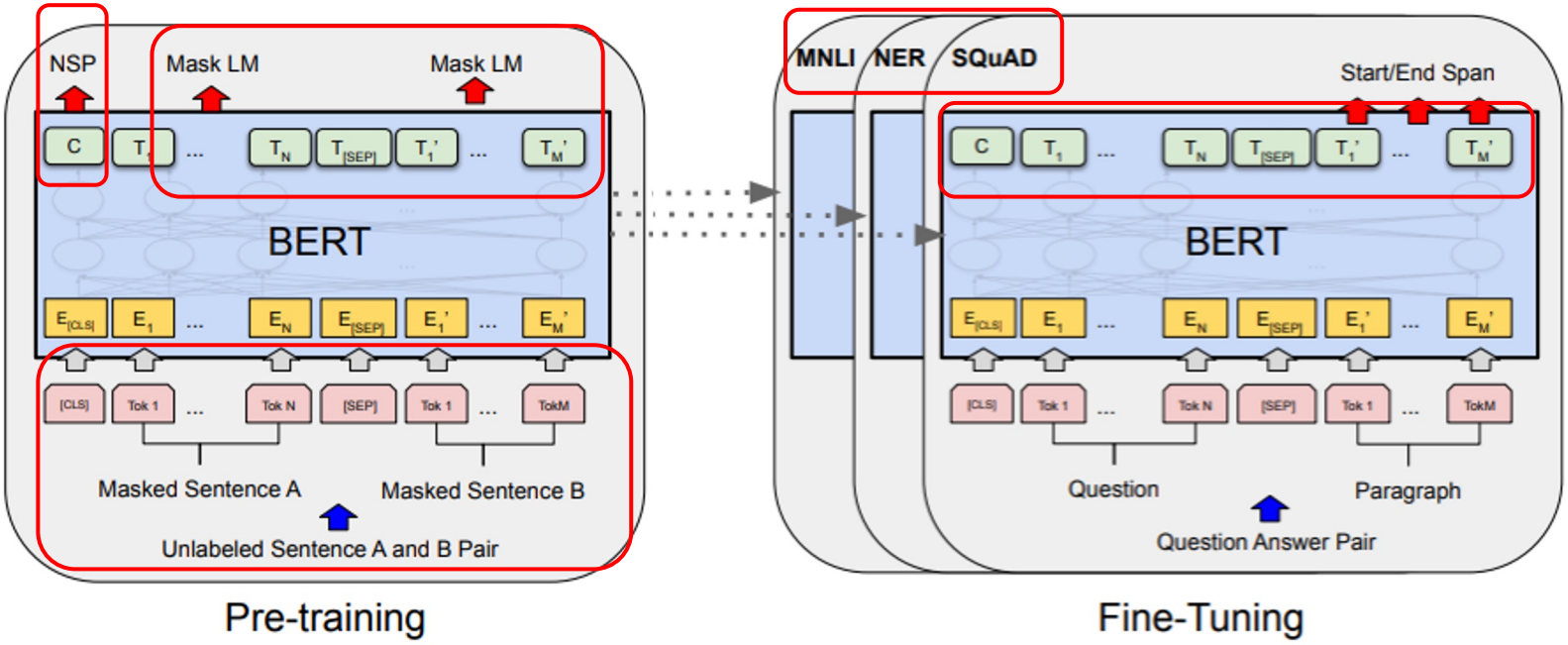
Sentence embeddings

- Early sentence representations are learned from skip-thought or other sentence-level tasks in a multi-task manner



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

Sentence representation with BERT

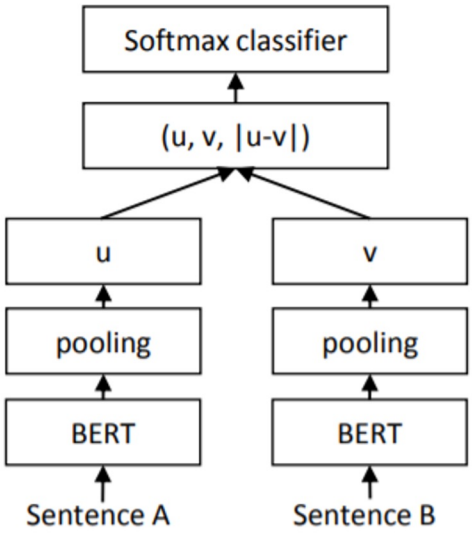


With BERT, we found that MLM training create good sentence representation too!

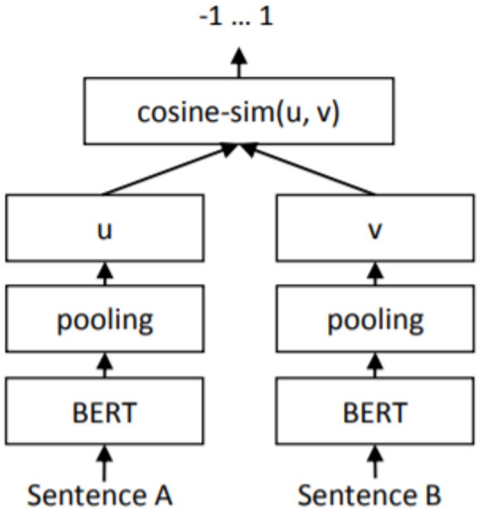
We can use NSP embedding or pool the token embeddings to create a sentence representation

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 ± 0.76
SBERT-STSb-base	84.67 ± 0.19
SRoBERTa-STSb-base	**84.92** ± 0.34
BERT-STSb-large	**85.64** ± 0.81
SBERT-STSb-large	84.45 ± 0.43
SRoBERTa-STSb-large	85.02 ± 0.76
Trained on NLI data + STS benchmark data	
BERT-NLI-STSb-base	**88.33** ± 0.19
SBERT-NLI-STSb-base	85.35 ± 0.17
SRoBERTa-NLI-STSb-base	84.79 ± 0.38
BERT-NLI-STSb-large	**88.77** ± 0.46
SBERT-NLI-STSb-large	86.10 ± 0.13
SRoBERTa-NLI-STSb-large	86.15 ± 0.35

https://arxiv.org/abs/1908.10084

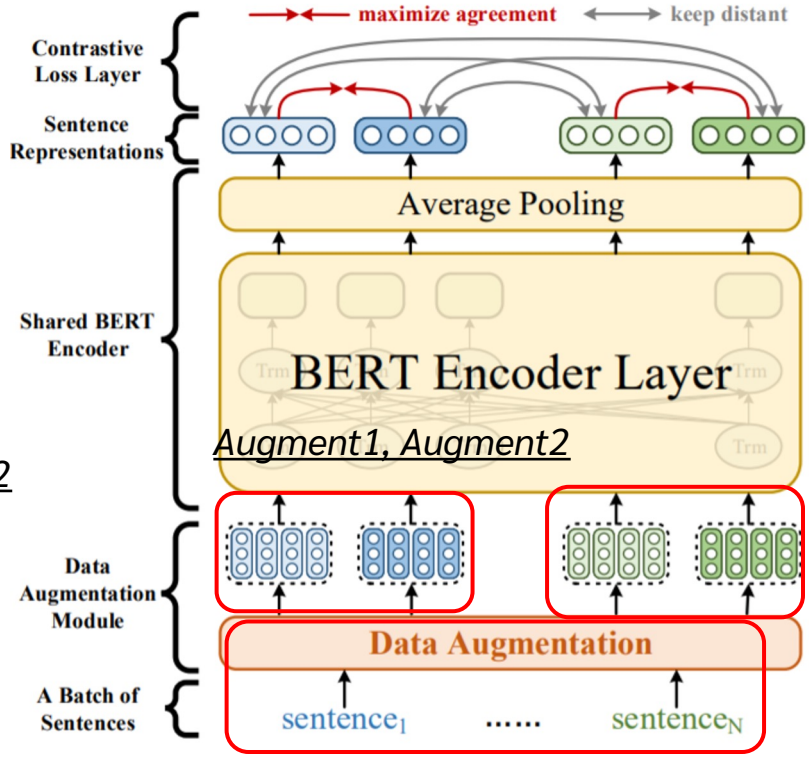
Sentence level contrastive learning

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

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

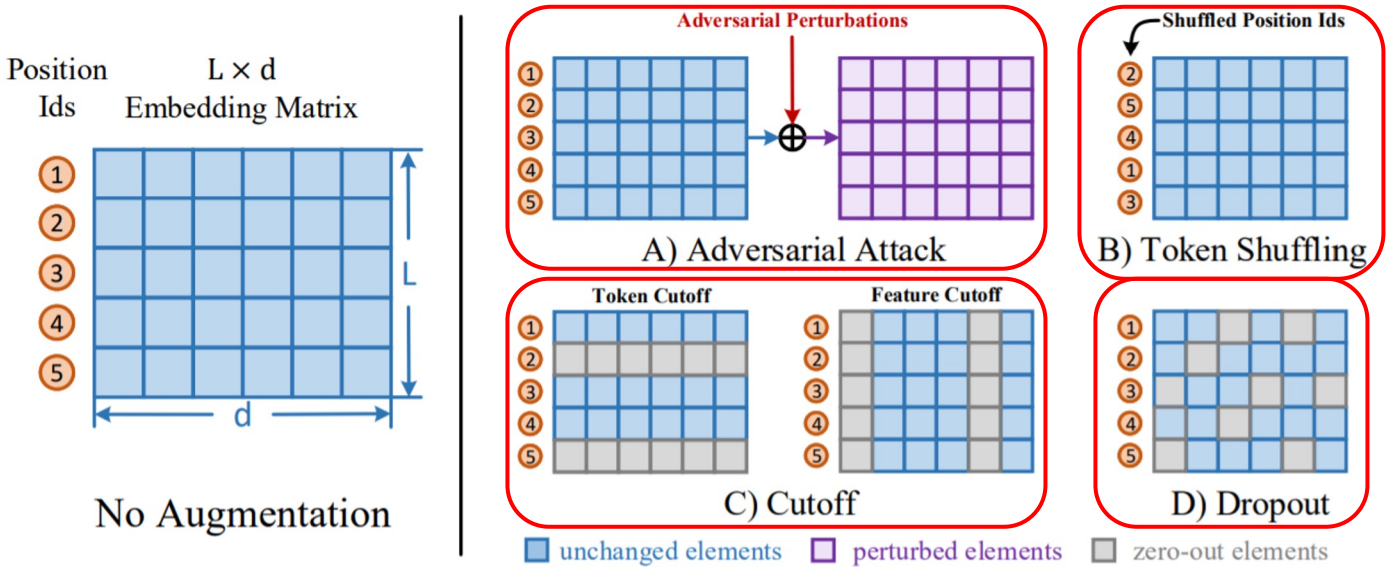
Augment1, Augment2

Not augment2

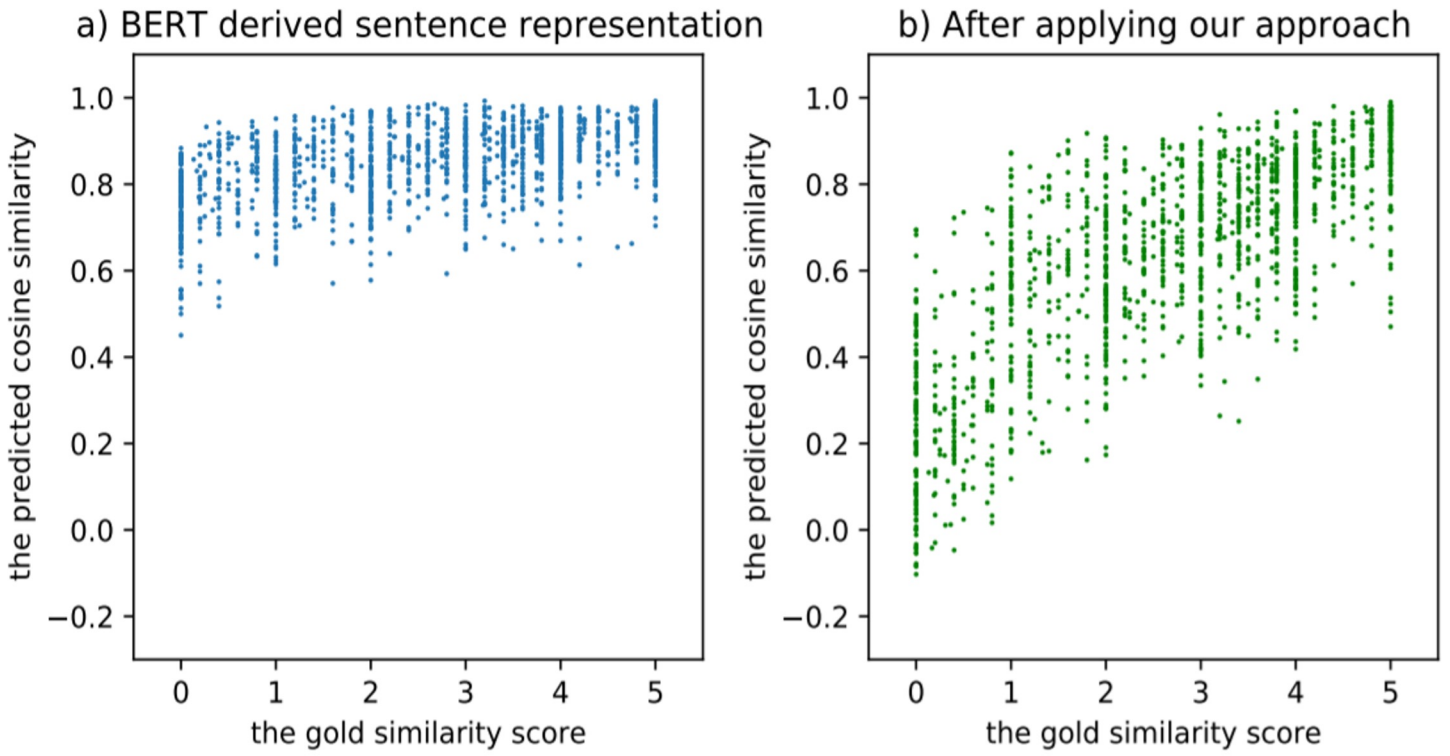


ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer (2021)

ConSERT augmentations

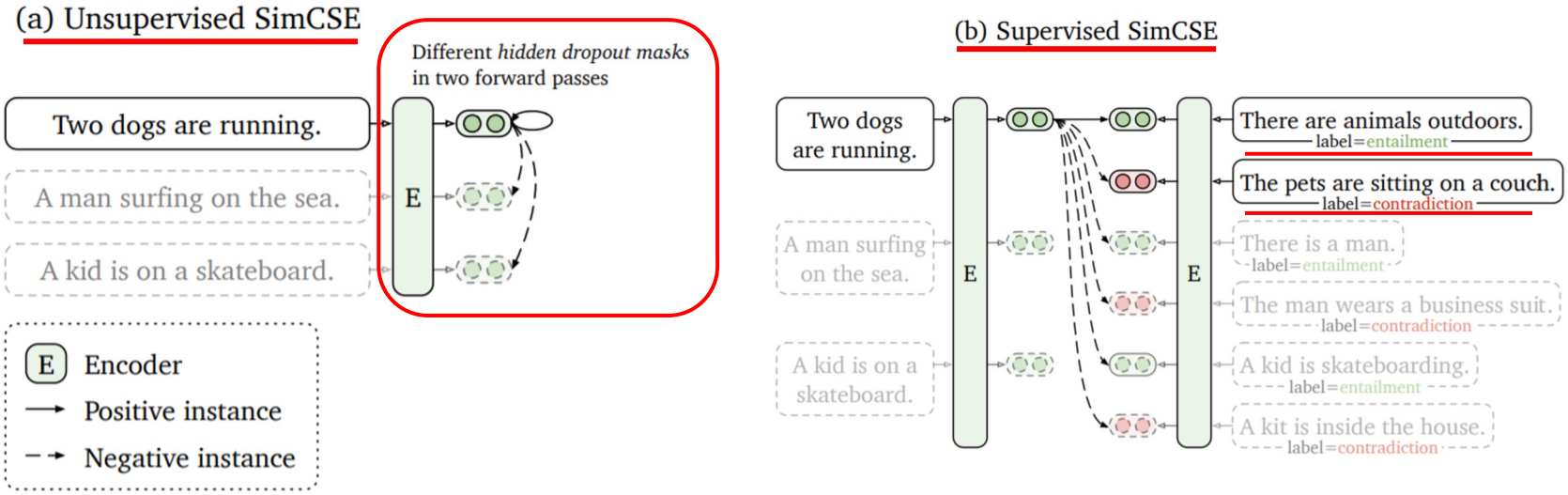


ConSERT alignment



SimCSE

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



SimCSE

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

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

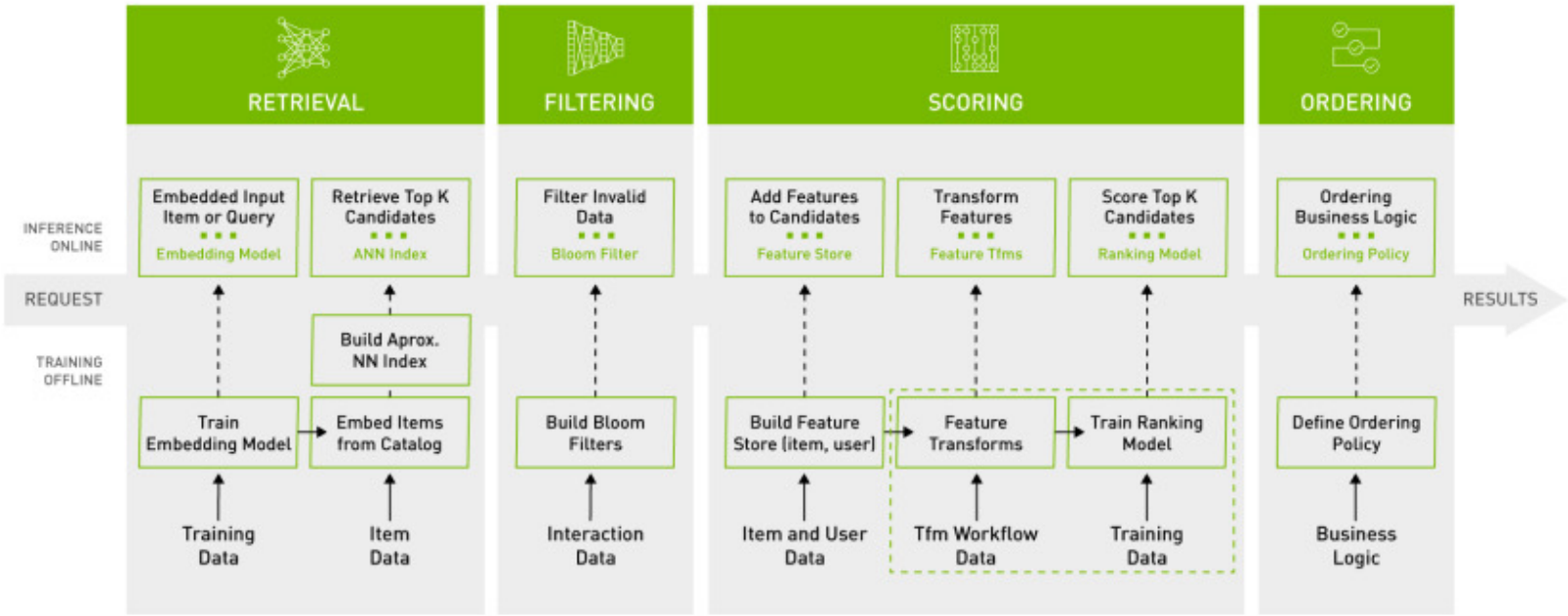
Rather than contrastive, predict next sentence, 1 of 3 next sentences

Some demo on self-supervised techniques

- https://drive.google.com/file/d/1T-IJI_VA49xf8w9nPcqJrftoaoa-R-fQ/view?usp=sharing

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

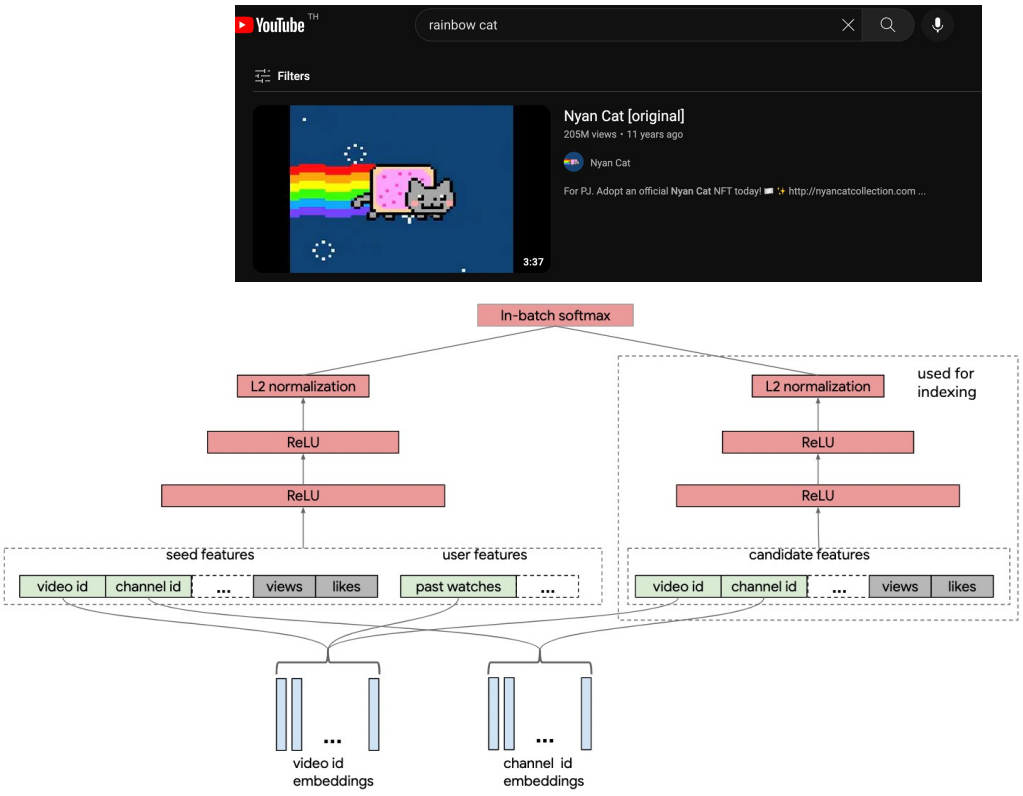
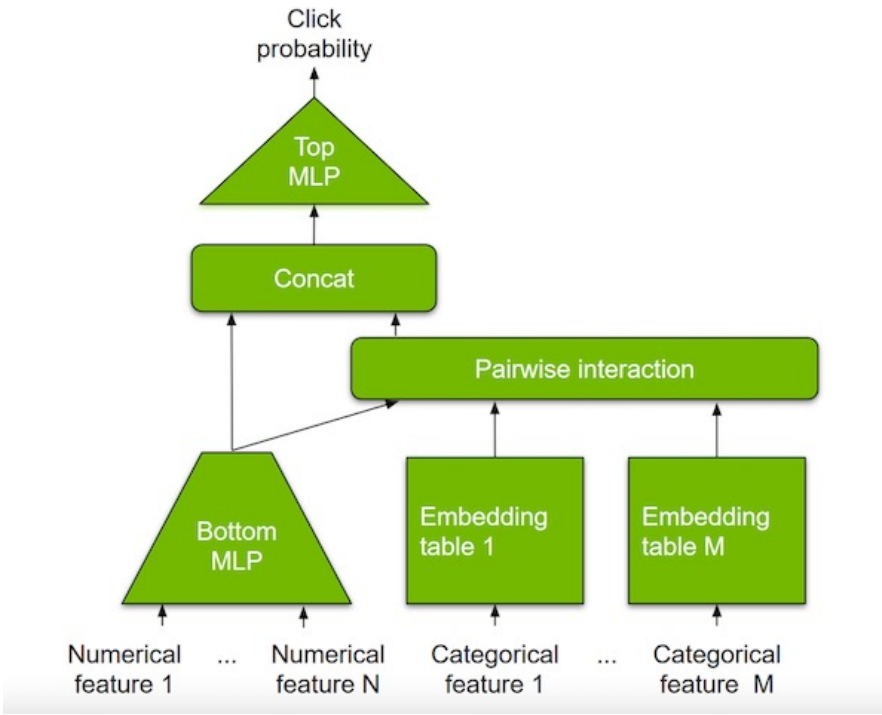


Figure 2: Illustration of the Neural Retrieval Model for YouTube.

Two tower model



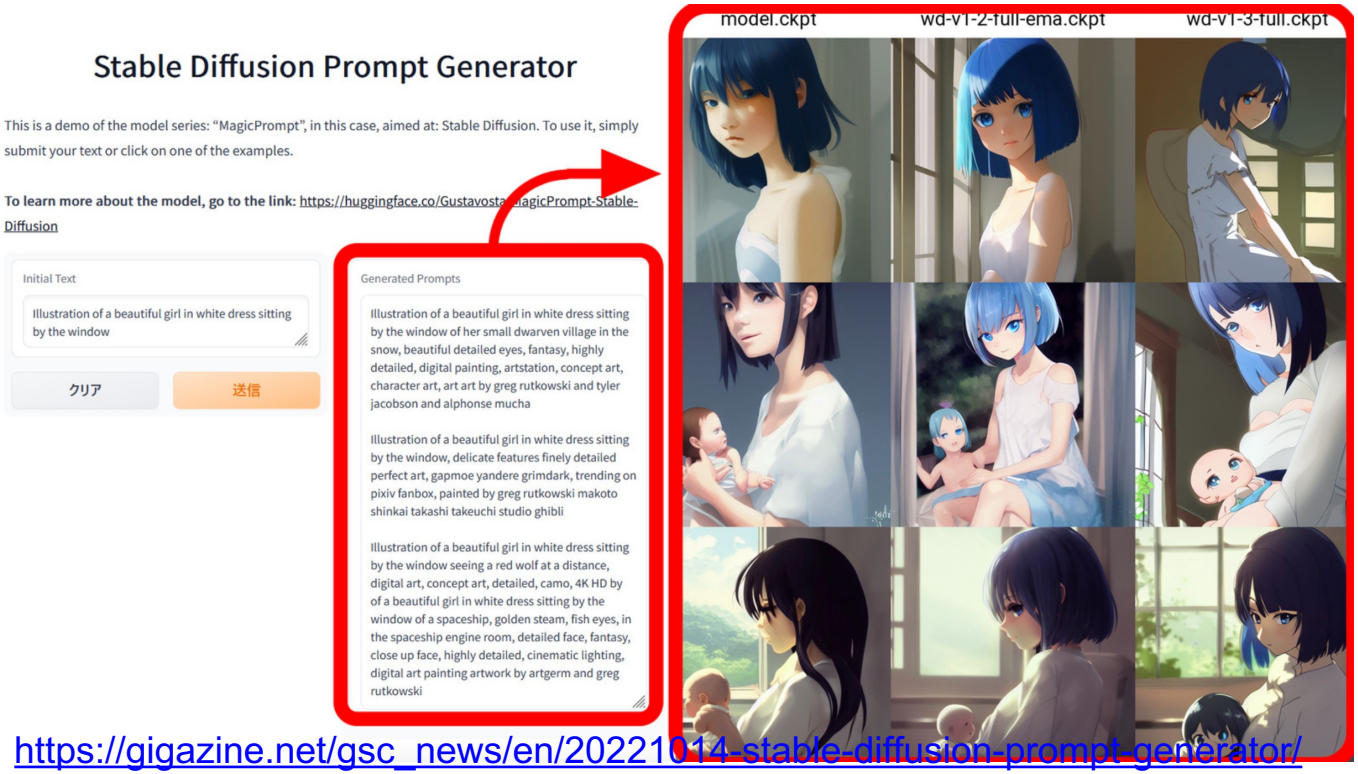
Joint interaction model

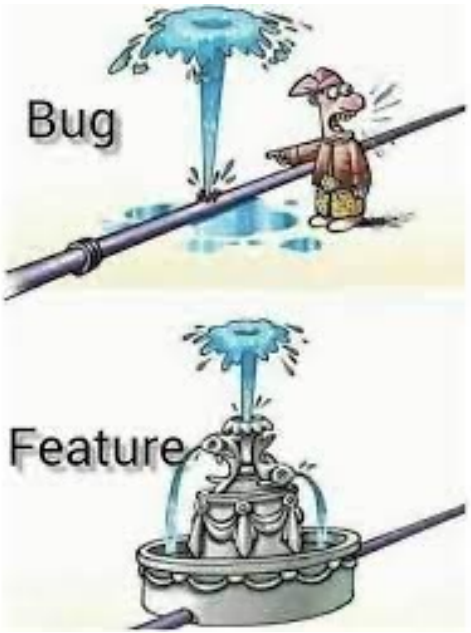
https://nvidia-merlin.github.io/models/main/models_overview.html#deep-learning-recommender-model

Let’s discuss self-supervised and chatGPT

Zero-shot capabilities from self-supervised models

- GPT3 has zero-shot, few-shot capabilities.
  - But you need to learn how to interact
    - Prompt engineering
    - Figure out a way to tease out the answers





It's a bug not a feature!

https://gigazine.net/gsc_news/en/20221014-stable-diffusion-prompt-generator/

Why prompting?

https://www.alignmentforum.org/
Web forum discussing the alignment problem

- It's hard to make a self-superivsed model do what we want. This is called the alignment problem (aligns model capabilities with users' interests). Ultimately, you will NEED supervision for this!
- However, it's unweildy to finetune a large model
  - Takes training and data collection time (effects time to market)
  - Catastophically forgets
  - Compute (RAM and FLOP limitations)
- Current solutions
  - Prompt engineer (manual)
  - Low parameter finetuning (see papers that your friends will present ☺) – LoRa, mixture of experts, learned prompts, etc.
  - Reinforcement Learning with Human Feedback (RLHF)

RLHF

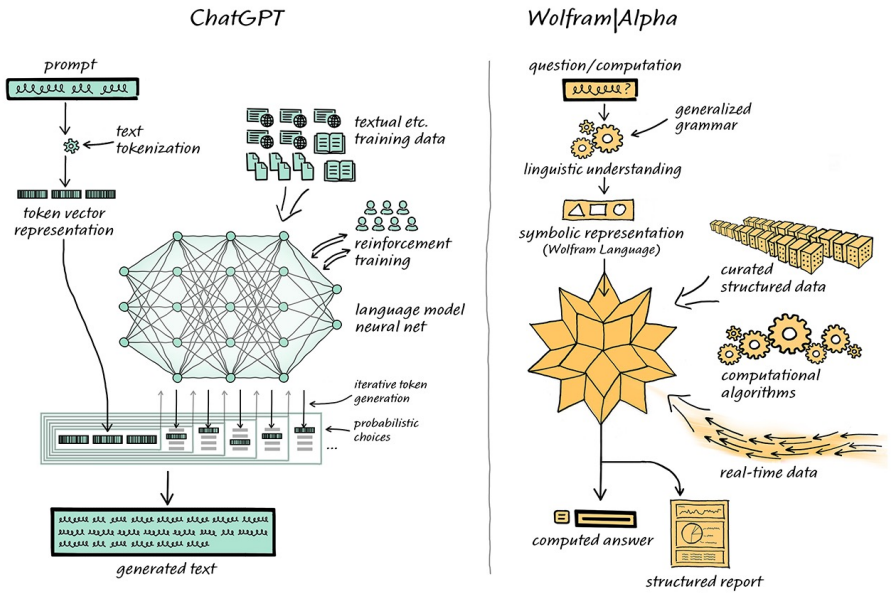
- Imitation learning
  - A term in the RL community for learning techniques from demonstrations
  - E.g. Supervised learning
  - Downside– imitation data is usually limited
    - Overfits, cannot generalize
- Reinformance learning
  - Learn from the reward signal
    - Hand-crafted rewards
    - Learned rewards
  - Has better generalization
  - Downside- more data hungry due to less supervised signal
- However, people have found some success with using only finetuning on ChatGPT-like models

ChatGPT++

- People have been incorporating ChatGPT with external tools
  - Wolfram Alpha

Wolfram|Alpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT

January 9, 2023



https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/

ChatGPT++

- People have been incorporating ChatGPT with external tools
  - AzureAPIs

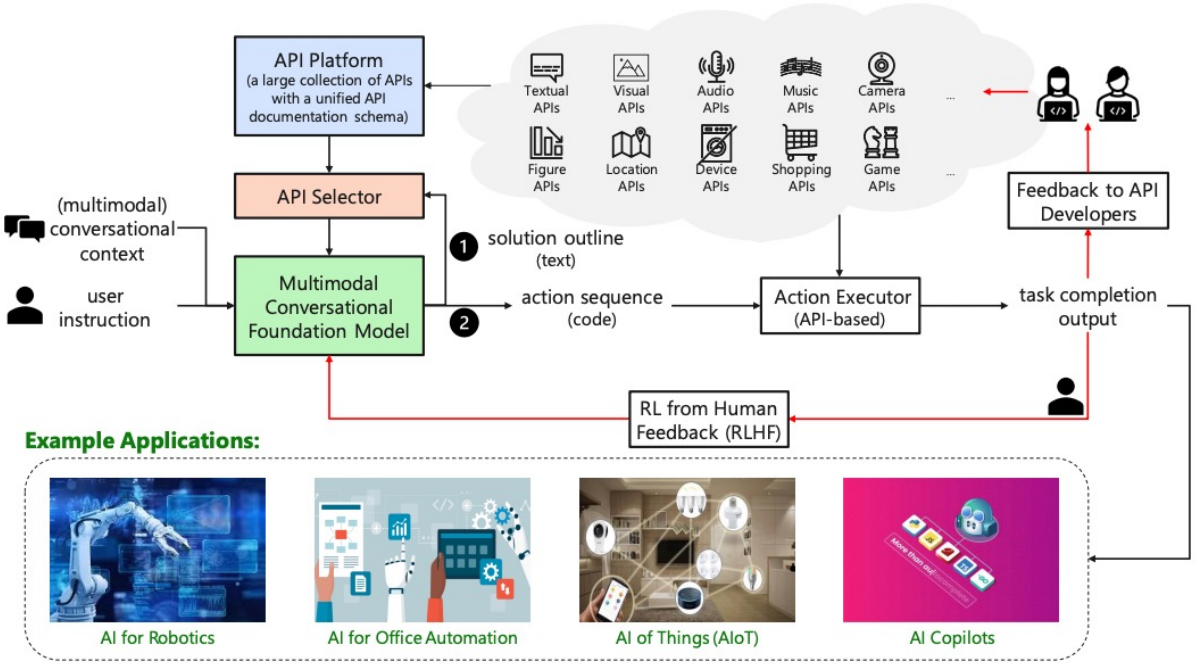
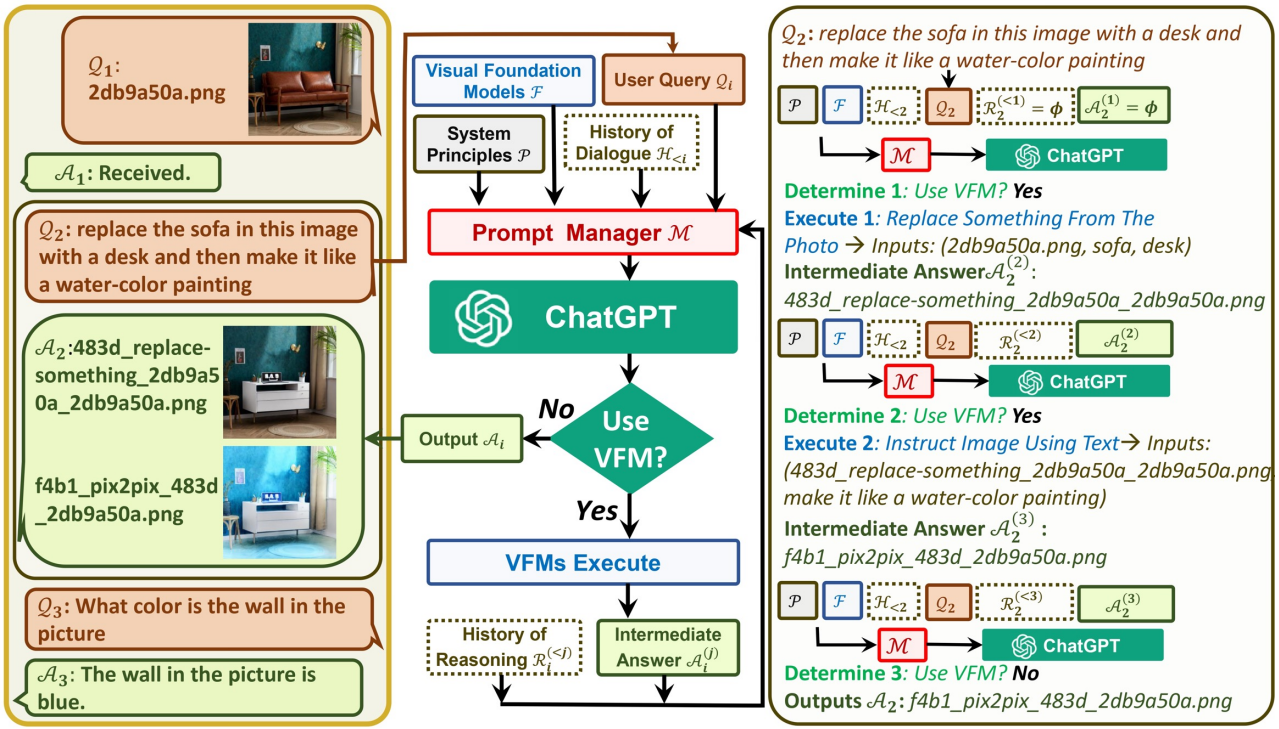


Figure 1: Overview of TaskMatrix.AI. Given user instruction and the conversational context, the multimodal conversational foundation model (MCFM) first generates a solution outline (step ①), which is a textual description of the steps needed to solve the task. Then, the API selector chooses the most relevant APIs from the API platform according to the solution outline (step ②). Next, MCFM generates action codes using the recommended APIs, which will be further executed by calling APIs. Last, the user feedback on task completion is returned to MCFM and API developers.

https://arxiv.org/abs/2303.16434

ChatGPT++

- People have been incorporating ChatGPT with external tools
  - Visual models



https://github.com/microsoft/visual-chatgpt