Recognizing Multimodal Entailment

ACL 2021

Recognizing Multimodal Entailment

Available at multimodal-entailment.github.io

BROWN

DeepMind

UNIVERSITY of WASHINGTON

Google Research

UWNLP

IST AUSTRIA
Institute of Science and Technology

Presenters

Afsaneh Shirazi,
Google

Arjun Gopalan,
Google Research

Arsha Nagrani,
Google Research

Cesar Ilharco,
Google

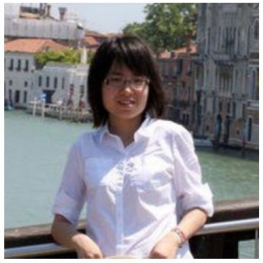
Christina Liu,
Google Research

Gabriel Barcik,
Google Research

Jannis Bulian,
Google Research

Jared Frank,
Google

Lucas Smaira,
DeepMind



Qin Cao,
Google Research

Ricardo Marino,
Google

Roma Patel,
Brown University

Acknowledgements

We would like to thank Abby Schantz, Abe Ittycheriah, Alex Ku, Aliaksei Severyn, Allan Heydon, Aly Grealish, Andrey Vlasov, Arkaitz Zubiaga, Ashwin Kakarla, Blaž Bratanič, Chen Sun, Chris Bregler, Clayton Williams, Cong Yu, Cordelia Schmid, Da-Cheng Juan, Dan Finnie, Dani Valevski, Daniel Rocha, David Chiang, David Price, David Sklar, Devi Krishna, Elena Kochkina, Elizabeth Hamon Reid, Enrique Alfonseca, Felipe Ferreira, Françoise Beaufays, Gabriel Ilharco, Georg Osang, Isabel Kraus-Liang, Isabelle Augenstein, Iulia Turc, Jacob Eisenstein, Jialu Liu, John Cantwell, John Palowitch, Jordan Boyd-Graber, Kenton Lee, Lei Shi, Luís Valente, Maria Voitovich, Mehmet Aktuna, Min Zhang, Mogan Brown, Mohammad Khan, Mor Naaman, Natalia P, Nidhi Hebbar, Pandu Nayak, Pete Aykroyd, Rahul Sukthankar, Richa Dixit, Sara Goetz, Sol Rosenberg, Steve Pucci, Tania Bedrax-Weiss, Thomas Leung, Tim Dettmers, Tobias Kaufmann, Tom Boulos, Tu Tsao, Vaiva Imbrasaite, Vladimir Chtchetkine, Yair Kurzion, Yifan Xu and Zach Hynes.

Agenda

01 Introduction

02 Natural Language Processing

03 Structured Data

04 Neural Graph Learning

05 Computer Vision

06 Multimodal Learning

07 Multimodal entailment

08 Closing Notes and Q&A

01

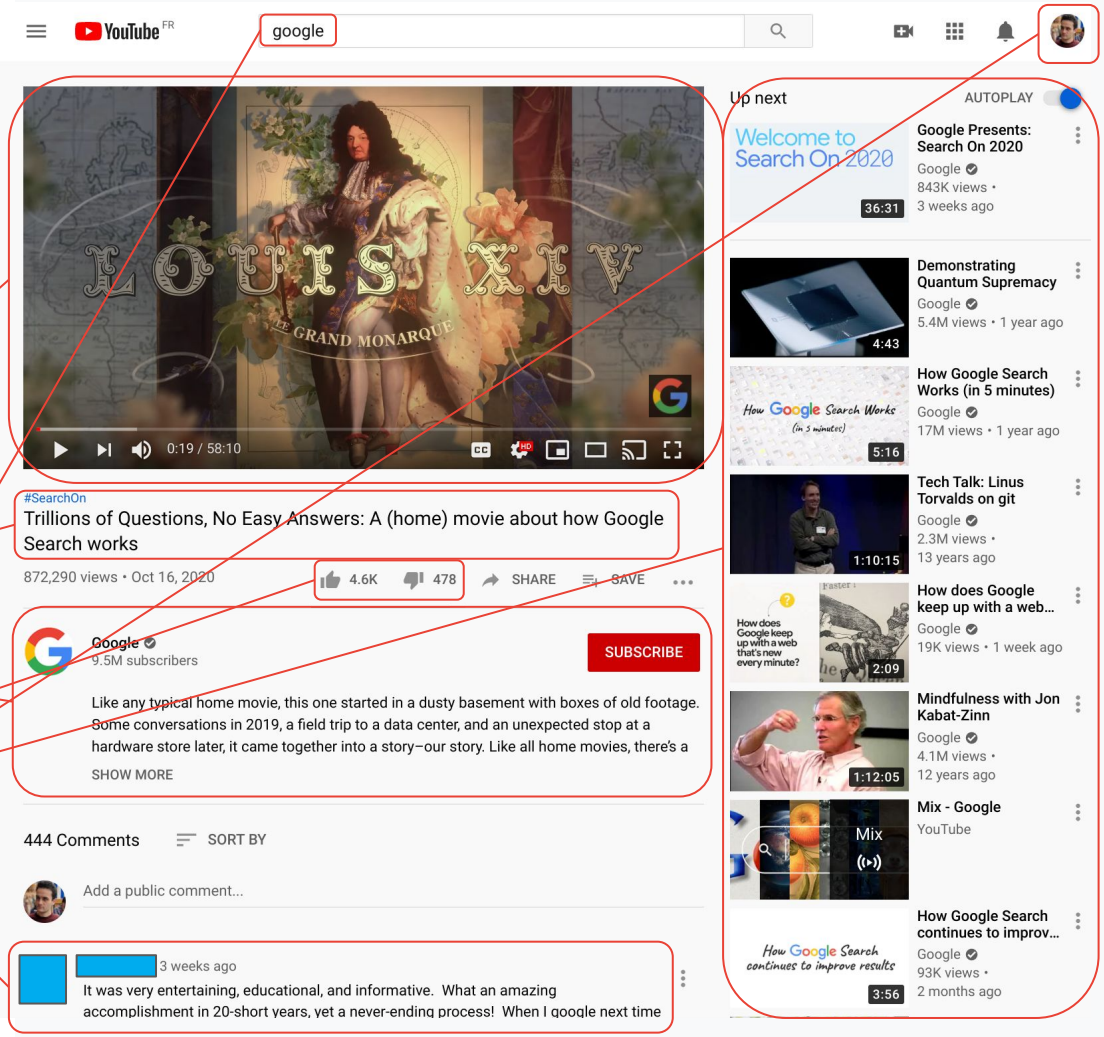
Introduction

The landscape of online content

The shapes of content

Diverse sources of content on this page.

- Screen
  - Audio
  - Video
- Title
- Description
- Channel
- Comments
- Likes
- Watch Next
- User
- Query



The shapes of content

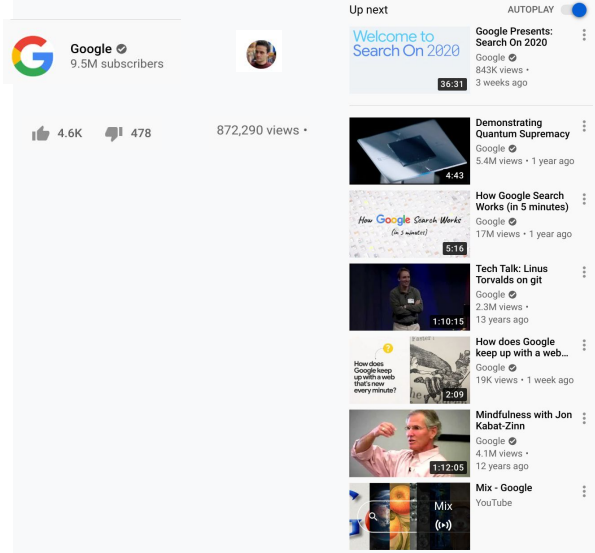
Classifying information sources

How can we reconcile all these methods and information sources?

Structured (Tabular)

Classical machine learning strategies

- Trees, Boosting
- Regressions
- SVM
- Graphs



Non-Structured

Deep Learning

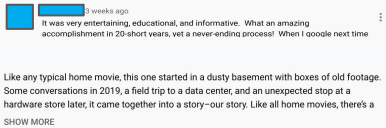
Image

- CNN

Sound

- RNN
- TCN

Text (Natural Language Processing)



- Transformer

A case for multimodal entailment inferences

Multimodal entailment

No single information channel is enough to determine agreement between content sources.

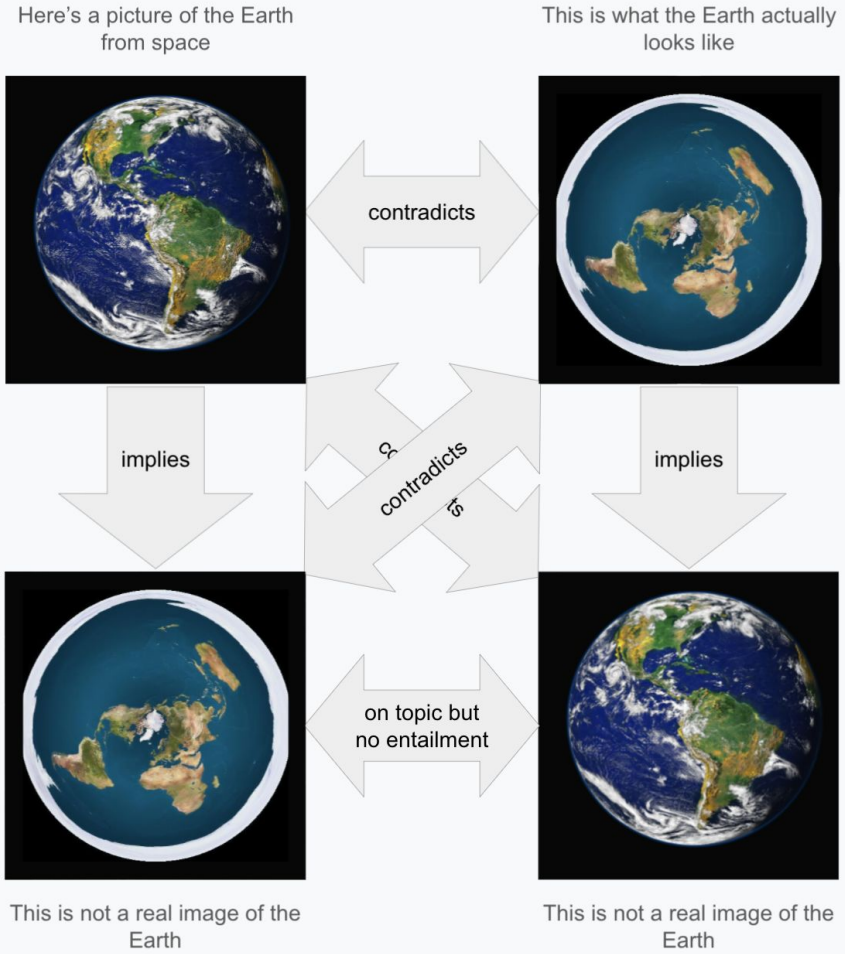
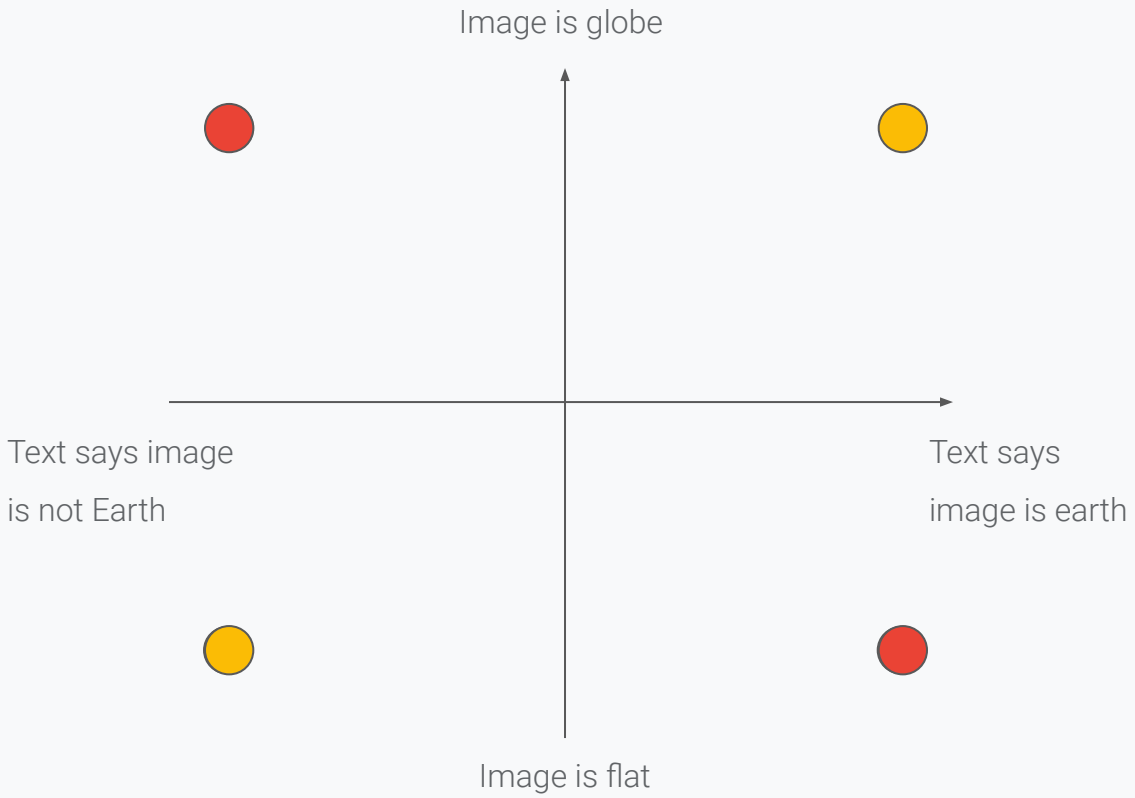


Image sources: pickpic and commons.wikipedia

Multimodal entailment

A complicated case of embeddings: how to go from image and text to "knowledge space"?



The challenges of multimodal entailment

- How can we generate representations of so many different kinds of content?
  - Language
  - Knowledge
  - Structured data
  - Vision
  - …
- How do we combine these representations / learn joint representations to provide true multimodal inference about content?
- How can we define the notion of entailment and agreement? How can we use the learned representations to classify whether e.g., two videos or a video and a text "match"?

Tutorial overview

What to expect today

- How to learn single-modal representations?
  - Text embedding and NLP
  - Using existing models to fine tune embeddings
  - Using embeddings to define textual entailment
  - Embedding structured and semi-structured data
  - Embedding knowledge using graphs
  - Foundations of computer vision

- How to learn multi-modal representations?
  - Attention bottlenecks for multimodal fusion
  - Self-supervised multimodal networks
  - Case studies: cross-modal fine-grained reasoning
  - Multimodal entailment theory and hands-on

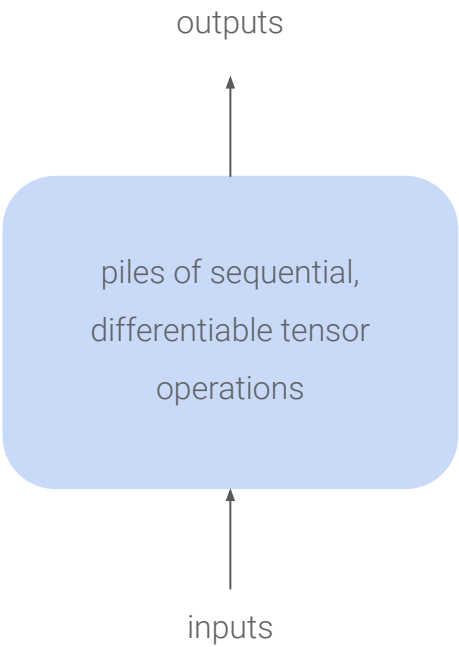
02

Natural Language Processing

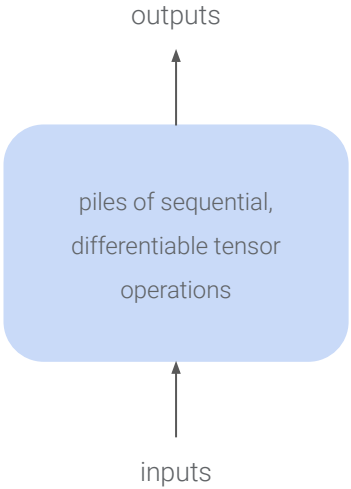
From word embeddings to contextualized representations

Sequence-to-sequence models

https://xkcd.com/1838/



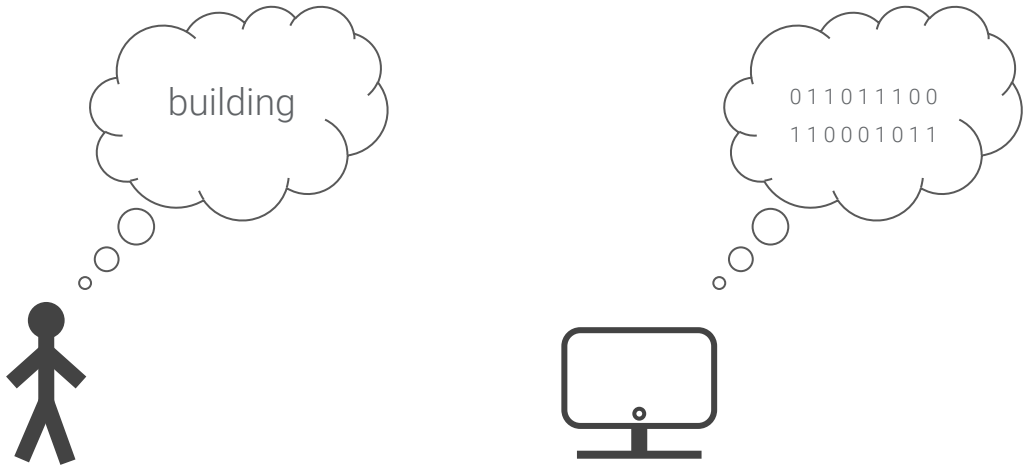
Sequence-to-sequence models



	inputs	outputs
Sentiment Analysis	Amazing movie, 10/10!	★★★★★
Machine Translation	Hello, world	Olá, mundo
Language modeling	-	The quick brown fox...
Speech recognition		Hello, world

Ilharco et al., EMNLP 2020

How can we make numeric representations out of words?



Ilharco et al., KDD 2019

Word embeddings

One-hot encodings

How to represent words?

Ilharco et al., KDD 2019

Word embeddings

Embeddings: learned latent representations of words

How to represent words?

Ilharco et al., KDD 2019

Token embeddings

Embeddings: learned latent representations of tokens

Decreased vocab size: words not in reduced dictionary will be split

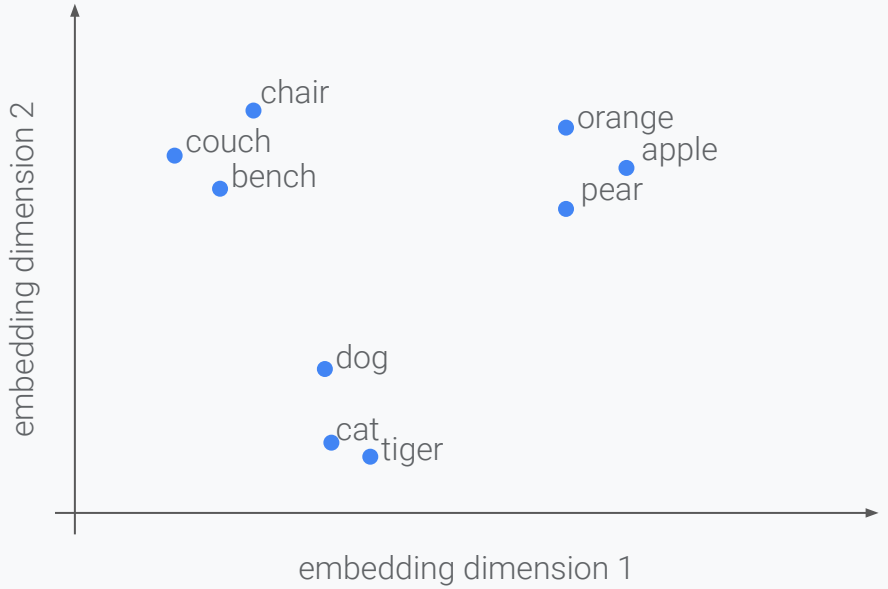
How to represent tokens?

Ilharco et al., KDD 2019

Word embeddings

An illustrative point

How to represent words?



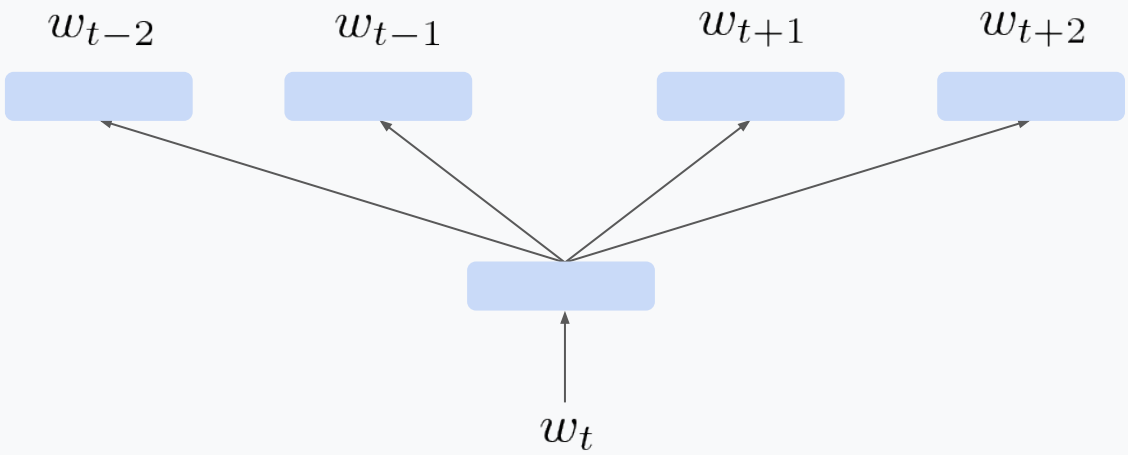
Ilharco et al., KDD 2019

Word embeddings

Distributed Representations of Words and Phrases and their Compositionality (Mikolov et al., 2013)

= dense layer

Word2Vec — Skip-gram



From word to contextual embeddings

Google AI

Sit by the **fire** and relax

They wouldn't **fire** John, he is a great employee

Jill has been publishing a lot recently, she's on **fire**

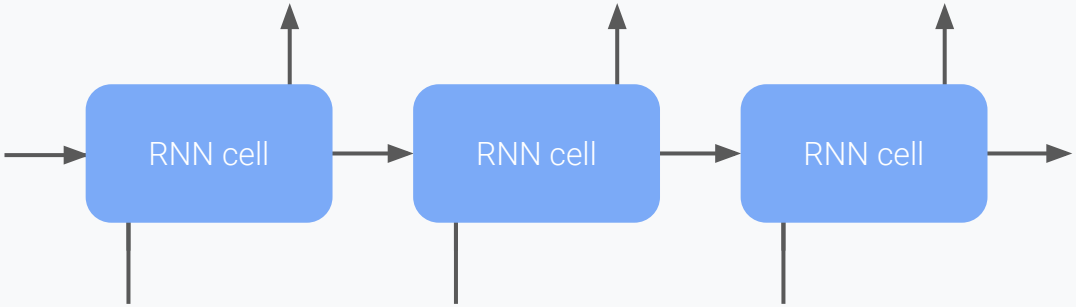
Neurons that **fire** together wire together

Nerf guns **fire** foam darts

• • •

RNNs

Computations over
sequences of arbitrary length



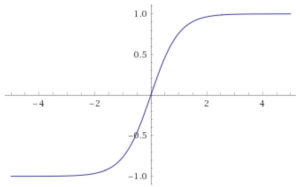
Ilharco et al., EMNLP 2020

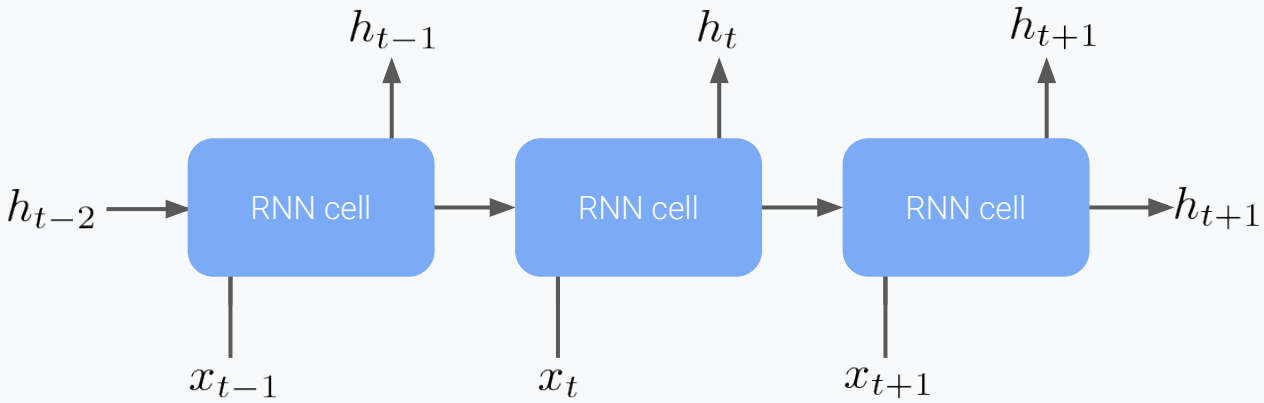
RNNs

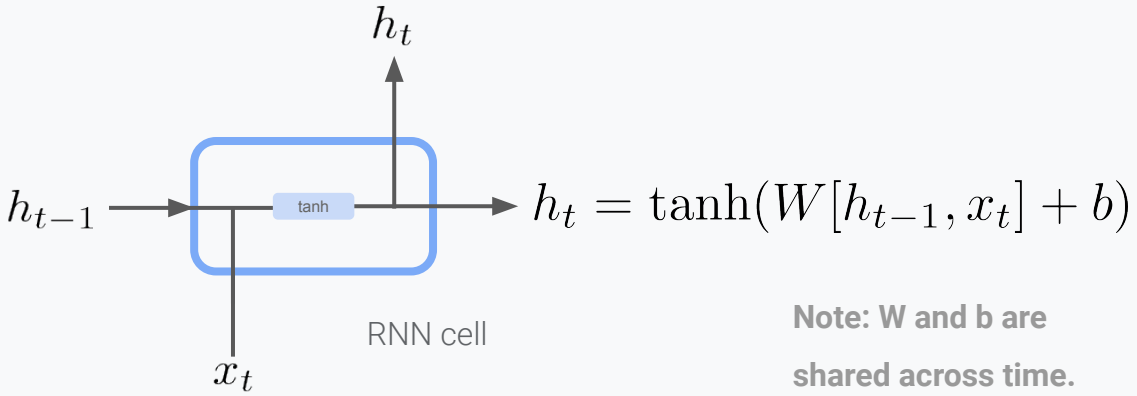
Computations over sequences of arbitrary length

= dense layer

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$







Note: W and b are shared across time.

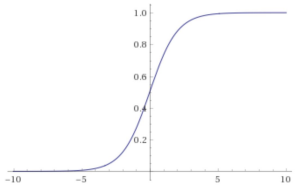
LSTMs

Or LSTMs

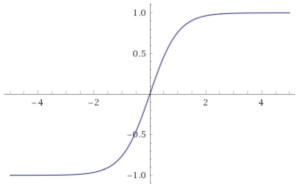
= dense layer

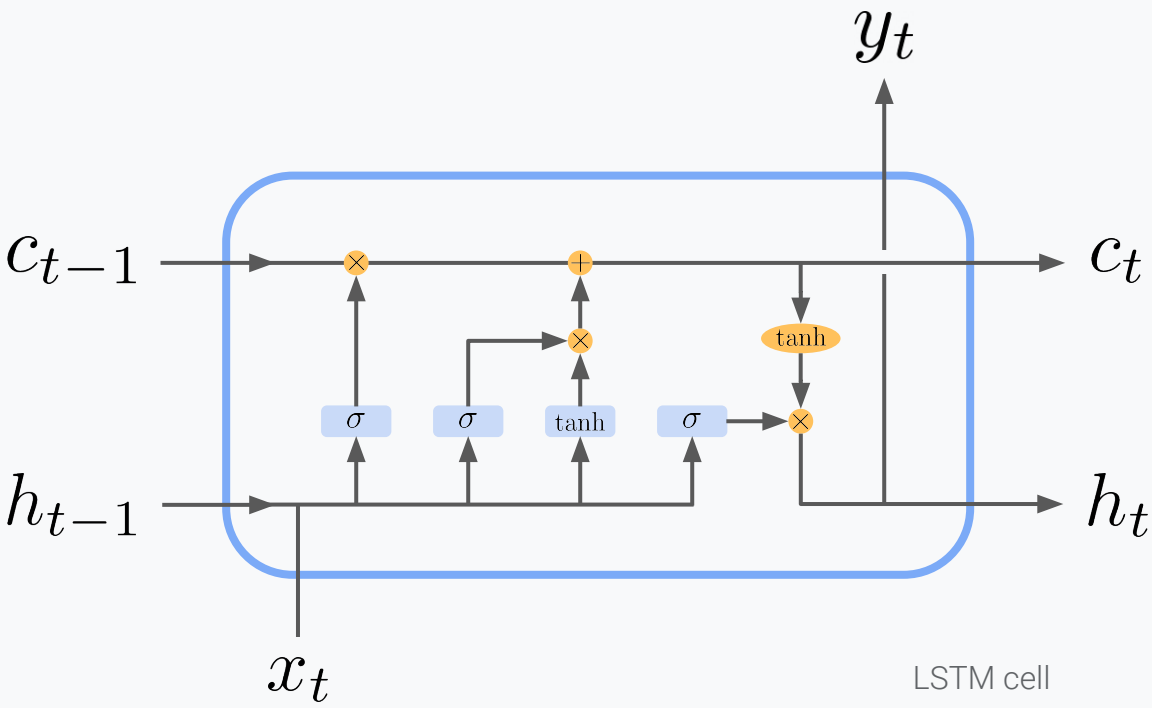
= pointwise operation

$$\sigma(x) = \frac{e^x}{e^x + 1}$$



$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$





LSTM cell

GRUs

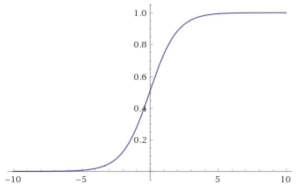
RNNs come in multiple 'flavors', e.g. GRUs

= concatenate

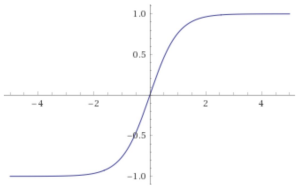
= dense layer

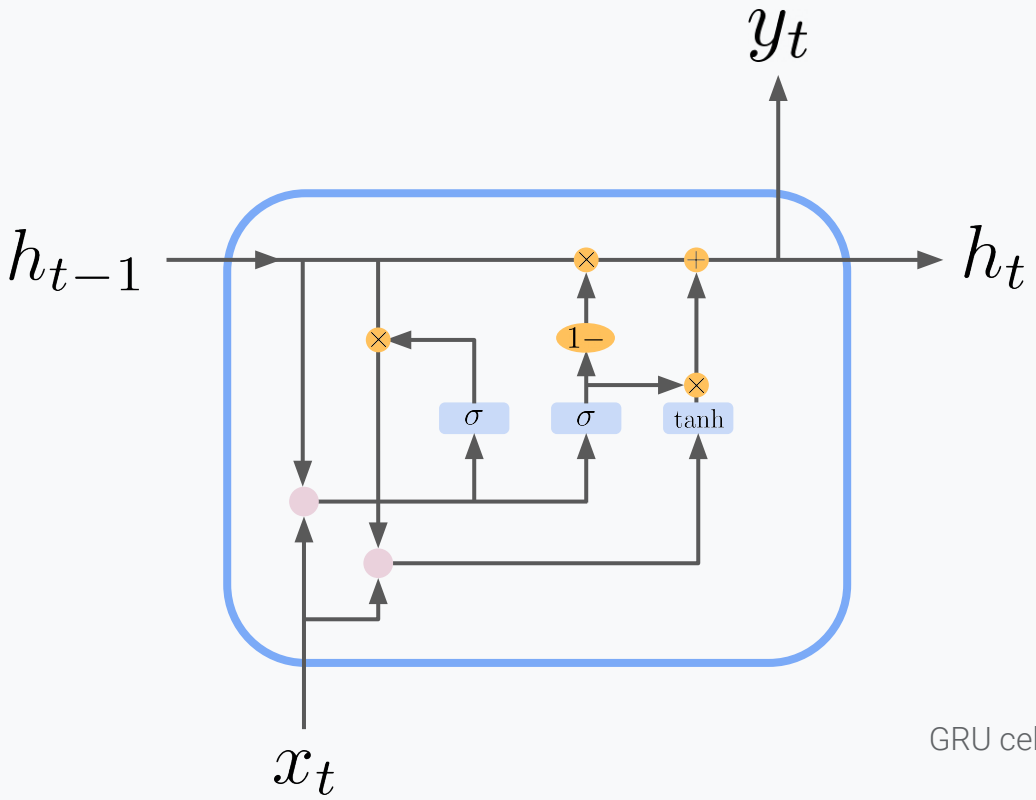
= pointwise operation

$$\sigma(x) = \frac{e^x}{e^x + 1}$$



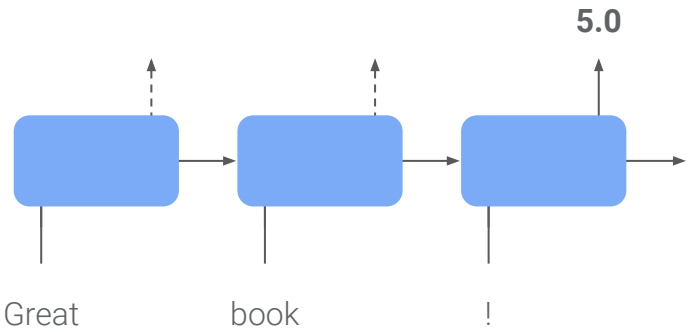
$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$





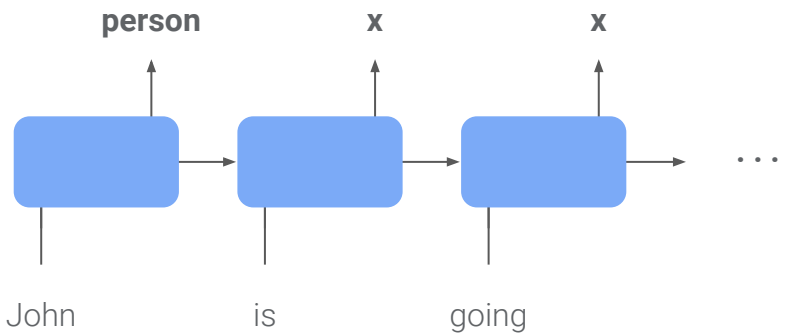
GRU cell

Sentiment analysis



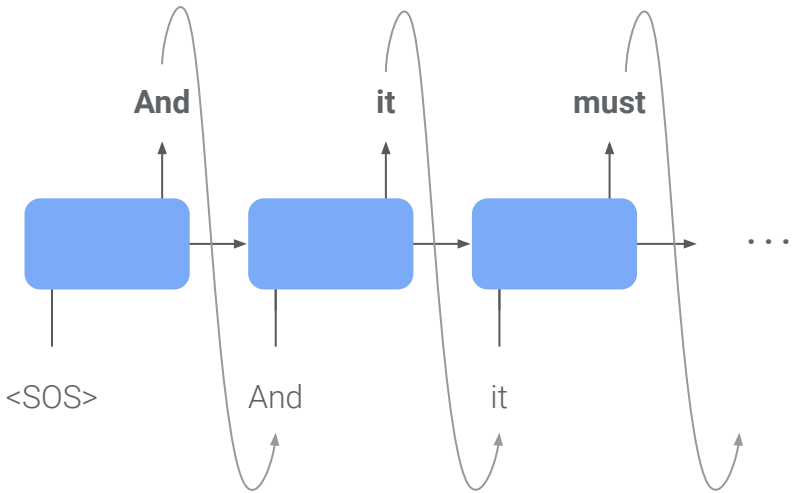
Ilharco et al., EMNLP 2020

Named-entity recognition



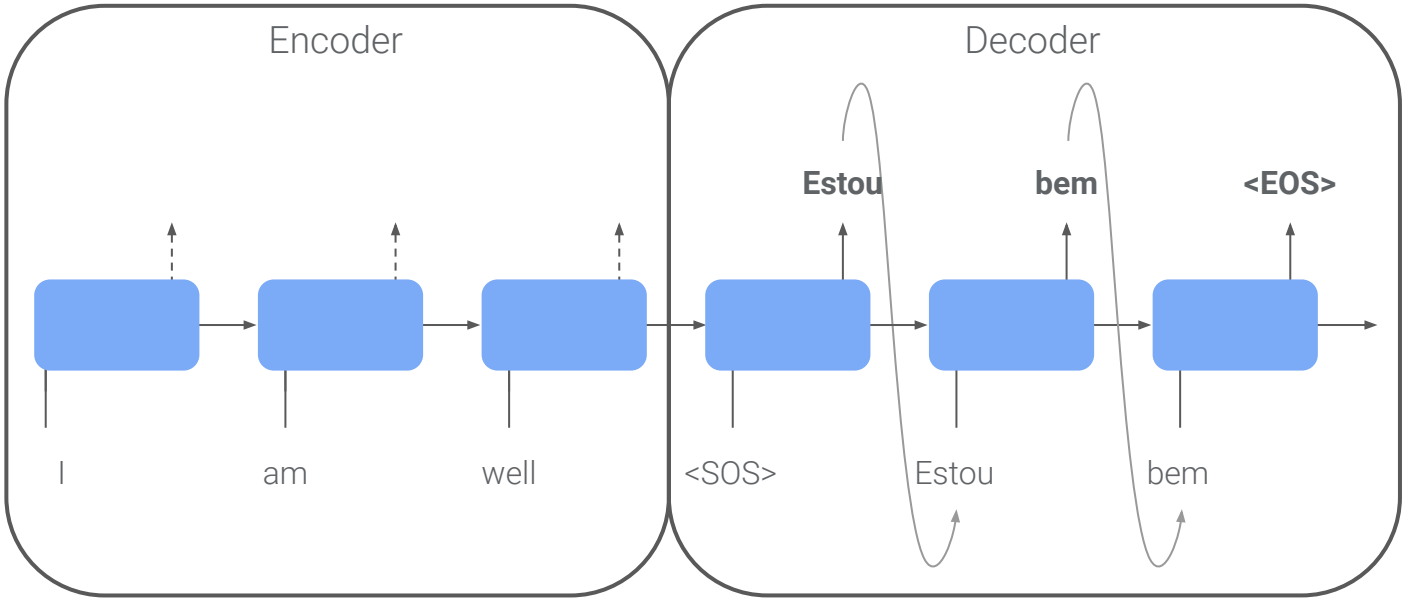
Ilharco et al., EMNLP 2020

Language Models



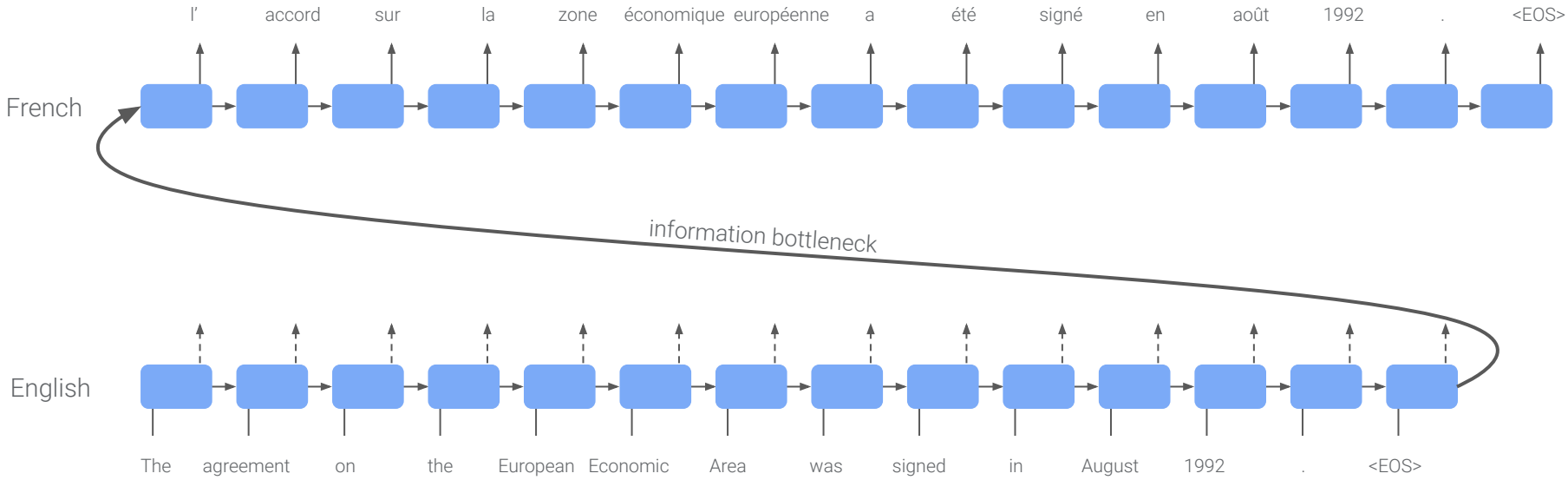
Ilharco et al., EMNLP 2020

Machine Translation



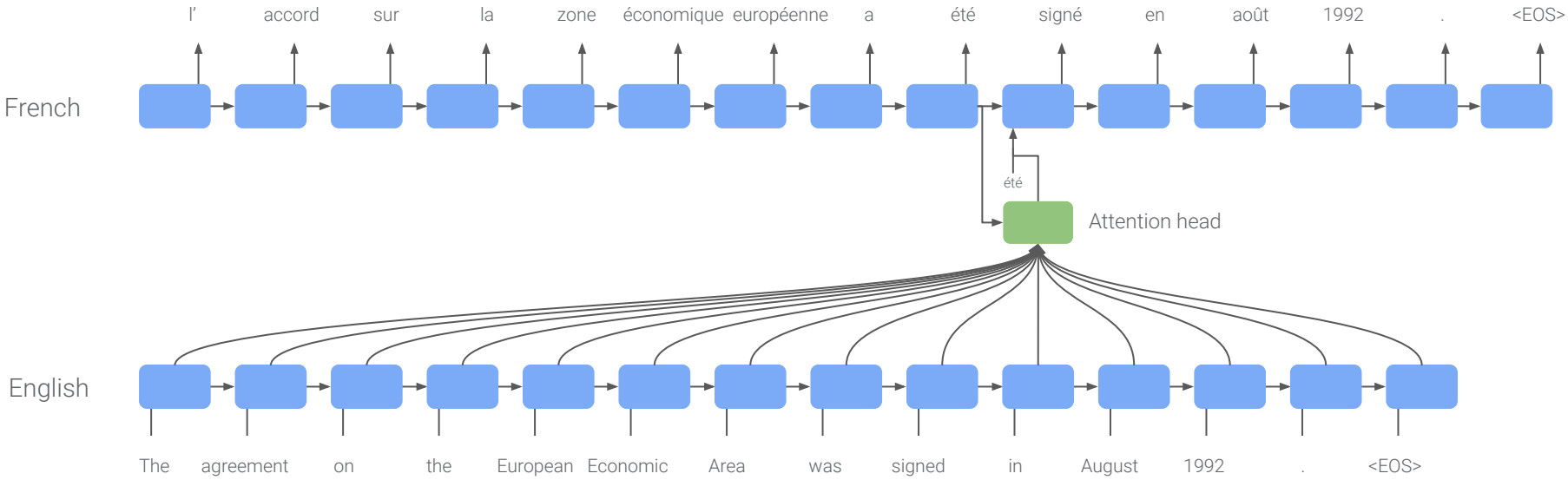
Ilharco et al., EMNLP 2020

The encoder-decoder bottleneck



Example derived from [Bahdanau, et al. 2014]

Attention



Example derived from [Bahdanau, et al. 2014]

Attention

Bahdanau et al.

Neural machine translation by jointly learning to align and translate. 2014

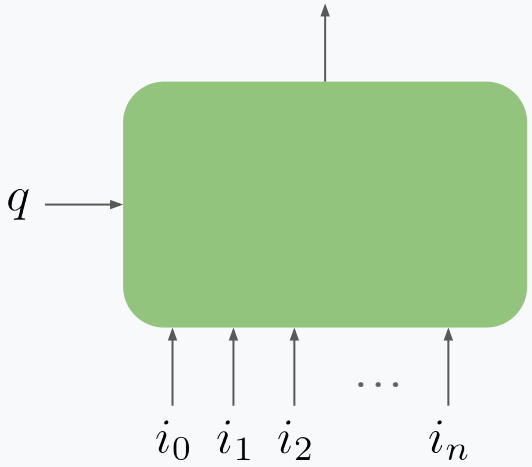
Thang Luong et al.

Effective approaches to attention-based neural machine translation. 2015

A *summary* of *i*,

based on how similar their

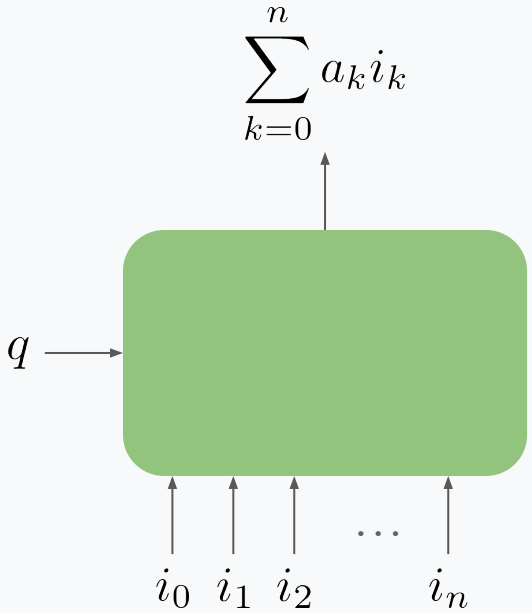
are with the query



Dot product attention

Thang Luong et al.

Effective approaches to attention-based neural machine translation. 2015



Attention head

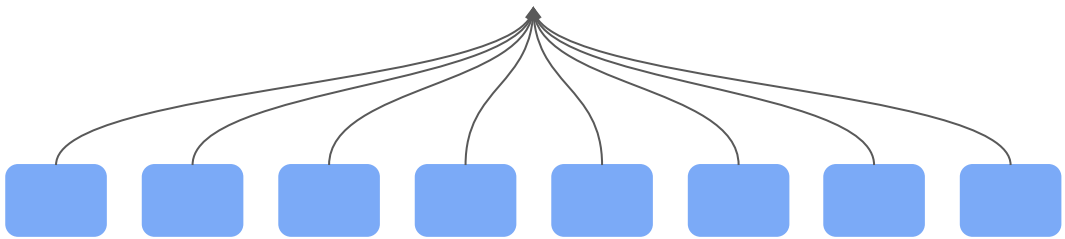
$$\phi(q, i_k) = \exp(q^\top i_k)$$

$$a_k = \frac{\phi(q, i_k)}{\sum_{j=0}^{n} \phi(q, i_j)}$$

Transformers

Motivation

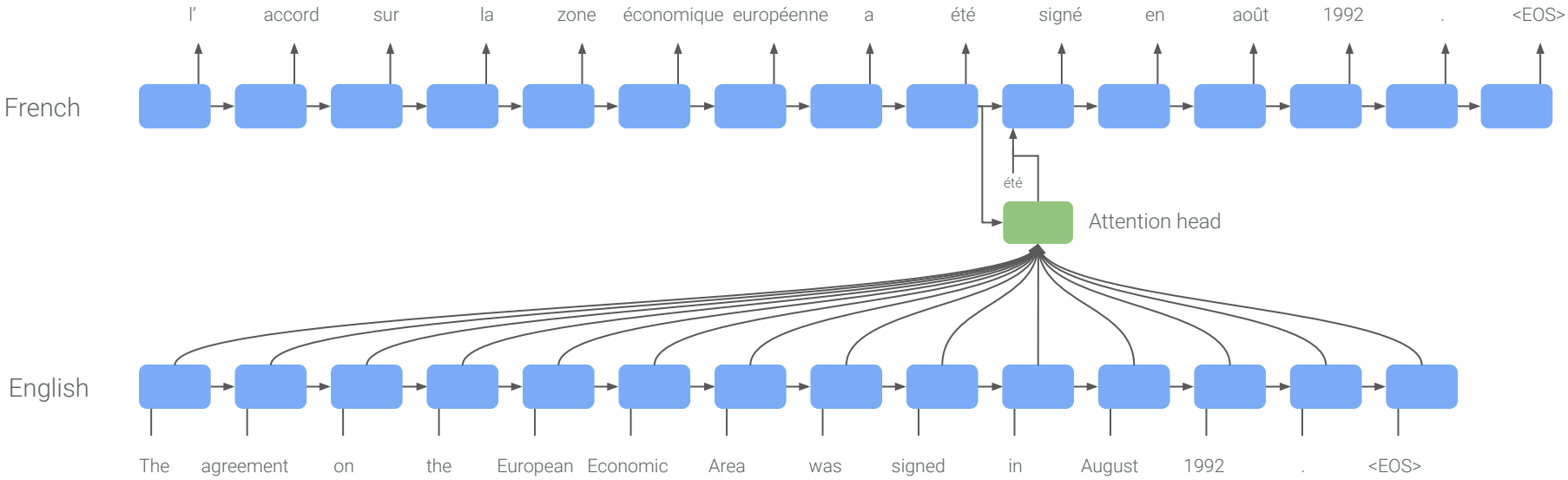
sequential



parallel

Ilharco et al., EMNLP 2020

Motivation

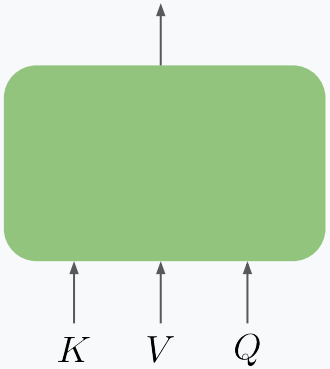


Example derived from Bahdanau, et al. 2014 (https://arxiv.org/pdf/1409.0473.pdf)

Scaled Dot-Product Attention

Queries, keys and values

A *summary* of values,
based on how similar their
corresponding keys are
with the query



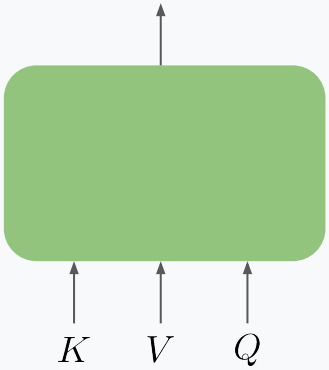
Ilharco et al., EMNLP 2020

Scaled Dot-Product Attention

Queries, keys and values

For some similarity function ϕ

$$O_i = \sum_{j=0}^{l} a_{ij} V_j$$



$$a_{ij} = \frac{\phi(Q_i, K_j)}{\sum_{p=0}^{l} \phi(Q_i, K_p)}$$

Ilharco et al., EMNLP 2020

Scaled Dot-Product Attention

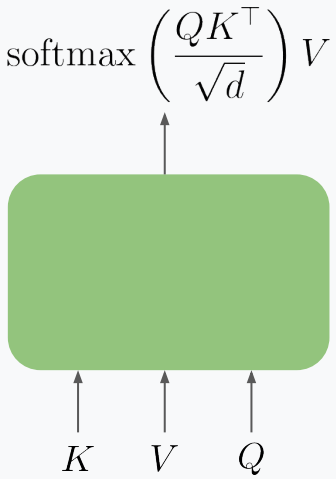
Using dot-product similarity, we can vectorize nicely

$$\phi(Q_i, K_j) = \exp\left(\frac{Q_i K_j^\top}{\sqrt{d}}\right)$$

d = feature dim

$$\text{softmax}(x)_i = \frac{\exp x_i}{\sum_j \exp x_j}$$

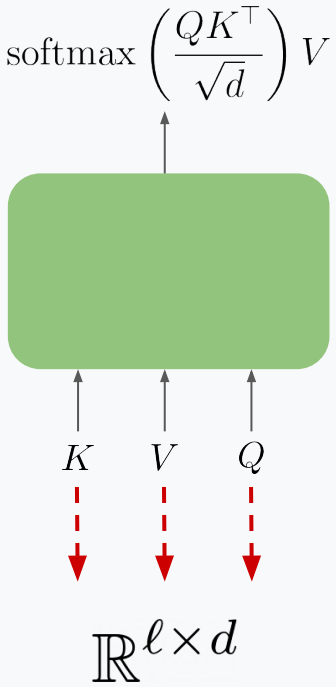
Normalization factor for numerical stability



Scaled Dot-Product Attention

Let's dive into the dimensions
(batch omitted for simplicity)

ℓ = sequence length
d = feature dim

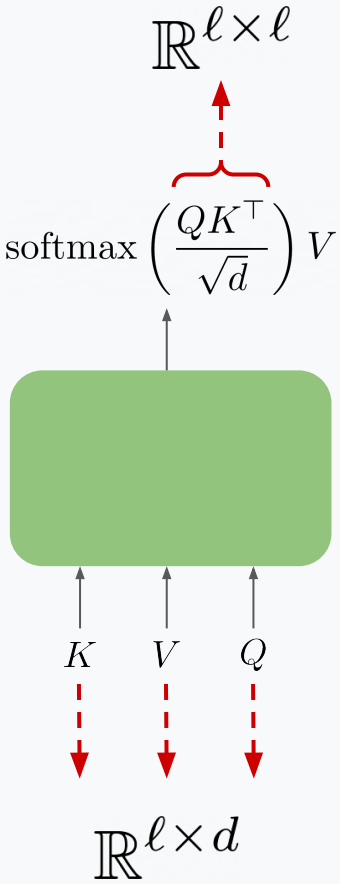


Ilharco et al., EMNLP 2020

Scaled Dot-Product Attention

Let's dive into the dimensions
(batch omitted for simplicity)

ℓ = sequence length
d = feature dim

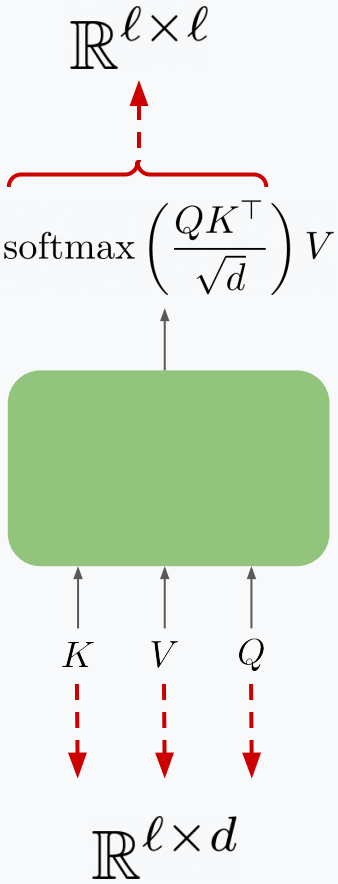


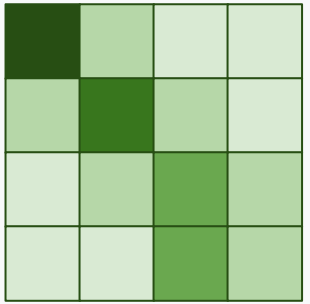
Ilharco et al., EMNLP 2020

Scaled Dot-Product Attention

Let's dive into the dimensions
(batch omitted for simplicity)

ℓ = sequence length
d = feature dim



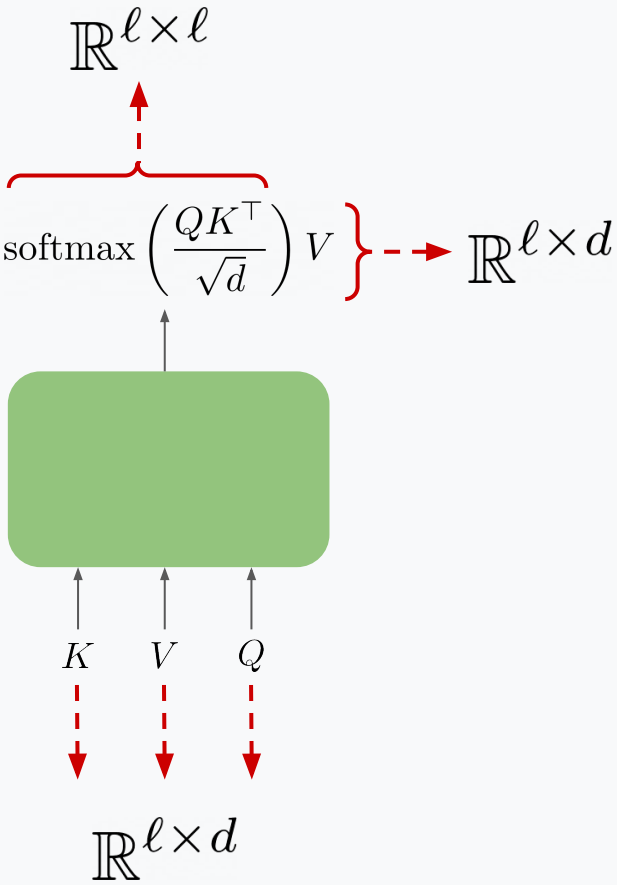


Ilharco et al., EMNLP 2020

Scaled Dot-Product Attention

Let's dive into the dimensions
(batch omitted for simplicity)

ℓ = sequence length
d = feature dim

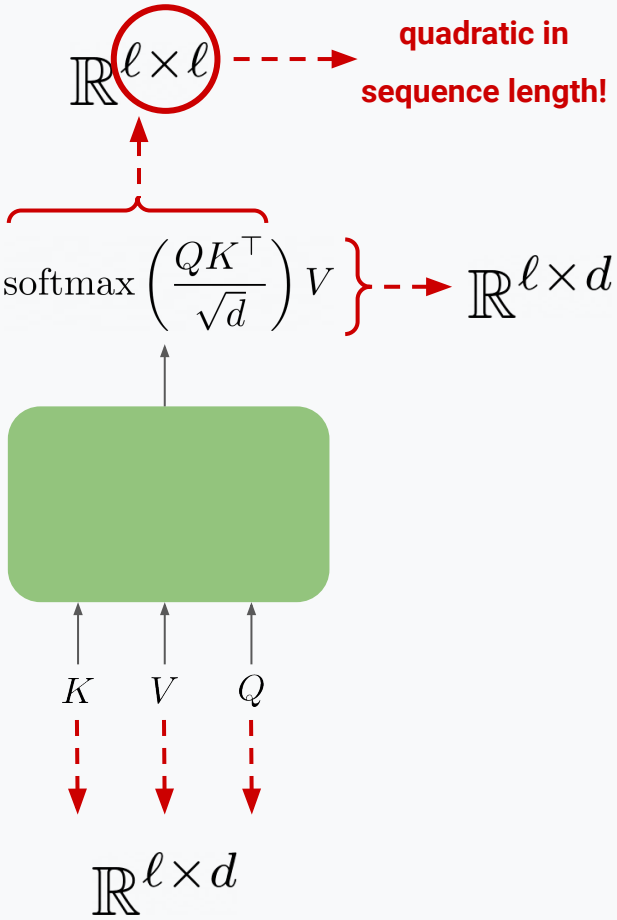


Ilharco et al., EMNLP 2020

Scaled Dot-Product Attention

Let's dive into the dimensions
(batch omitted for simplicity)

ℓ = sequence length
d = feature dim



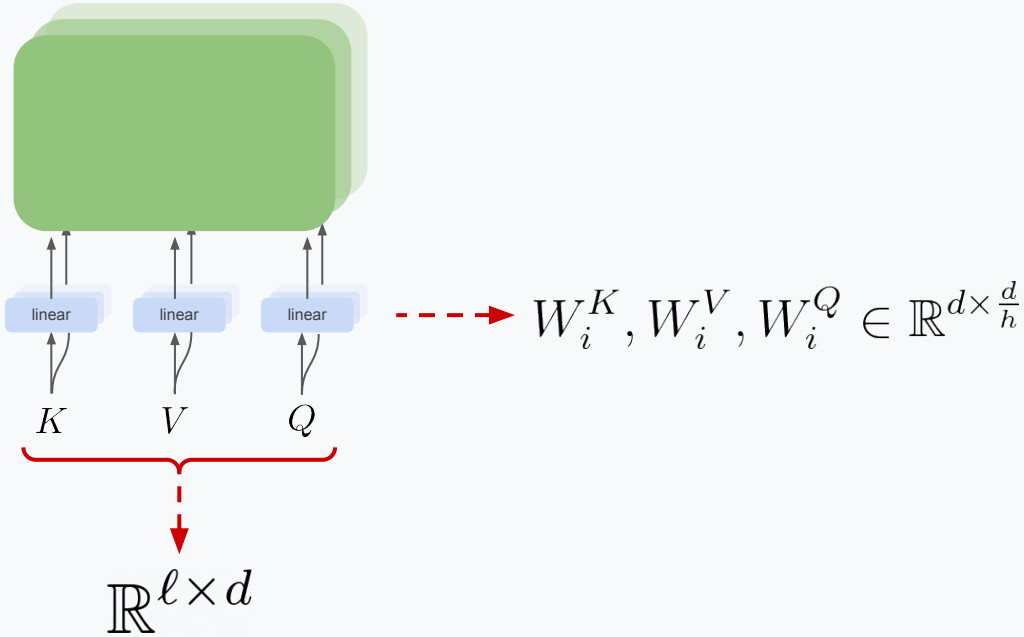
Ilharco et al., EMNLP 2020

Multi-head attention

ℓ = sequence length

d = feature dim

h = # of attention heads



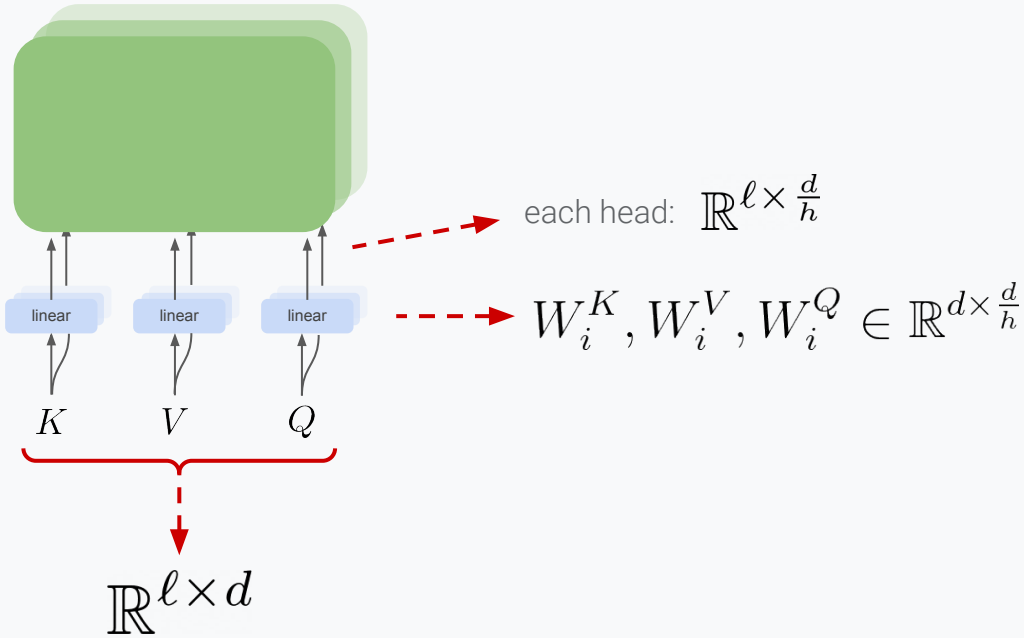
Ilharco et al., EMNLP 2020

Multi-head attention

ℓ = sequence length

d = feature dim

h = # of attention heads



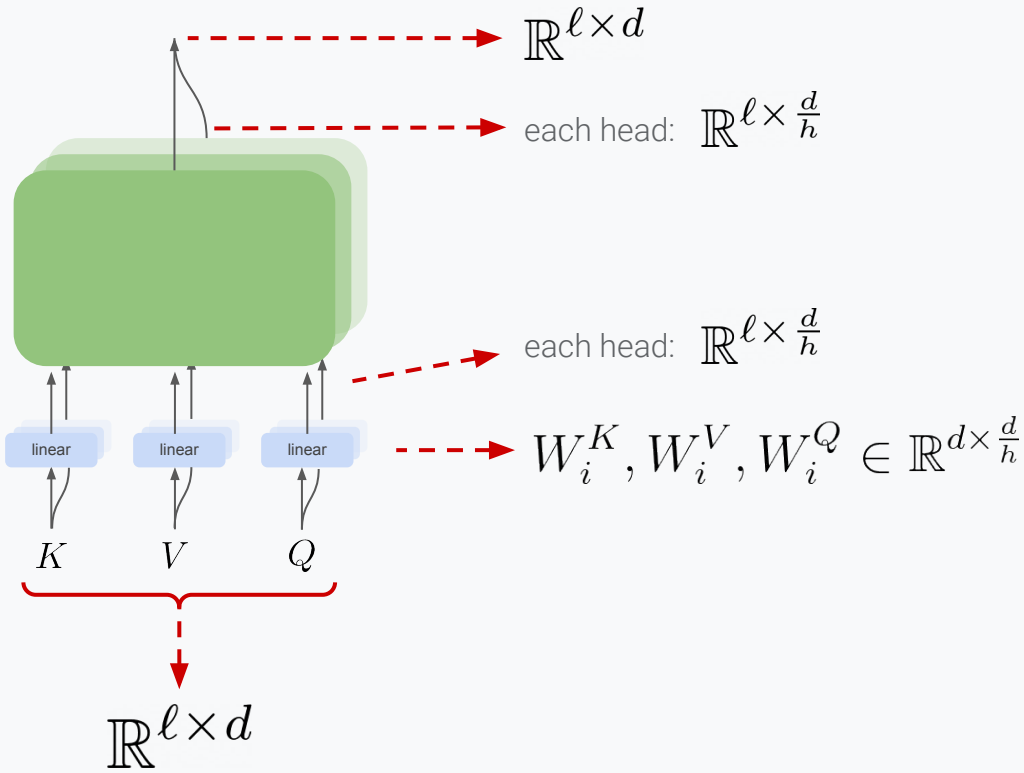
Ilharco et al., EMNLP 2020

Multi-head attention

ℓ = sequence length

d = feature dim

h = # of attention heads



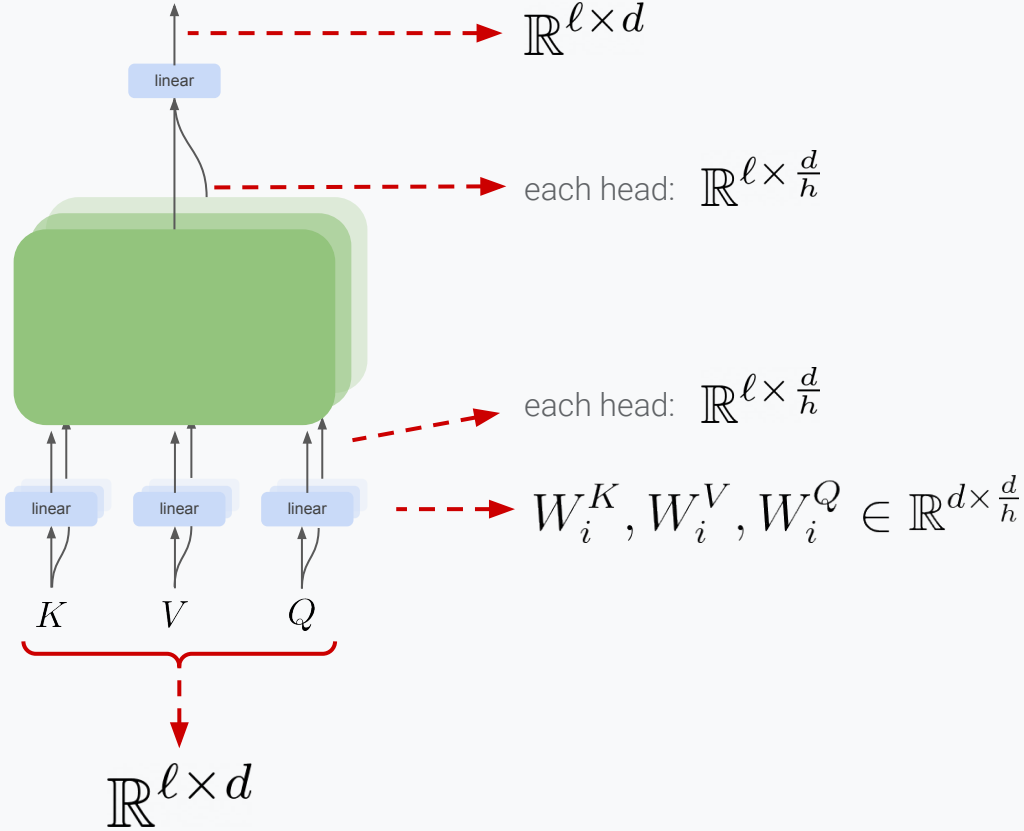
Ilharco et al., EMNLP 2020

Multi-head attention

ℓ = sequence length

d = feature dim

h = # of attention heads



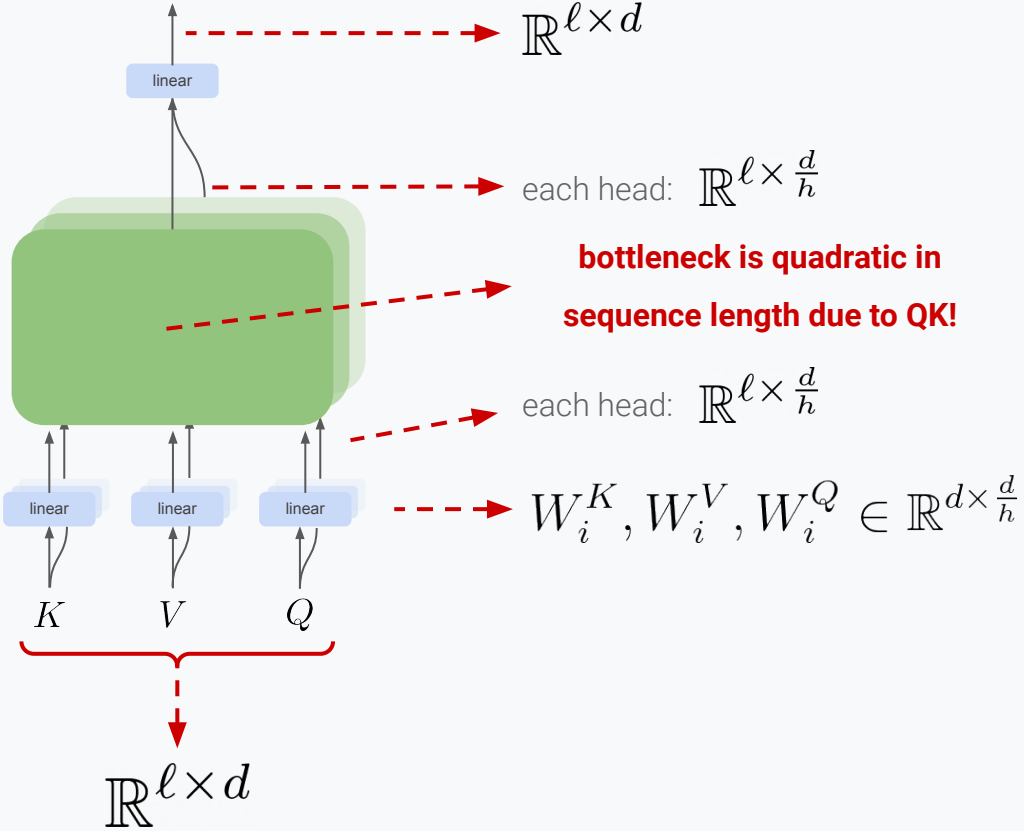
Ilharco et al., EMNLP 2020

Multi-head attention

ℓ = sequence length

d = feature dim

h = # of attention heads

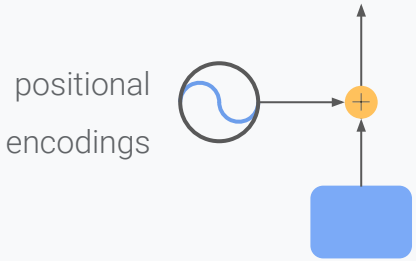


Ilharco et al., EMNLP 2020

Positional encodings

So far, attention has been a set operation.

Let's add positional information!



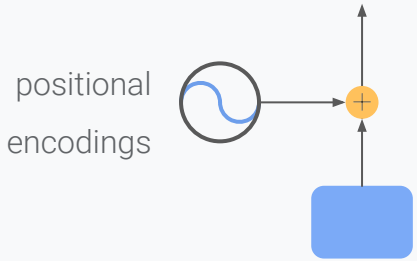
Ilharco et al., EMNLP 2020

Positional encodings

So far, attention has been a set operation.

Let's add positional information!

These can be either **learned** or **fixed**.

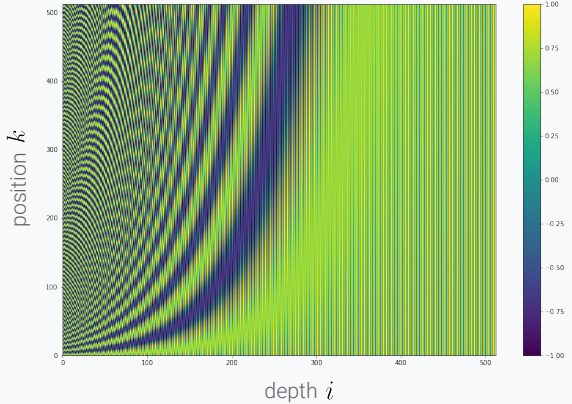


Fixed:

for a position k in the sequence

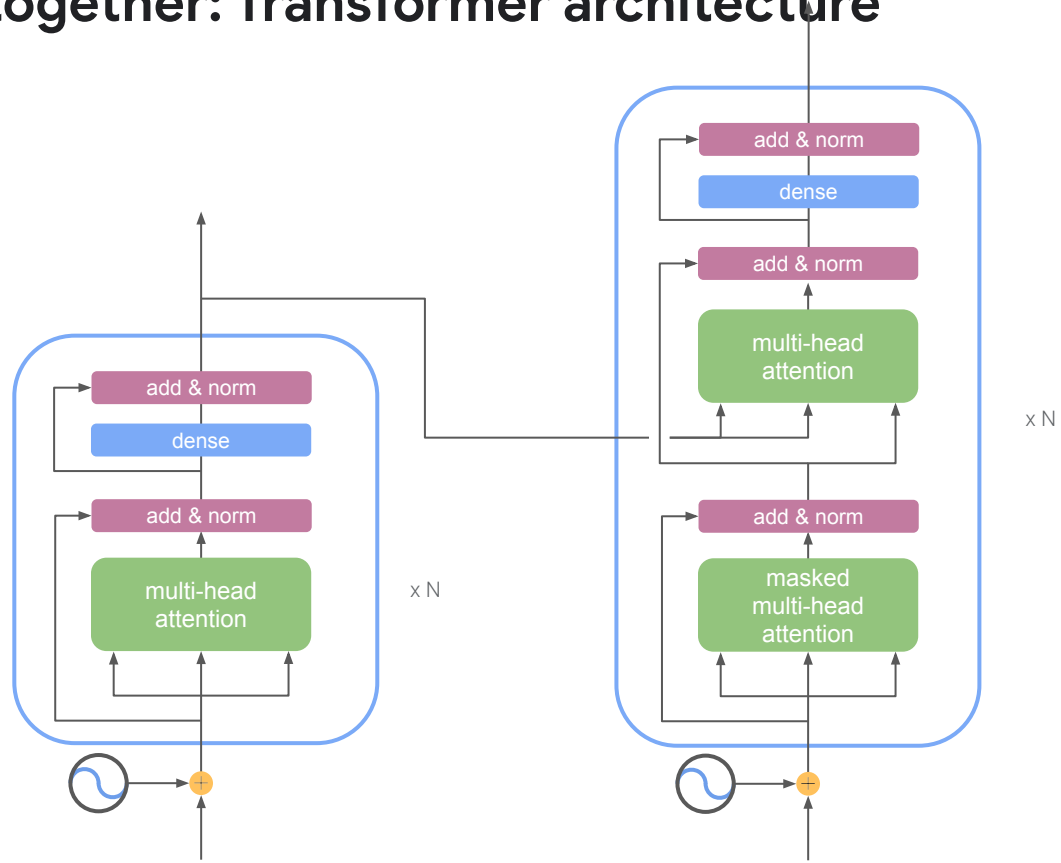
and i in the feature space

$$E_{ki} = \begin{cases} \sin\left(k/10000^{\frac{i}{N}}\right) & \text{if } i \text{ is even} \\ \cos\left(k/10000^{\frac{i-1}{N}}\right) & \text{if } i \text{ is odd} \end{cases}$$



Ilharco et al., EMNLP 2020

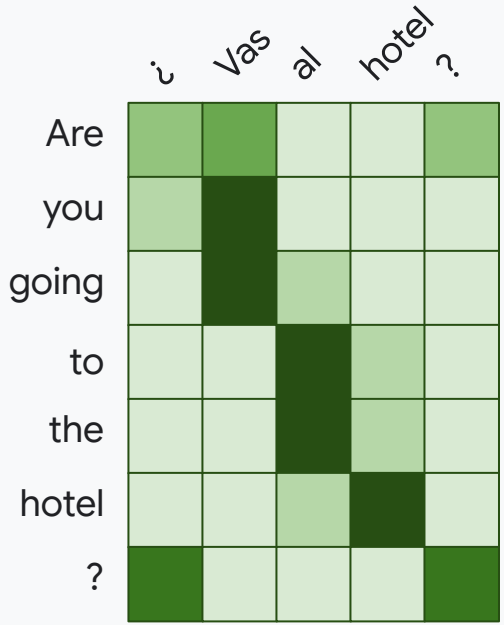
Putting it all together: Transformer architecture



Vaswani et al., 2017

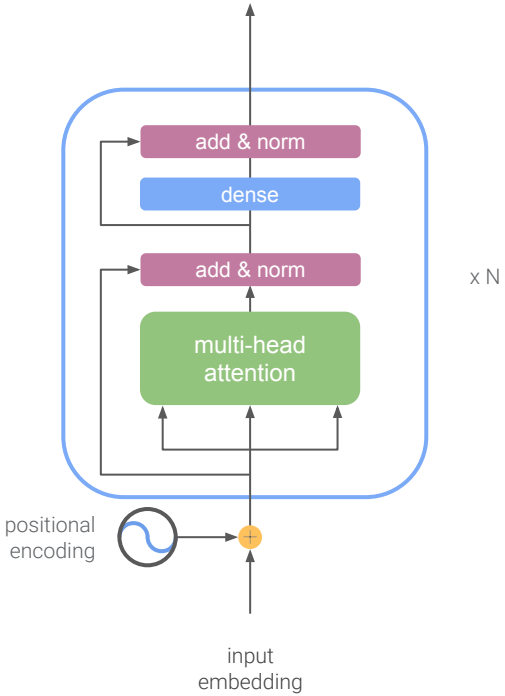
Attention mechanisms

The attention matrix



Ilharco et al., EMNLP 2020

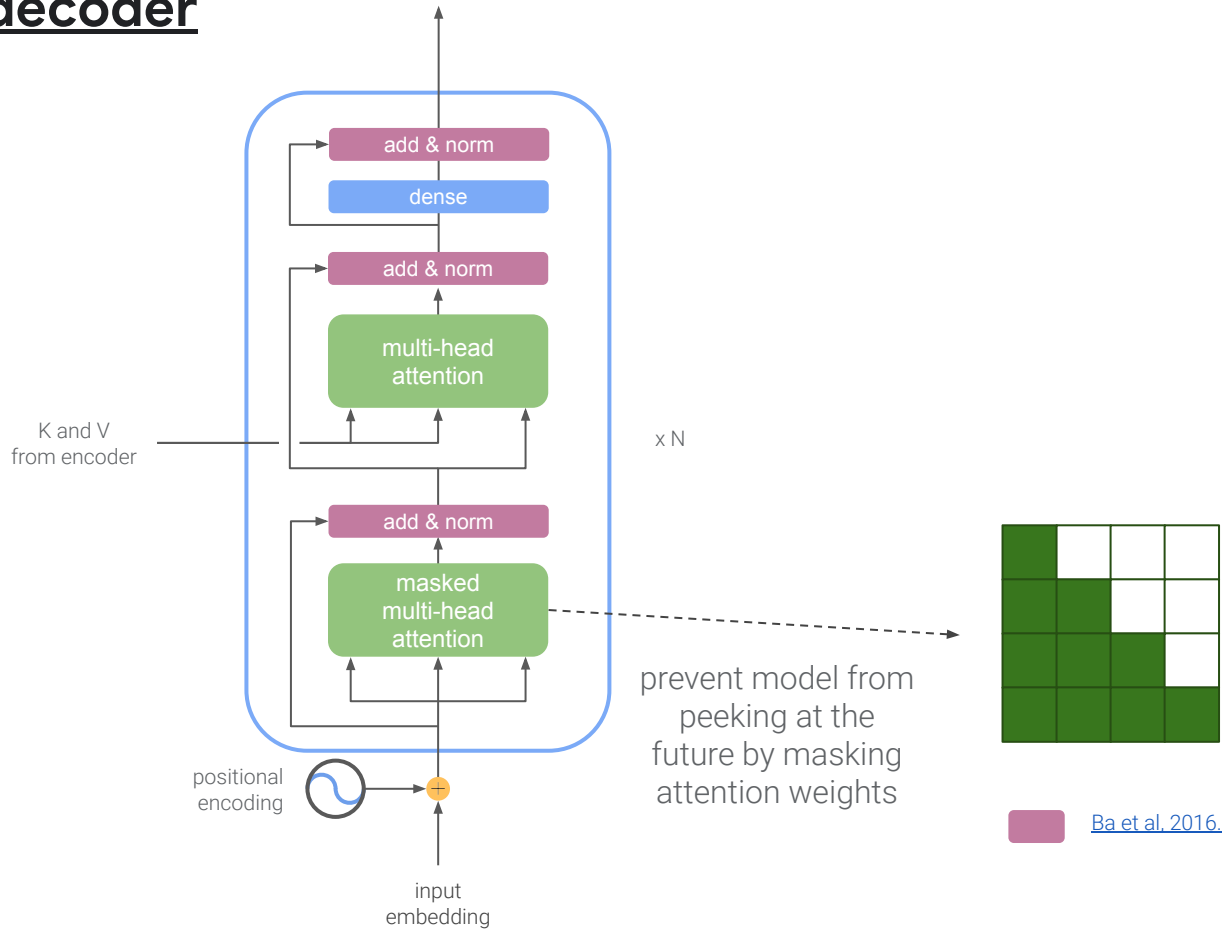
The transformer encoder



Ba et al, 2016.

Vaswani et al., 2017

The transformer decoder

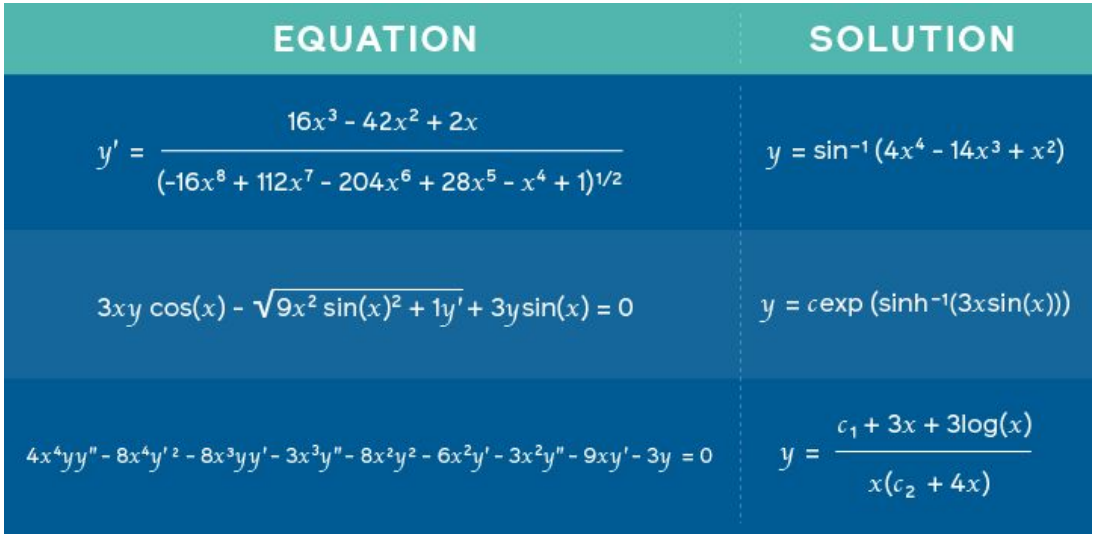


Vaswani et al., 2017

Transformers in recent literature

Transformers have become **ubiquitous in NLP,** and successful in a wide range of domains and applications, including:

- Mathematics and theorem proving (e.g. Lample et al., 2019, Clark et al., 2020)

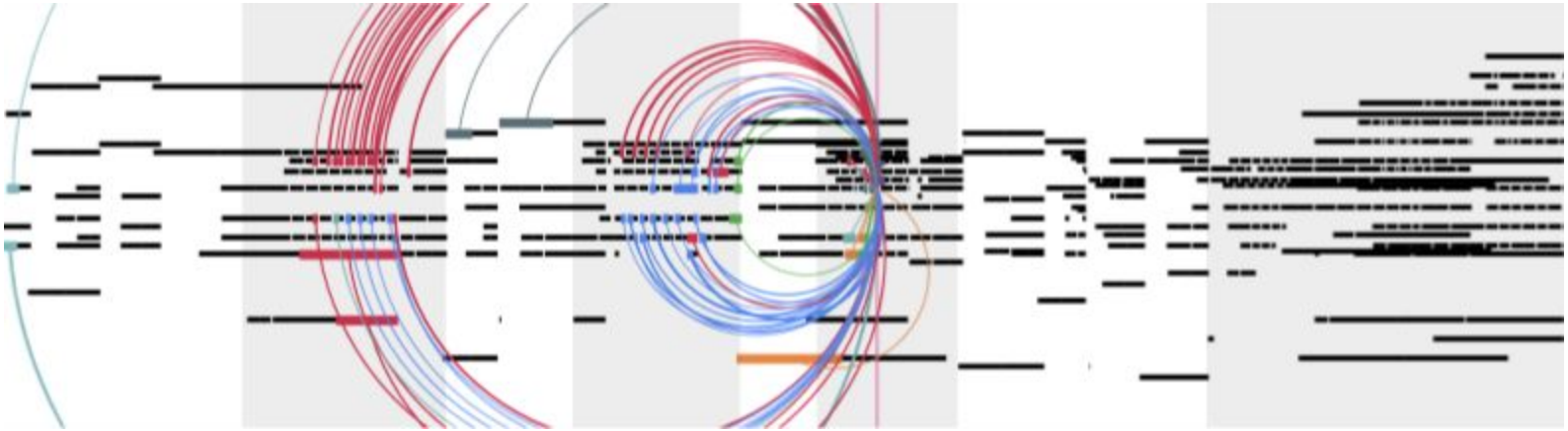


https://ai.facebook.com/blog/using-neural-networks-to-solve-advanced-mathematics-equations/

Transformers in recent literature

Transformers have become **ubiquitous in NLP,** and successful in a wide range of domains and applications, including:

- Mathematics and theorem proving (e.g. Lample et al., 2019, Clark et al., 2020)
- Music generation (e.g. Anna Huang et al., 2019)

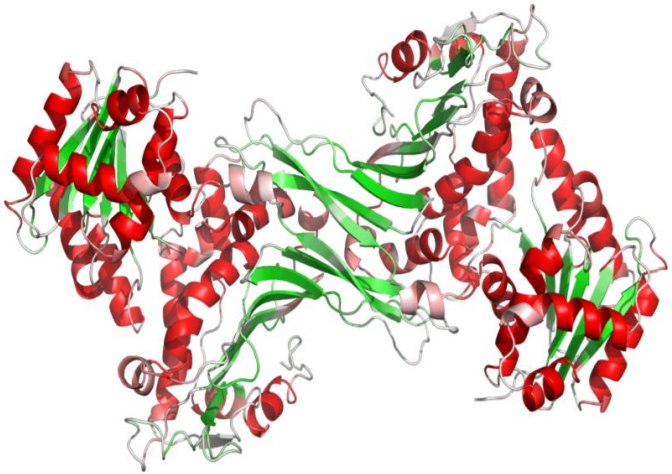


https://magenta.tensorflow.org/music-transformer

Transformers in recent literature

Transformers have become **ubiquitous in NLP,** and successful in a wide range of domains and applications, including:

- Mathematics and theorem proving (e.g. Lample et al., 2019, Clark et al., 2020)
- Music generation (e.g. Anna Huang et al., 2019)
- Biology (e.g. Rives et al., 2019, Madani et al., 2020)



Transformers in recent literature

Transformers have become **ubiquitous in NLP,** and successful in a wide range of domains and applications, including:

- Mathematics and theorem proving (e.g. Lample et al., 2019, Clark et al., 2020)
- Music generation (e.g. Anna Huang et al., 2019)
- Biology (e.g. Rives et al., 2019, Madani et al., 2020)
- Vision and Language (e.g. Tan et al., 2019, Lu et al., 2019, Chen et al., 2020)

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

Visual Question Answering (Agrawal et al., 2015)

Transformers in recent literature

Transformers have become **ubiquitous in NLP,** and successful in a wide range of domains and applications, including:

- Mathematics and theorem proving (e.g. Lample et al., 2019, Clark et al., 2020)
- Music generation (e.g. Anna Huang et al., 2019)
- Biology (e.g. Rives et al., 2019, Madani et al., 2020)
- Vision and Language (e.g. Tan et al., 2019, Lu et al., 2019, Chen et al., 2020)
- Computer Vision (e.g. Ramachandran et al., 2019, Dosovitskiy et al., 2021)

	Vision Transformer	BiT-L (ResNet 152)
ImageNet	**88.4**	87.5
CIFAR-100	**94.6**	93.5
VTAB (19 tasks)	**77.2**	75.9
TPU v3 -days	**2.5k**	9.9k

Transformers in NLP

In NLP, large-scale self-supervised pre-training methods have been enormously successful (e.g. BERT, ALBERT, T5, GPT-3, among many others).

Models are typically used in 3 different scenarios:

Pre-training

- Large corpus (e.g. web crawled data)
- Typically unsupervised (e.g. masked language modeling)
- Usually runs in GPUs or TPUs

Fine-tuning

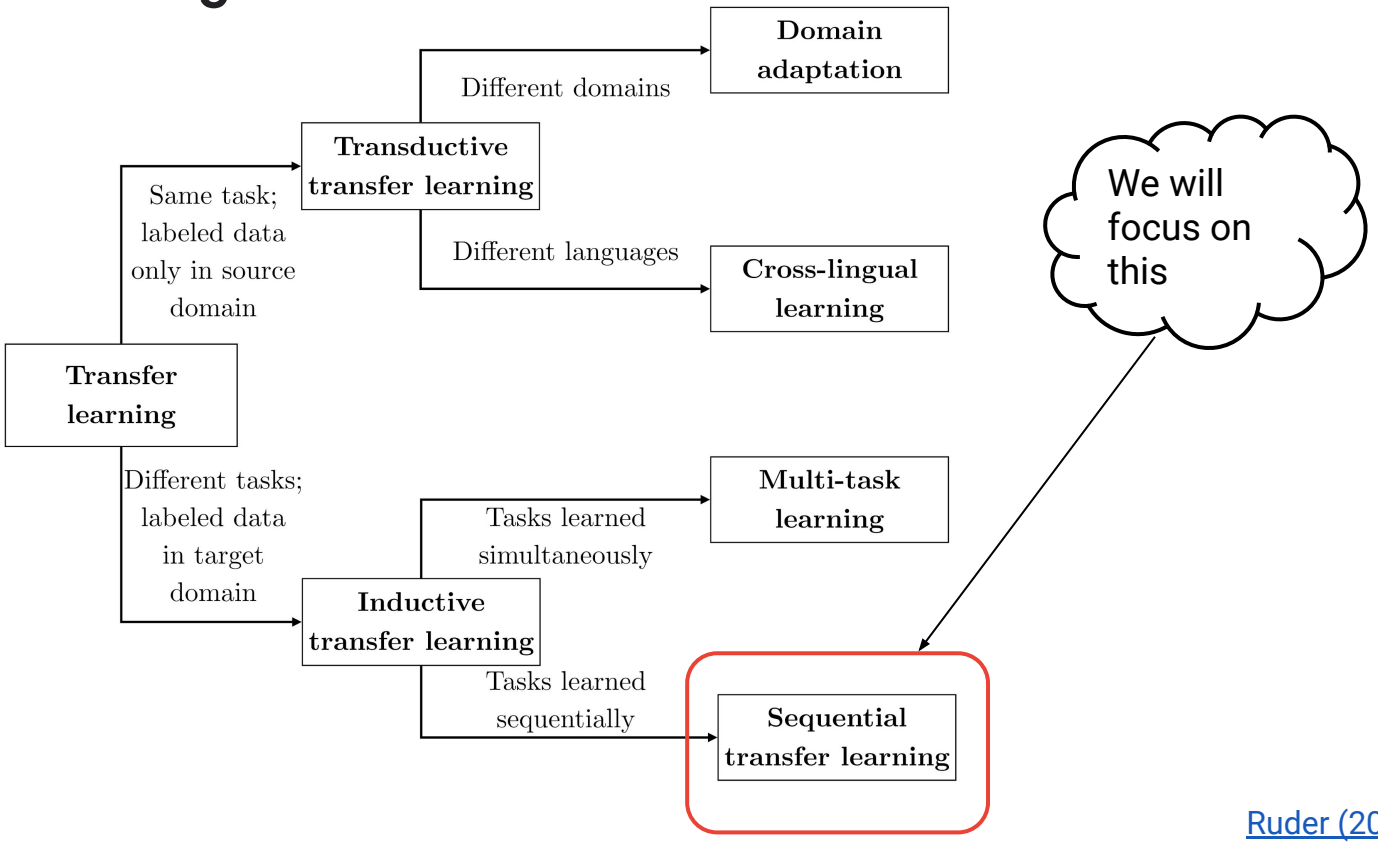
- Smaller corpus
- Typically supervised (e.g. question answering, natural language inference)
- Usually runs in GPUs or TPUs

Production

- Inference
- Usually runs in CPUs, sometimes in mobile devices

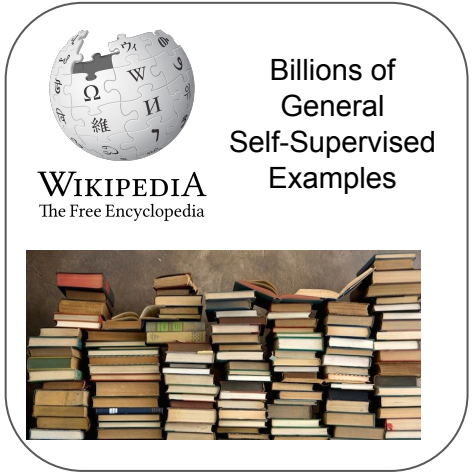
Fine-tuning pretrained models on downstream tasks

Transfer Learning in NLP



Ruder (2019)

Sequential Transfer Learning in NLP



Immense Computational Resources

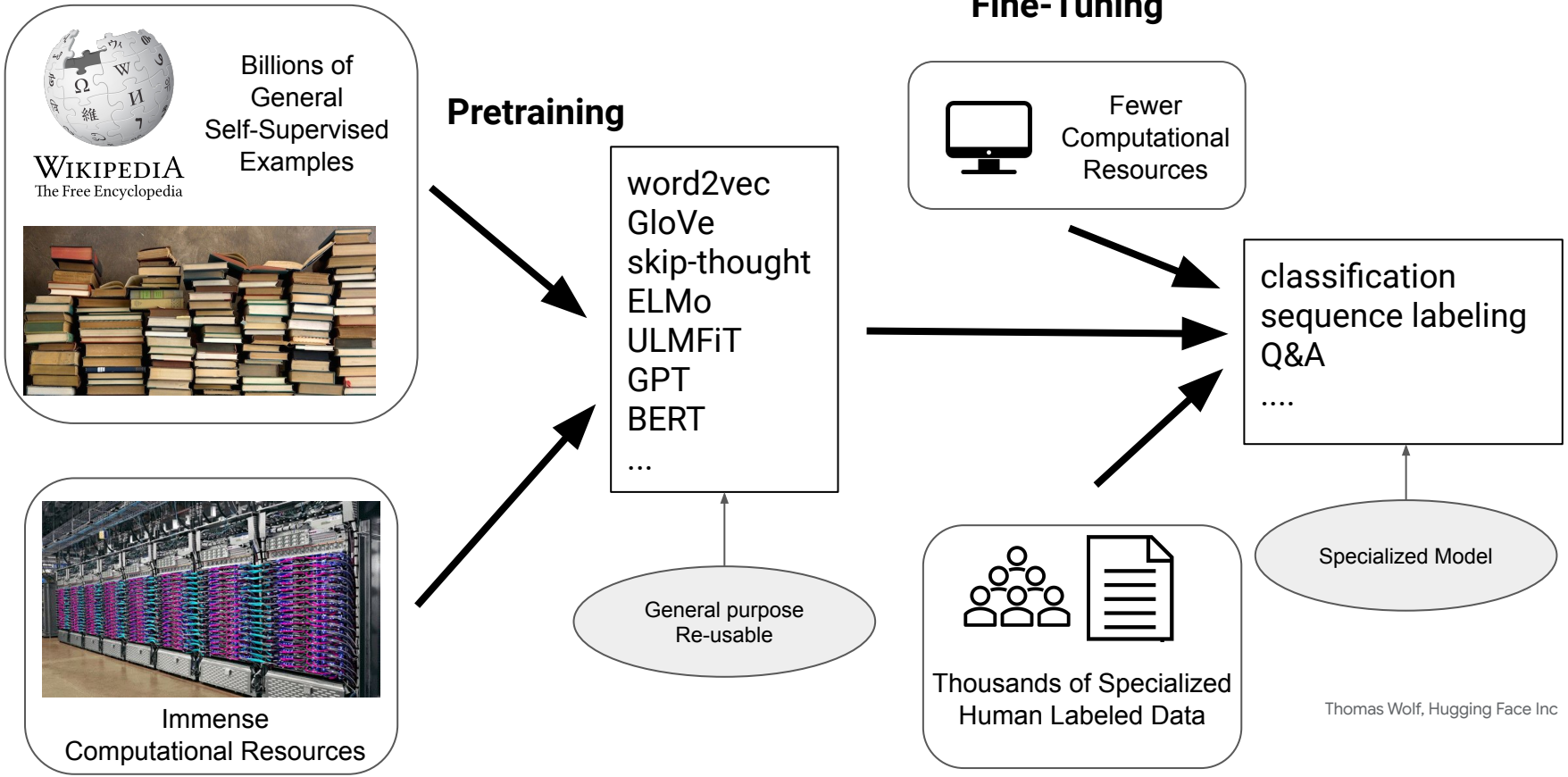
Pretraining

word2vec
GloVe
skip-thought
ELMo
ULMFiT
GPT
BERT
...

General purpose
Re-usable

Thomas Wolf, Hugging Face Inc

Sequential Transfer Learning in NLP

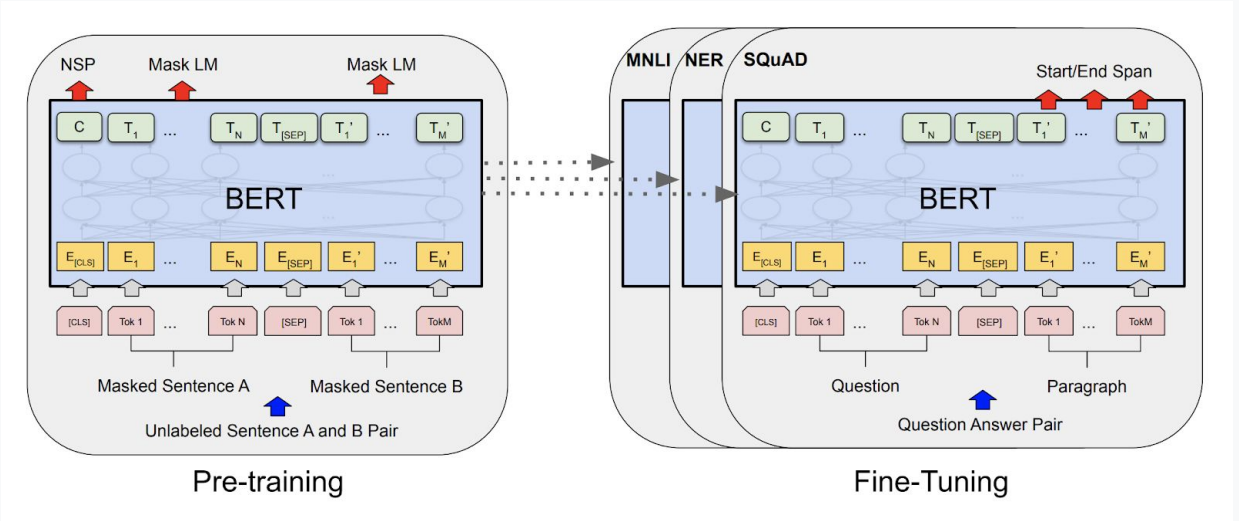


Thomas Wolf, Hugging Face Inc

BERT

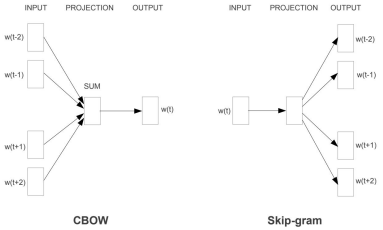
- Reusing model architecture
- Leveraging pretrained weights for initialization

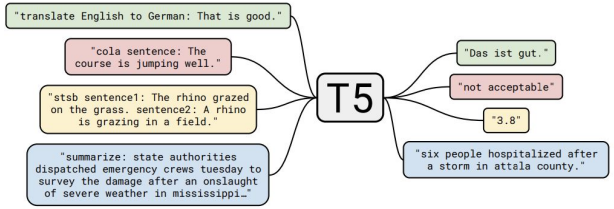
Fine-tuning



https://arxiv.org/pdf/1810.04805.pdf

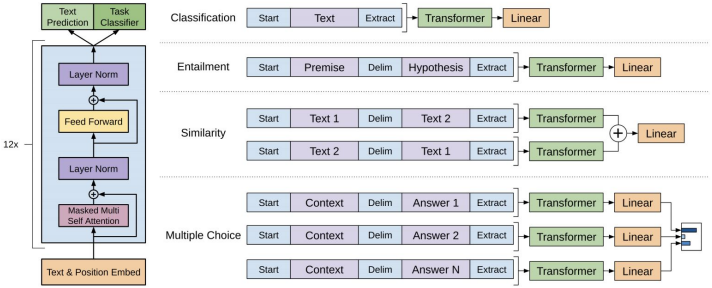
A brief history of pre-training in NLP





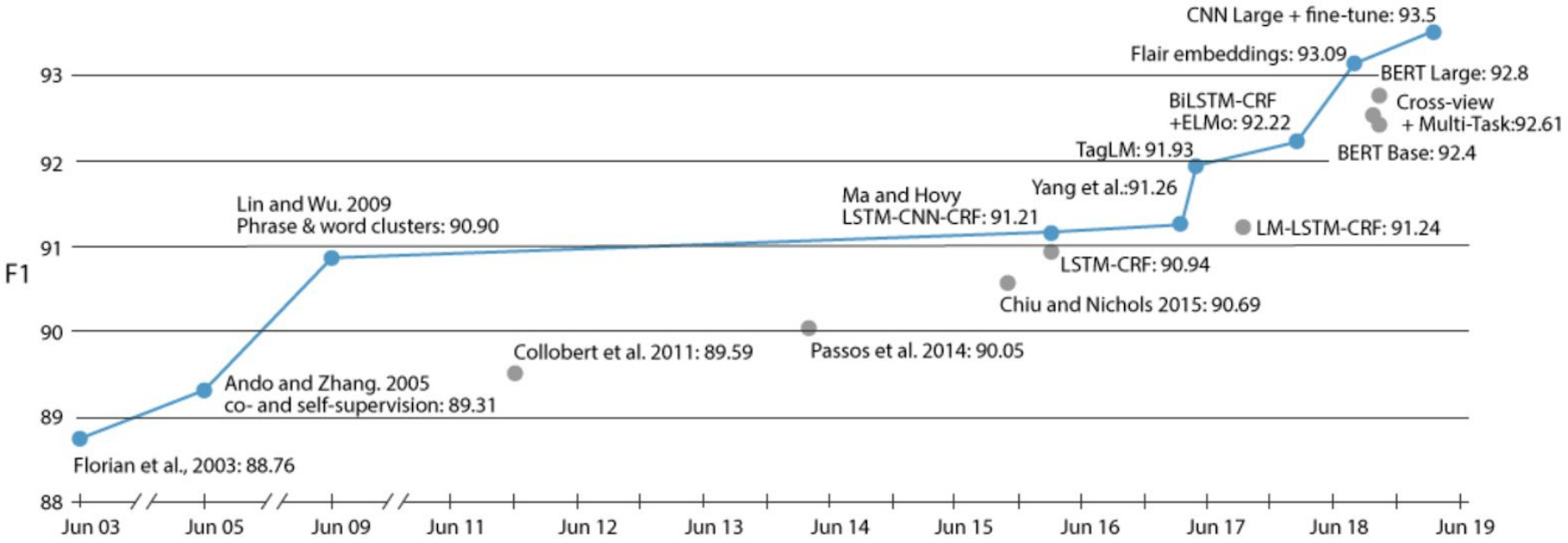
Word2Vec Mikolov et al., 2013
CoVe McCann et al., 2017
ELMo Peters et al., 2018
BERT Devlin et al., 2018
XLM Lample et al., 2019
RoBERTa Liu et al., 2019
T5 Raffel et al, 2019
GPT-3 Brown et al., 2020
DeepSeed 1T Majumder et al., 2020

Glove Pennington et al., 2014
ULMFiT Howard et al., 2018
GPT Radford et al., 2018
GPT-2 Radford et al., 2019
XLNet Yang et al., 2019
ALBERT Google AI Language, 2019
ELECTRA Clark et al., 2020
GShard Lepikhin et al., 2020
...



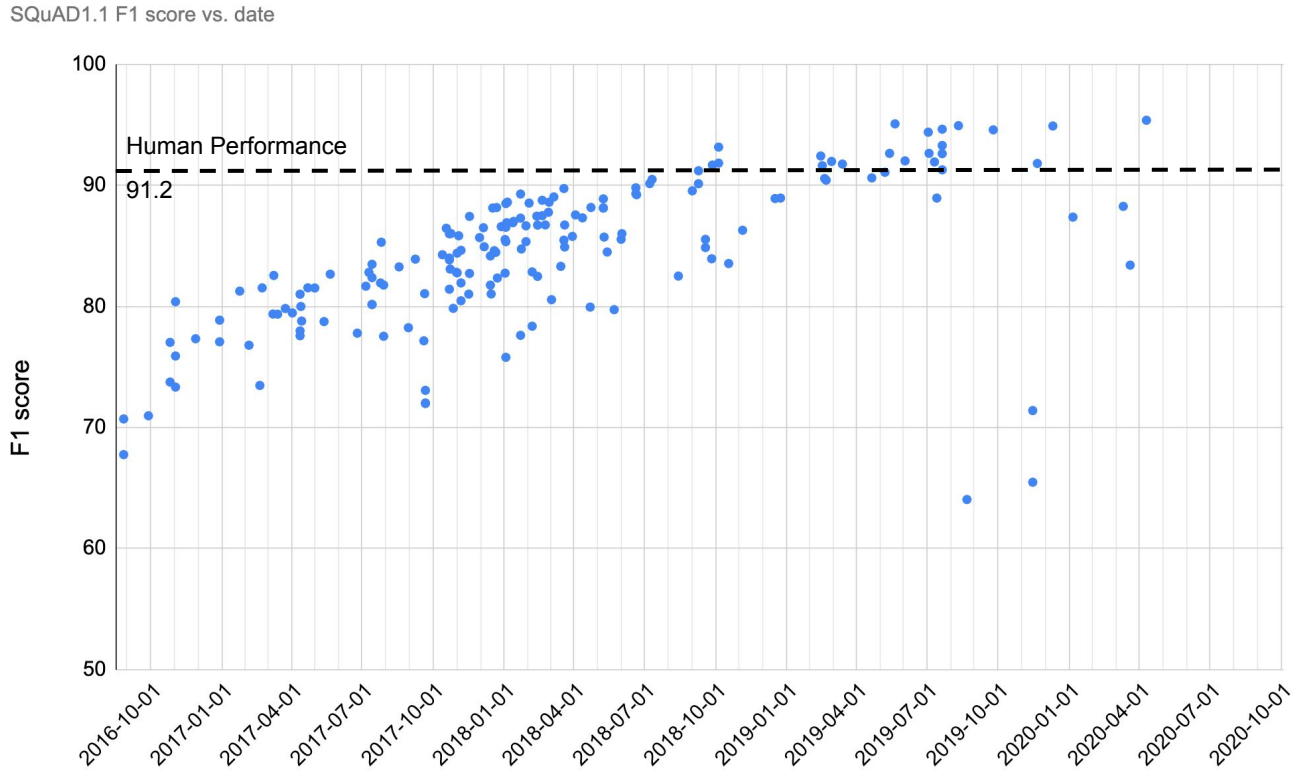
Why transfer learning in NLP (empirically)

Performance on Named Entity Recognition (NER) on CoNLL-2003 (English) over time



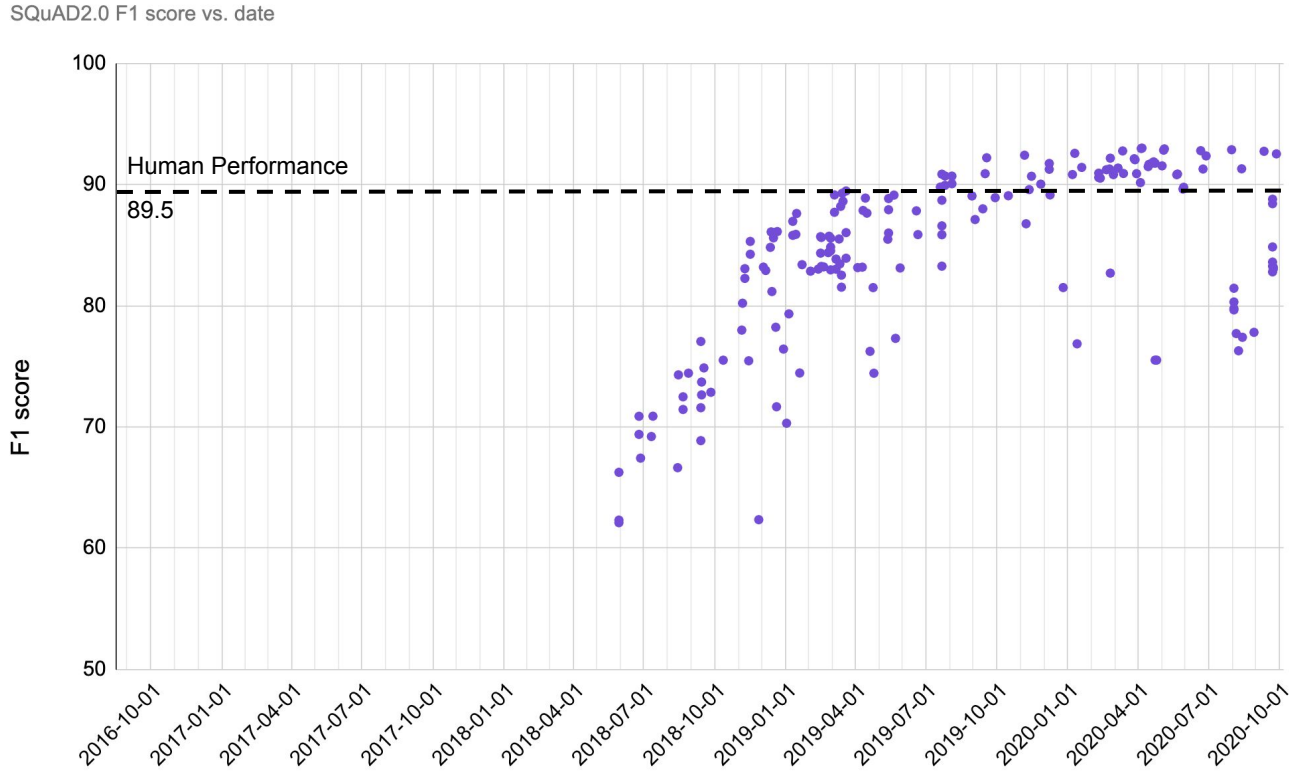
Thomas Wolf, Hugging Face Inc

Benchmarks through the years - SQuAD 1.1



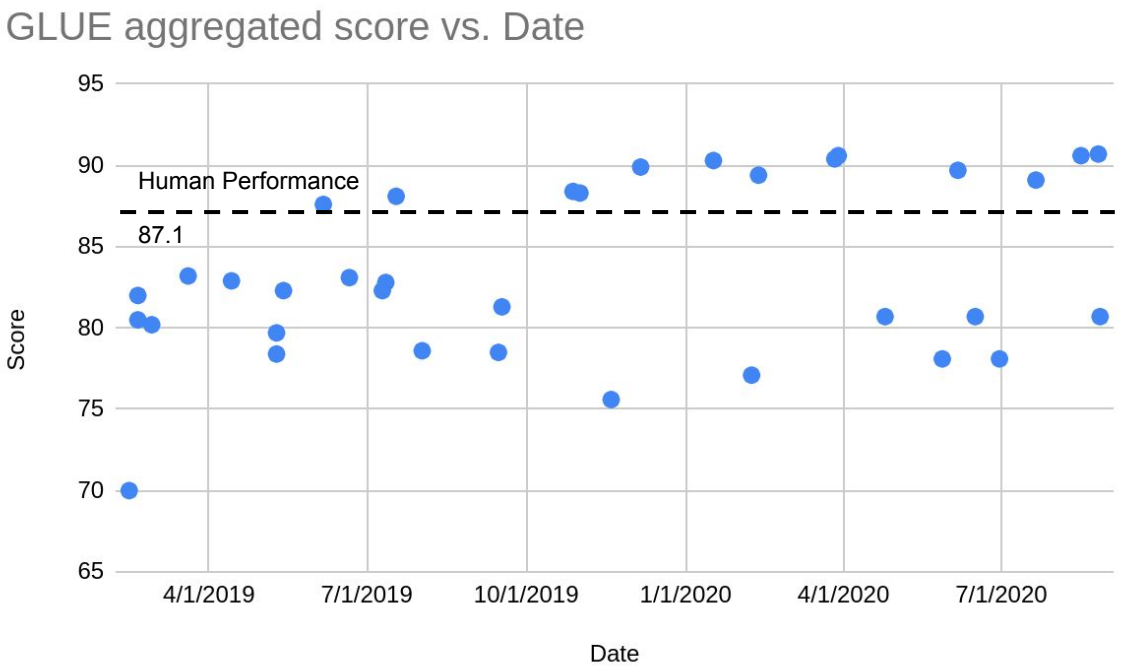
The Stanford Question Answering Dataset, https://rajpurkar.github.io/SQuAD-explorer/

Benchmarks through the years - SQuAD 2.0



The Stanford Question Answering Dataset, https://rajpurkar.github.io/SQuAD-explorer/

Benchmarks through the years - GLUE



The GLUE Benchmark (Wang et al., 2018)

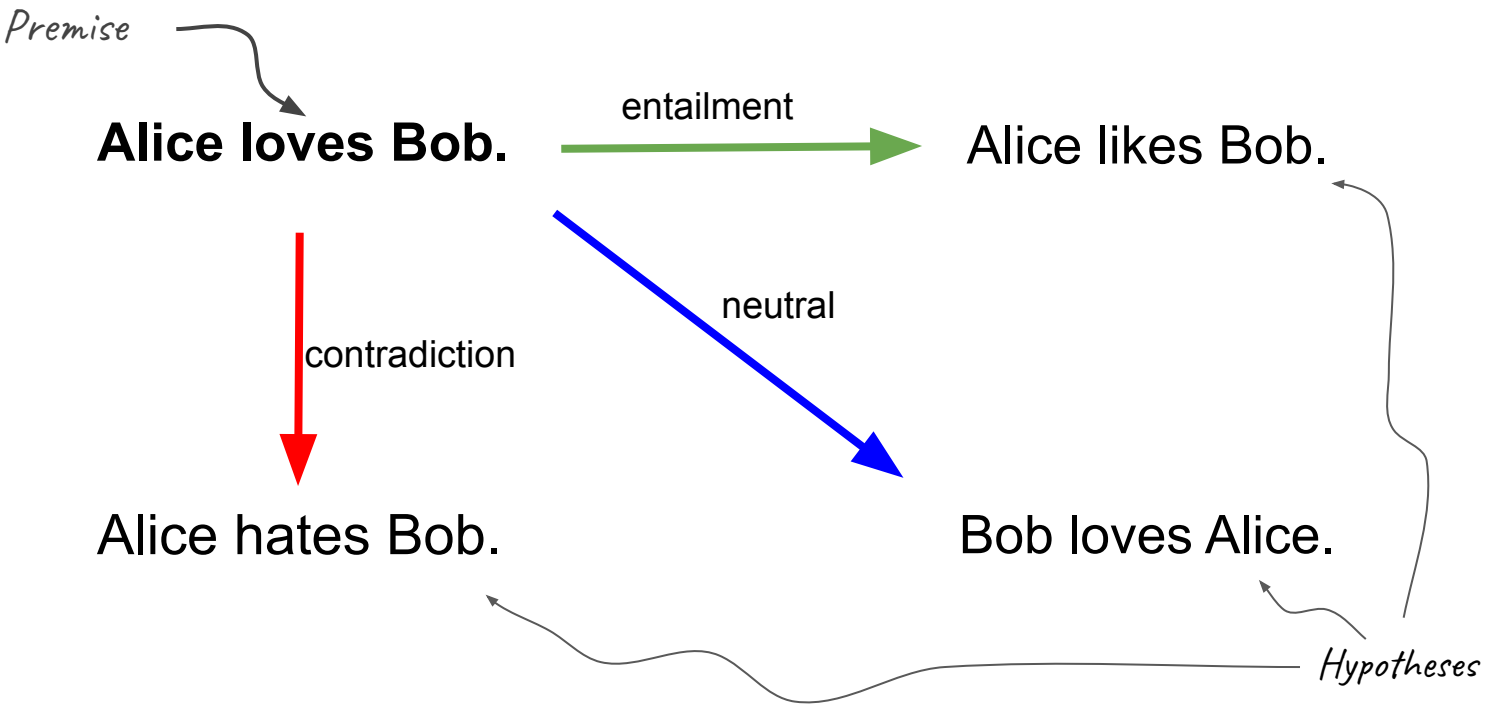
Colabs

Predicting Movie Reviews with BERT on TF Hub (fine-tuning on IMDb dataset)

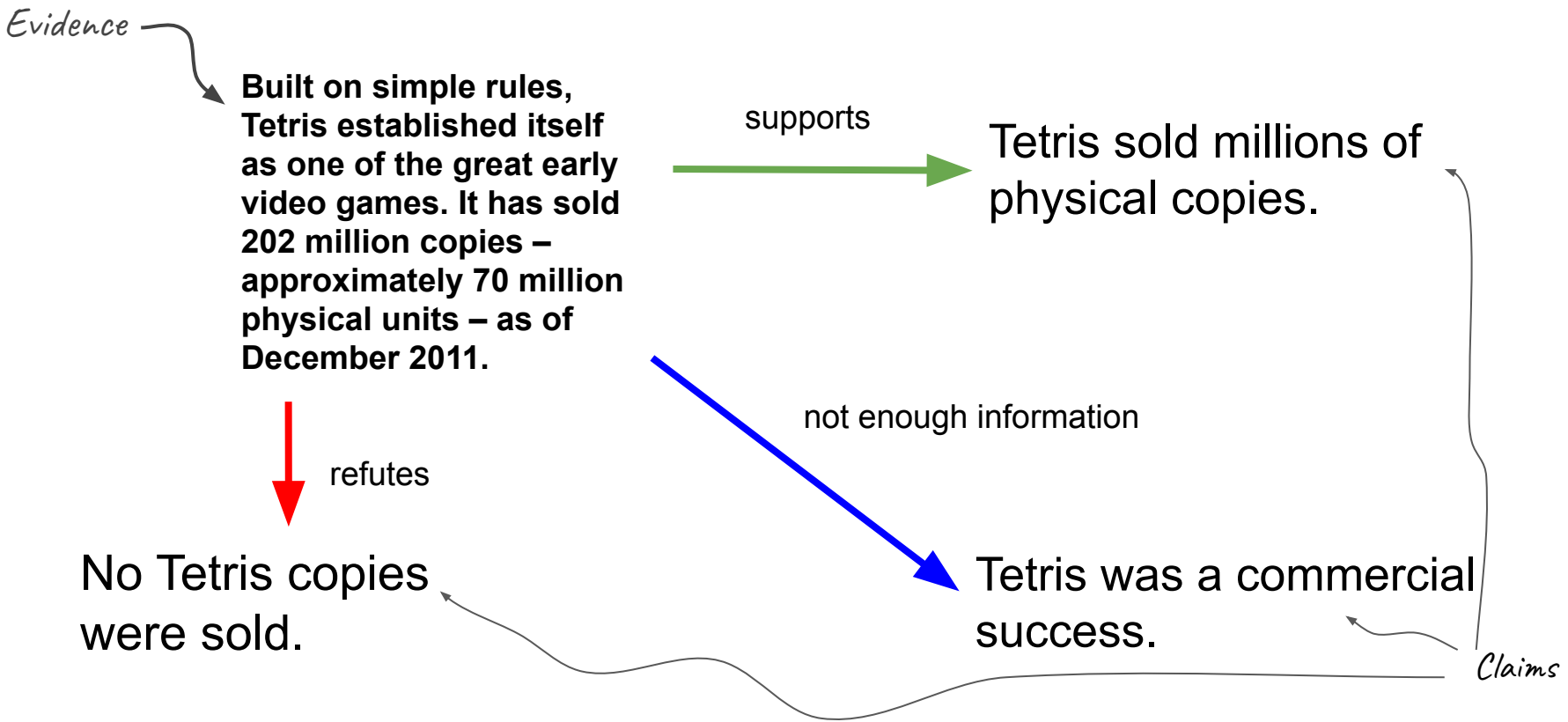
BERT End to End (Fine-tuning + Predicting) with Cloud TPU (MRPC, CoLA)

The textual entailment problem

Entailment: Natural language inference / entailment



Variant: Claim Validation



A solved problem?

Example: MultiNLI is part of GLUE

Rank	Name	Model	URL	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	HFL iFLYTEK	MacALBERT + DKM		90.7	74.8	97.0	94.5/92.6	92.8/92.6	74.7/90.6	91.3	91.1	97.8	92.0	94.5	52.6
2	Alibaba DAMO NLP	StructBERT + TAPT		90.6	75.3	97.3	93.9/91.9	93.2/92.7	74.8/91.0	90.9	90.7	97.4	91.2	94.5	49.1
3	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6	73.5	97.2	94.0/92.0	93.0/92.4	76.1/91.0	91.6	91.3	97.5	91.7	94.5	51.2
4	ERNIE Team - Baidu	ERNIE		90.4	74.4	97.5	93.5/91.4	93.0/92.6	75.2/90.9	91.4	91.0	96.6	90.9	94.5	51.7
5	T5 Team - Google	T5		90.3	71.6	97.5	92.8/90.4	93.1/92.8	75.1/90.6	92.2	91.9	96.9	92.8	94.5	53.1
6	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9	69.5	97.5	93.7/91.6	92.9/92.5	73.9/90.2	91.0	90.8	99.2	89.7	94.5	50.2
7	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7	70.5	97.5	93.4/91.2	92.6/92.3	75.4/90.7	91.4	91.1	95.8	90.0	94.5	51.6
8	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4	71.7	97.1	93.1/90.7	92.9/92.5	75.6/90.8	91.3	90.8	95.8	89.8	91.8	50.7
9	Huawei Noah's Ark Lab	NEZHA-Large		89.1	69.9	97.3	93.3/91.0	92.4/91.9	74.2/90.6	91.0	90.7	95.7	88.7	93.2	47.9
10	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)		88.4	68.0	96.8	93.1/90.8	92.3/92.1	74.8/90.3	91.1	90.7	95.6	88.7	89.0	50.1
	GLUE Human Baselines	GLUE Human Baselines		87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-

What do actual NLI examples look like (MultiNLI)?

entailment

"When I found out that the infant mortality rate in Indiana was 86 percent greater than for mothers in Sweden, I was shocked."

"The news was a huge surprise to me."

neutral

"Are they, like, rounded?"

"Are they rounded up?"

contradiction

"All that day, we didn't get up (off the floor)."

"We had fallen on the floor because the floor was wet."

Or what do actual claims look like (FEVER)?

Supported

- Woody Allen is a person.
- The Shining was directed.

Not enough information

- International Relations includes animals.
- Lisa Kudrow was in a car.

Refuted

- Toy Story is incapable of being a film.
- Tipper Gore was created in 1048.

Is the premise/evidence needed?

It was observed that models perform surprisingly well without the premise / evidence (e.g. SNLI, MNLI, FEVER).

Analyses show that the data often contains artifacts (e.g. negation is correlated with a contradiction/refutes label):

- 67% of SNLI and 53% of MultiNLI (55% in [2]) can be decided correctly by a simple text categorization model [1].
- A claim-only BERT model achieves 61.7% accuracy on FEVER [3].

[1] Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, Noah A. Smith: Annotation Artifacts in Natural Language Inference Data
[2] Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, Benjamin Van Durme: Hypothesis Only Baselines in Natural Language Inference
[3]Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, Regina Barzilay: Towards Debiasing Fact Verification Models

03

Structured Data

Semi-structured and tabular text

Semi-structured and tabular data

Motivation

- 40–50% of the content on the web is template content (Gibson et al., 2005)
- Includes various domains and languages
- Richer than free text
- Easier to extract information from than free text

Semi-structured and tabular data

Challenges

- Diverse
  - layout / format
  - terms
  - languages
  - domains
  - entities
- Lack of training data for supervised learning
- Mixed textual structure (unstructured, semi-structured, tables)

Semi-structured and tabular data

Structured data from free text

- Extract triple relationships (subject, predicate, object)
- Handle word and entity synonyms
- Handle alternative phrasings
- Find agreement among articles

Knowledge graphs

Knowledge Graphs

Motivation

Knowledge graphs are critical to many enterprises today: They provide the structured data and factual knowledge that drive many products and make them more intelligent and "magical."

Noy et al., 2019

Example of a Knowledge Graph

Entity:

- Name : American Revolution
- Type: /time/event
- Property: ...

Videos

The American Revolution - OverSimplified (Part 1)
OverSimplified
YouTube - Aug 30, 2018

The American Revolution: 1775 (video)
The Aspen Institute
Khan Academy - Sep 27, 2014

The American Revolution - OverSimplified (Part 2)
OverSimplified
YouTube - Aug 30, 2018

American Revolution - Wikipedia
https://en.wikipedia.org/wiki/American_Revolution
The **American Revolution** was a colonial revolt that took place between 1765 and 1783. The American Patriots in the Thirteen Colonies won independence from ...
American Revolutionary War · Patriot (American Revolution) · Gaspee Affair

Revolutionary War - HISTORY
https://www.history.com/topics/american-revolution/american-revolution-history
Sep 11, 2018 - For more than a decade before the outbreak of the **American Revolution** in 1775, tensions had been building between colonists and the British ...

American Revolution: Causes and Timeline | HISTORY.com - HISTORY
https://www.history.com/topics/american-revolution
The **Revolutionary** War waged by the **American** colonies against Britain influenced political ideas and **revolutions** around the globe, as a small fledgling nation won its freedom from the greatest military force of its time. ... Battle of Bunker Hill. ... Writing of Declaration of Independence.

American Revolution | Causes, Battles, Aftermath, & Facts | Britannica ...
https://www.britannica.com/event/American-Revolution
American Revolution, also called United States War of Independence or **American Revolutionary** War, (1775–83), insurrection by which 13 of Great Britain's ...

American Revolution

The American Revolution was a colonial revolt that took place between 1765 and 1783. The American Patriots in the Thirteen Colonies won independence from Great Britain, becoming the United States of America. They defeated the British in the American Revolutionary War in alliance with France. Wikipedia

Period: 1765 – 1783

Location: Thirteen Colonies

Participants: Colonists in British America

Results: Independence of the United States of America from the British Empire, MORE

People also search for View 10+ more


American Civil War


World War I


World War II


Cold War


French Revolution

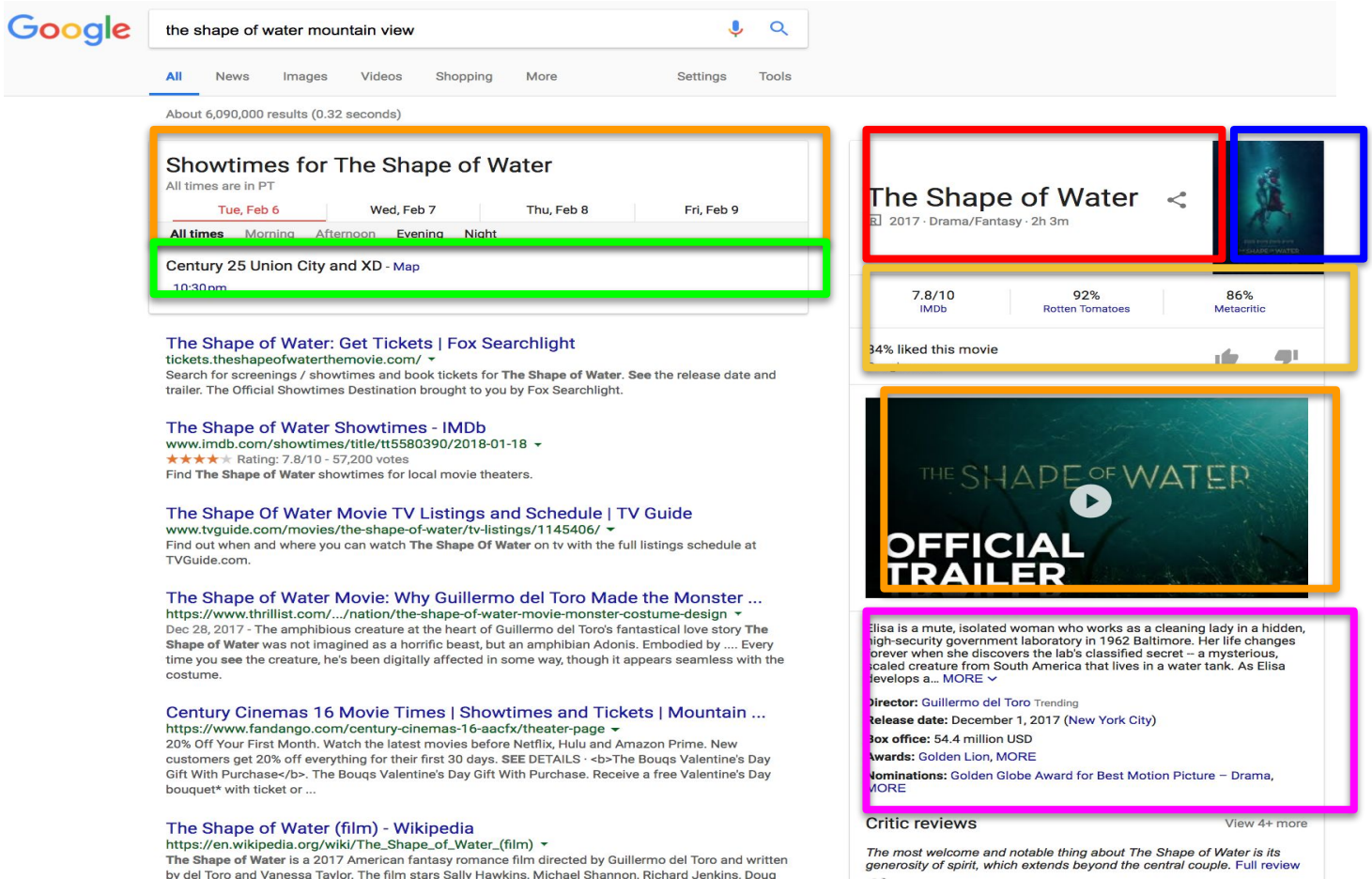
Feedback

See results about

American Revolutionary War
The American Revolutionary War, also known as the American War of Independence, was an 18th-...

[Muxagata et al., 2019]

Application example



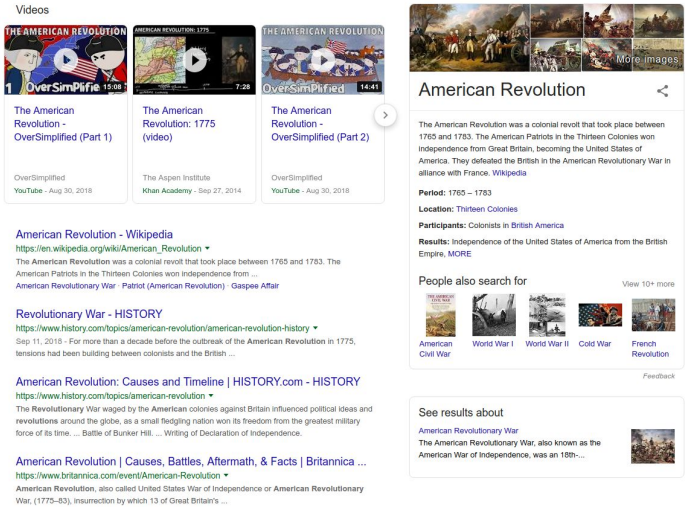
Muxagata et al., 2019

Knowledge Graphs

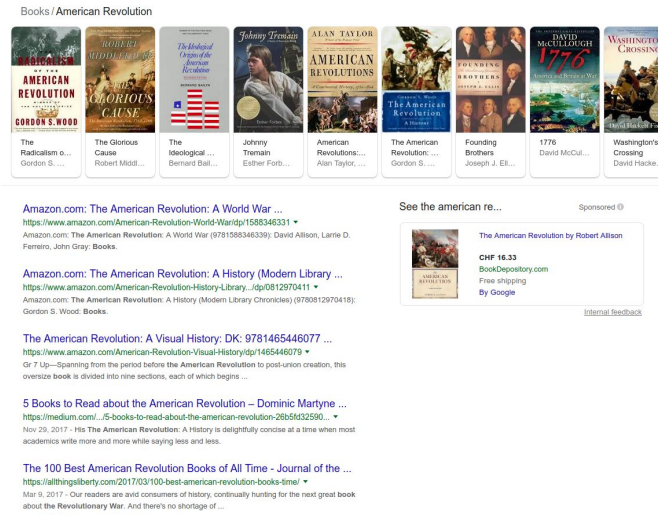
Challenges

- Diversity in languages, domains, etc
- Long tail knowledge
- Inconsistencies across sources (and reconciliation)

Example of identity problem



?



Entity1:

- Name : American Revolution
- Type: /time/event
- Property: ...

Entity2:

- Name : American Revolution
- Type: /business/product_line
- Property: ...

Muxagata et al., 2019

Knowledge Graphs

How

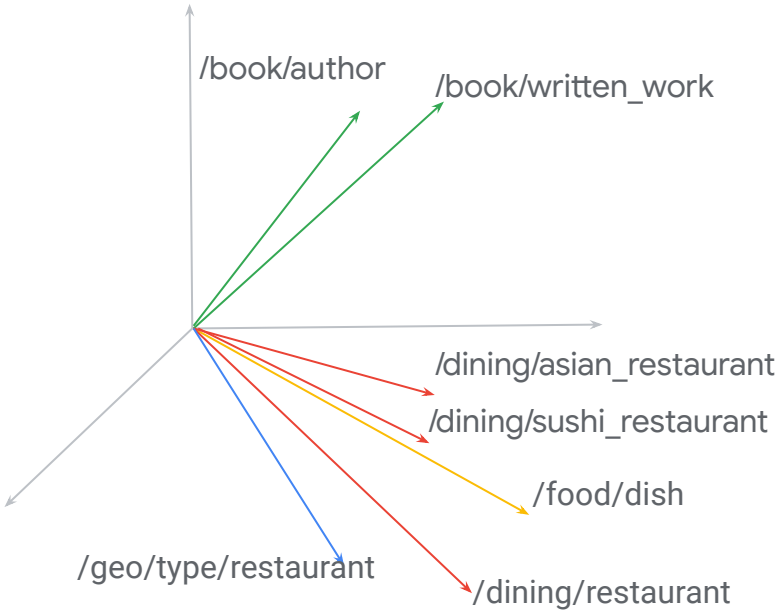
- Build a long-term, stable source of class and entity identities.
- Many practical implementations impose constraints on the links in knowledge graphs by defining a schema or ontology.
- To ensure a large-scale system remains consistent over time, it can be built from a basic set of low-level structures.
- Replicate similar structures and reasoning mechanisms at different levels of abstraction.

Noy et al., 2019

Semantic Type Embeddings

Benefits:

- Schema reconciliation
- Cross-verticals relationships
- Domain clustering of types
- Ontology scaling



Muxagata et al., 2019

Goal: A Robust Type Set similarity metric

Task: Compare two type sets and score their similarity

Set A: /common/topic, /people/person, /book/author, /government/us_president

Set B: /common/topic, /entwild/people, /book/author, /government/politician

Similarity(A, B) ?

Baseline approach:

- Pointwise mutual information
- Co-occurrences of types
- Conflicting types

Issues:

- Type which did not co-occur in the training set (no transitivity)
- Sensitivity to outlier types
- Targeted heuristics which developed over time

Muxagata et al., 2019

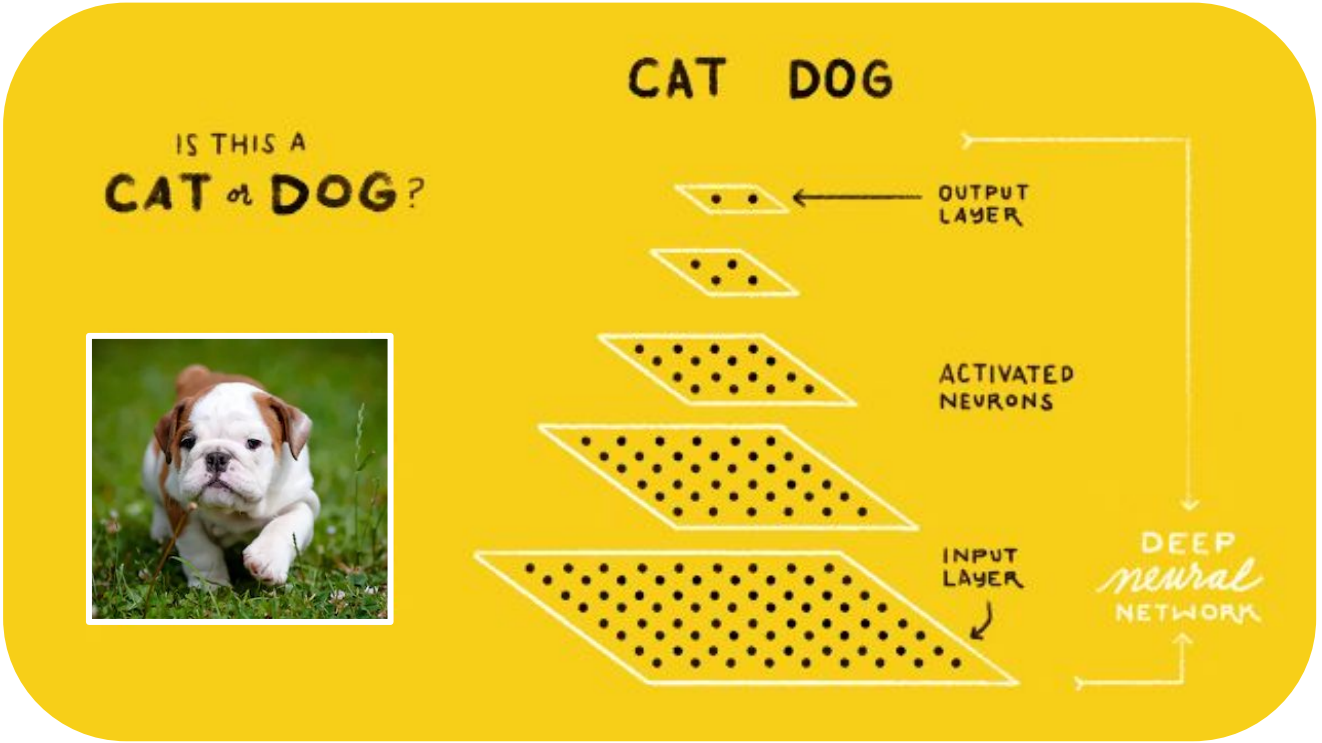
04

Neural Graph Learning

Leveraging structured signals with Neural Structured Learning

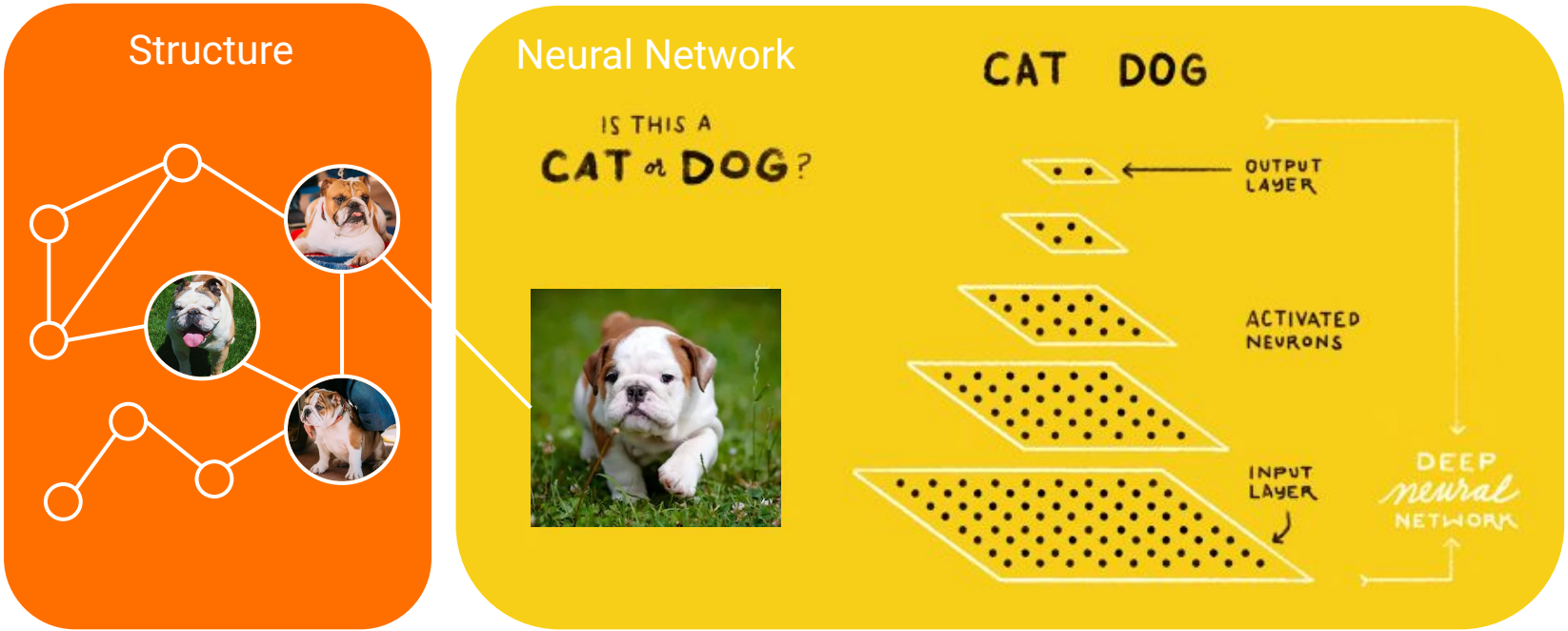
tensorflow.org/neural_structured_learning

How a Typical Neural Net Works



Gopalan et al, 2020.

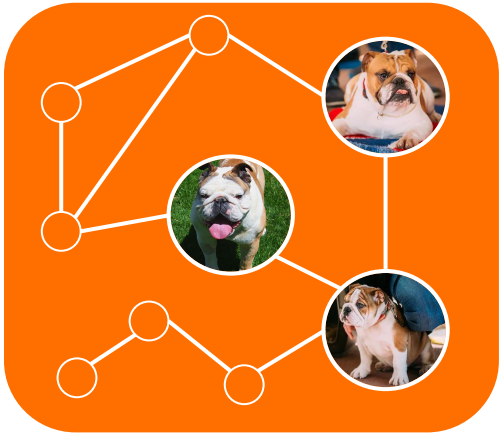
Neural Structured Learning (NSL)



Gopalan et al, 2020.

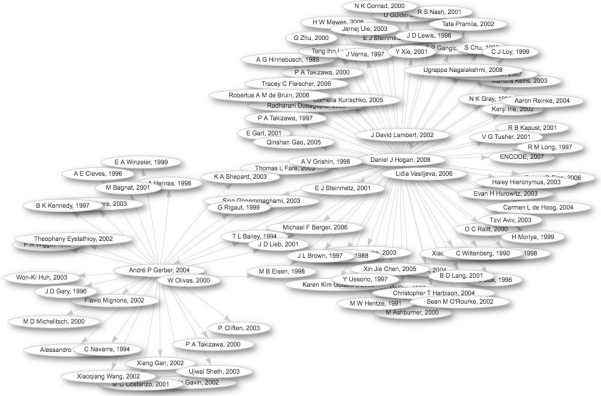
Structure Among Samples

Co-Occurrence Graph



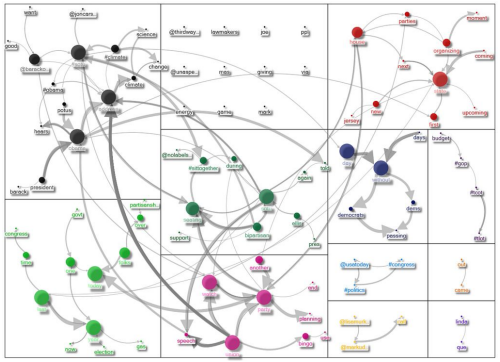
[Source: graph concept is from Juan et al., WSDM'20. Original images are from pixabay.com]

Citation Graph



[Source: https://commons.wikimedia.org/wiki/File:Partial_citation_graph_for_%22A_screen_for_RNA-binding_proteins_in_yeast_indicates_dual_functions_for_many_enzymes%22_as_of_April_12,_2017.png]

Text Graph



[Source: copied without modification from https://www.flickr.com/photos/marc_smith/6705382867/sizes/l/]

Gopalan et al, 2020.

NSL: Advantages of Learning with Structure

- **Less Labeled Data Required**
- **More Robust Model**

Gopalan et al, 2020.

Scenario I: Not Enough Labeled Data

Example task:

Document Classification

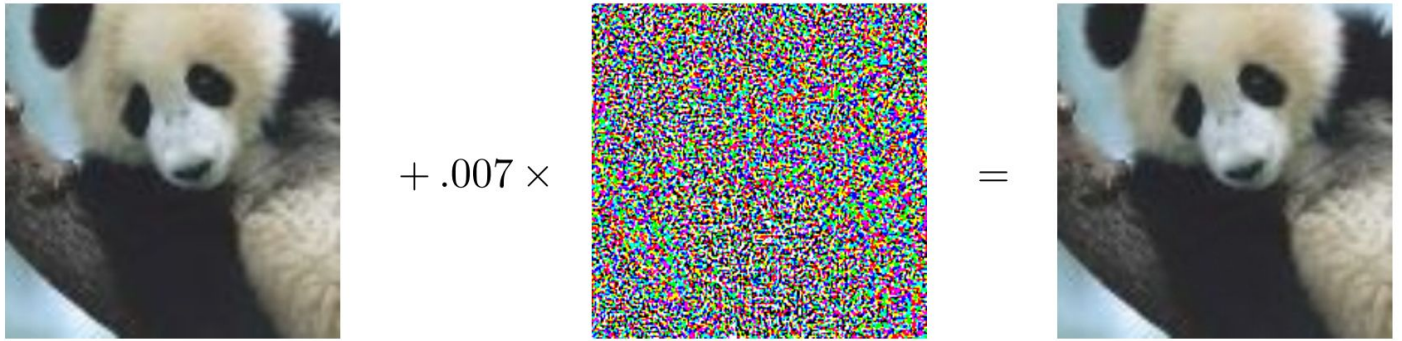
Lots of samples

Not enough labels

Gopalan et al, 2020.

Scenario II: Model Robustness Required

Example task: **Image Classification**

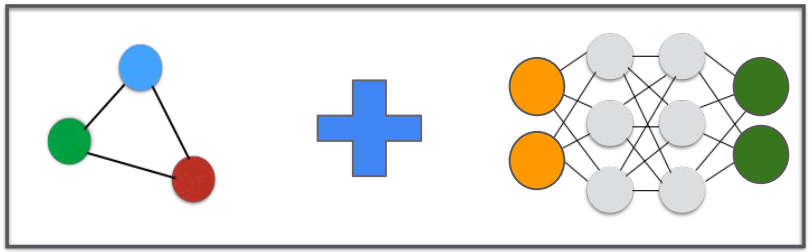


[Source: Goodfellow, et al., ICLR'15]

Gopalan et al, 2020.

NSL: Neural Graph Learning

Graph + Neural Net

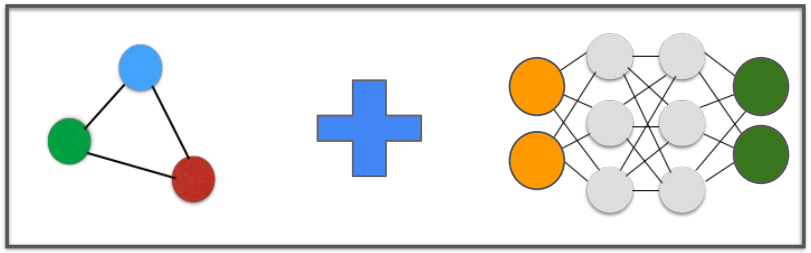


- **Jointly** optimizes both features & structured signals for better models

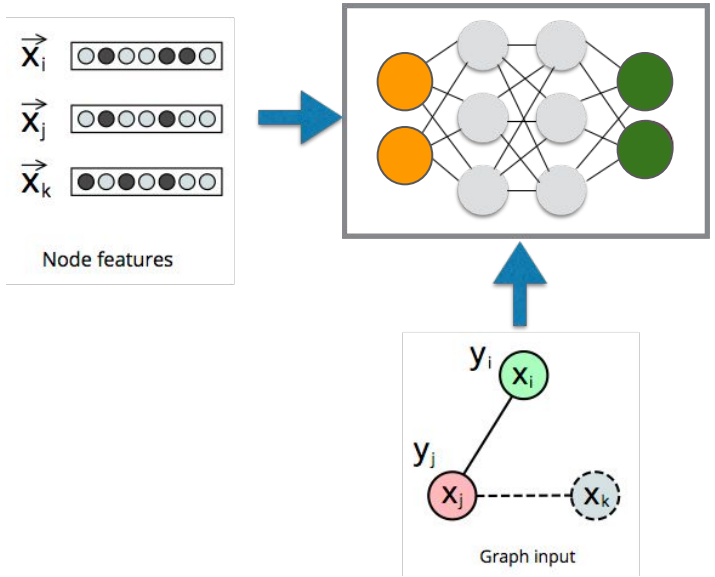
Gopalan et al, 2020.

NSL: Neural Graph Learning

Graph + Neural Net



- **Jointly** optimizes both features & structured signals for better models



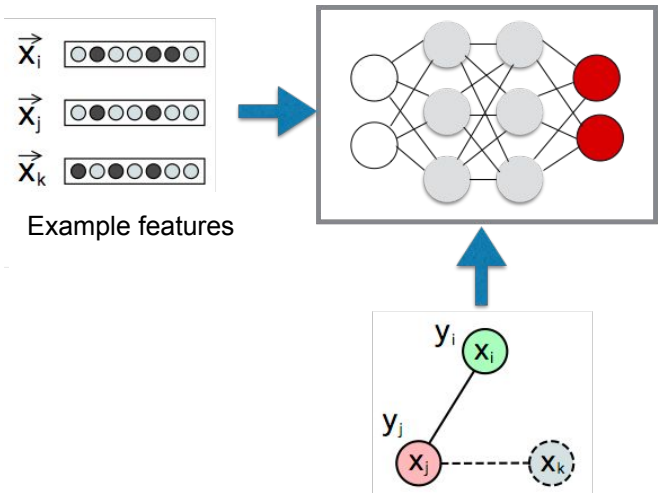
Neural Graph Machines (NGM)

Paper: Bui, Ravi & Ramavajjala [WSDM'18]

Gopalan et al, 2020.

NSL: Neural Graph Learning

Joint optimization with label and structured signals:



Optimize: $loss = \sum_{i=1}^{B} \mathcal{L}(y_i, \hat{y}_i) + \alpha \sum_{i=1}^{B} \mathcal{L}_{\mathcal{N}}(y_i, x_i, \mathcal{N}(x_i))$

$x_i \rightarrow f(.) \rightarrow y_i$

Supervised Loss

$$\sum_{i=1}^{B} \mathcal{E}(y_i, g_\theta(x_i))$$

Neighbor Loss

$$\sum_{x_j \in \mathcal{N}(x_i)} w_{ij} \cdot \mathcal{D}(h_\theta(x_i), h_\theta(x_j))$$

$g_\theta(x_i)$: NN output for input x_i

$\mathcal{E}(.)$: Loss function

Examples: L2 (for regression)
Cross-Entropy (for classification)

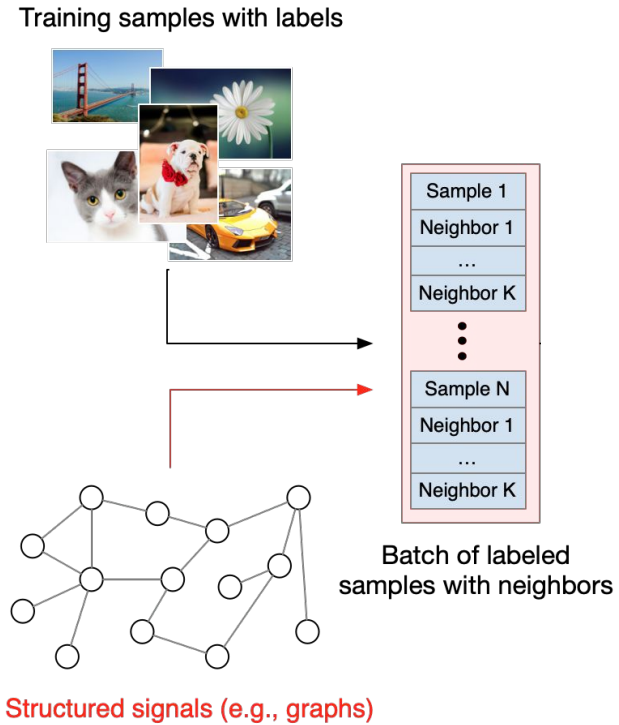
$h_\theta(.)$: Target hidden layer

$\mathcal{D}(.)$: Distance metric

Examples: L1, L2, ...

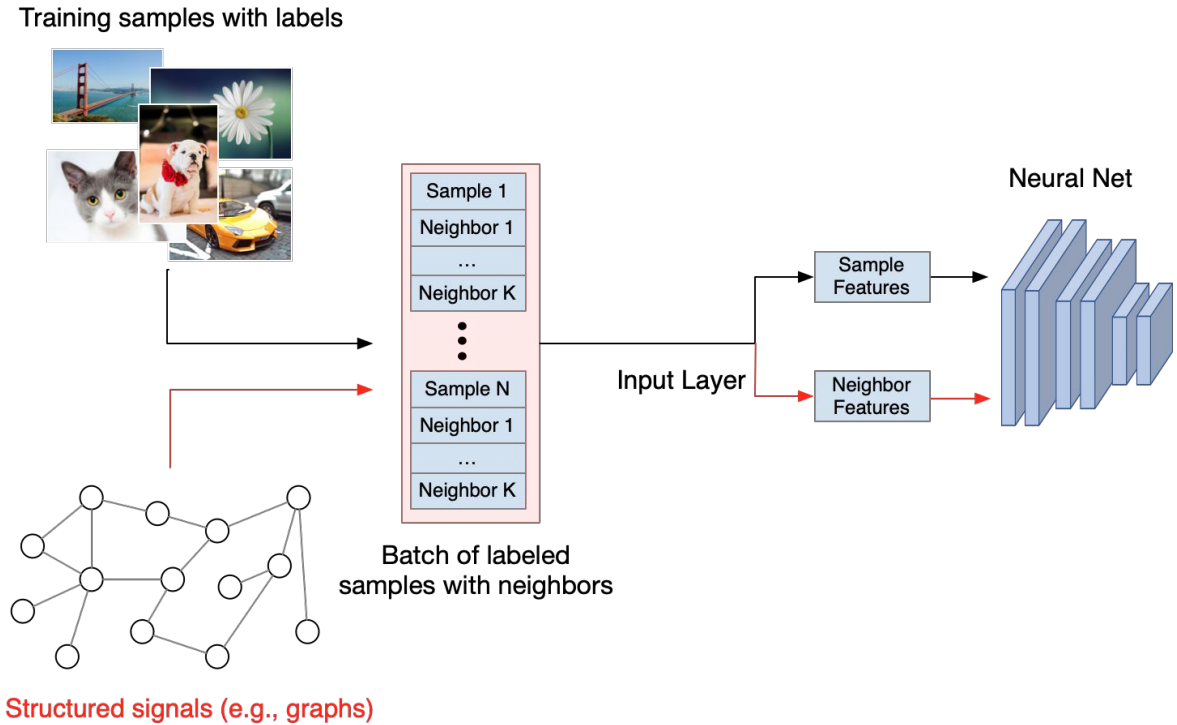
Gopalan et al, 2020.

NSL: Neural Graph Learning in Practice



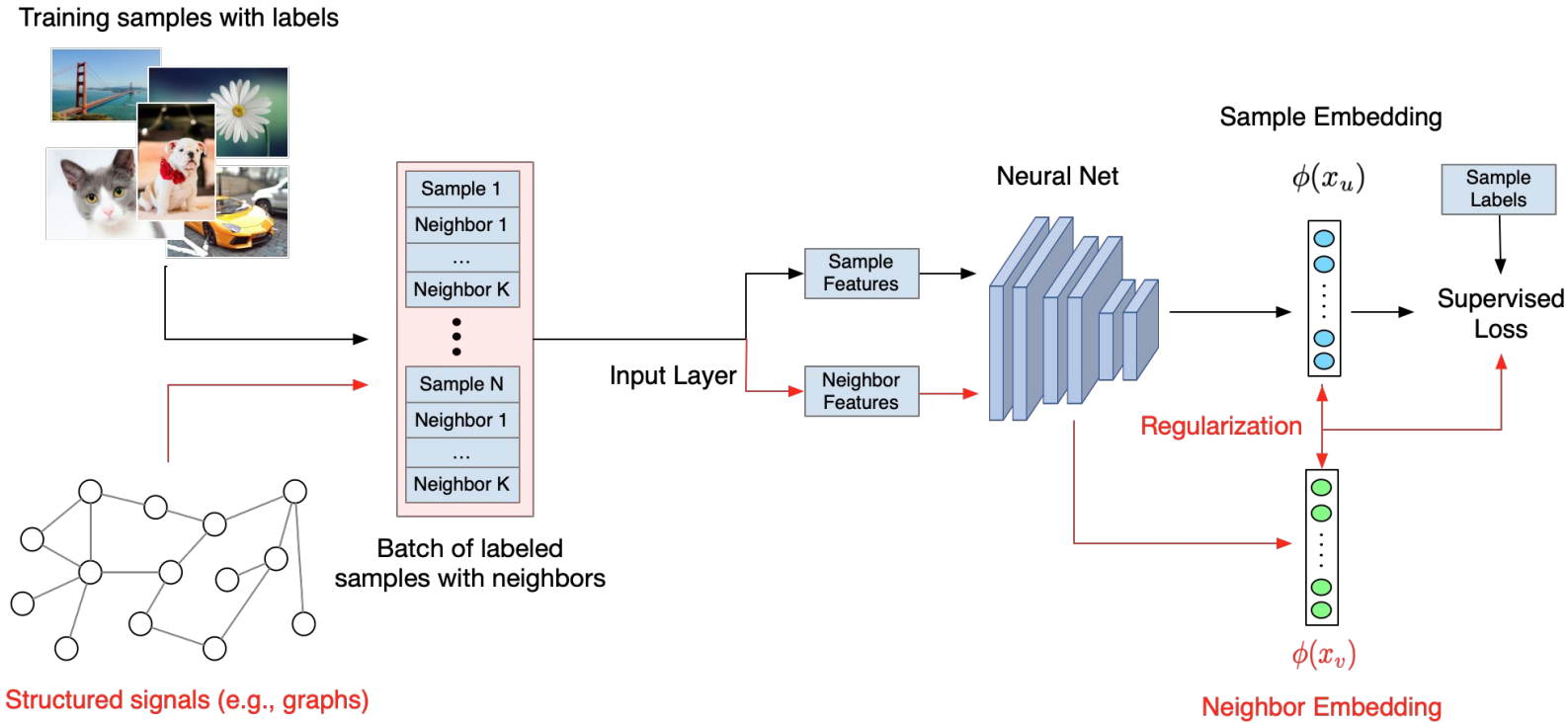
Gopalan et al, 2020.

NSL: Neural Graph Learning in Practice



Gopalan et al, 2020.

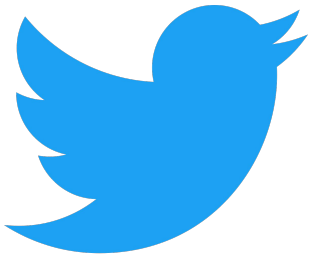
NSL: Neural Graph Learning in Practice

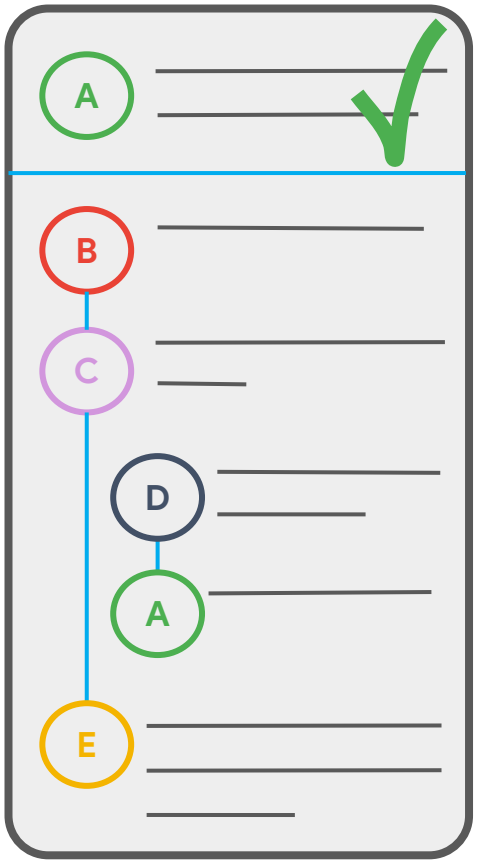


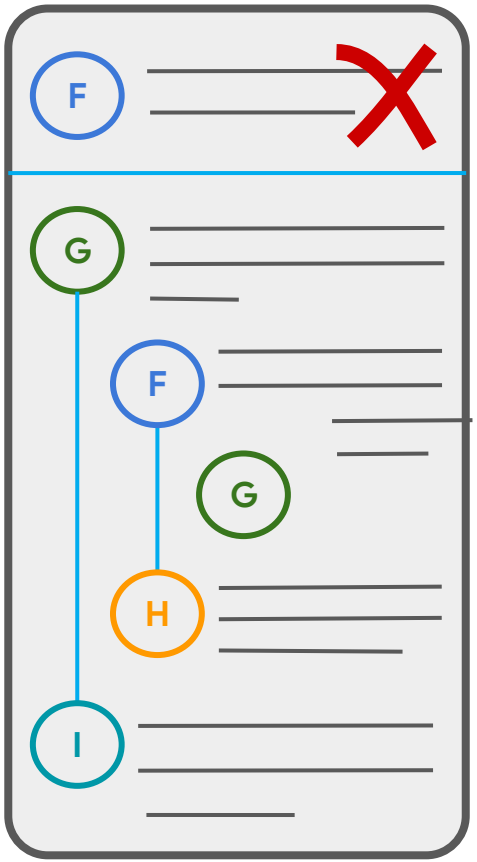
Gopalan et al, 2020.

Example

Structured reply threads as a natural graph



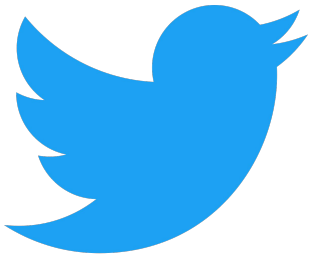




Example

PHEME dataset for Veracity Classification

Structured rumour threads with veracity labels



This dataset contains a collection of Twitter rumours posted during breaking news, and rumour veracities are labeled by professional journalists

Derczynski et al

Kochkina et al

Colab: Graph regularization for Twitter rumour veracity classification

NSL Resources

Website: tensorflow.org/neural_structured_learning

GitHub: github.com/tensorflow/neural-structured-learning

05

Computer Vision

Foundations of Computer Vision

Outline

- What is Computer Vision?
- Gaining intuition about convolutions
- Inductive biases
- Computer Vision Progress
- Recent trends in Computer Vision

What is Computer Vision?

The term has changed throughout the years.

Ballard and Brown (1982): The construction of explicit, meaningful description of physical objects from images.

Trucco and Verri (1998): Computing properties of the 3D world from one or more digital images.

Stockman and Shapiro (2001): To make useful decisions about real physical objects and scenes based on sensed images.

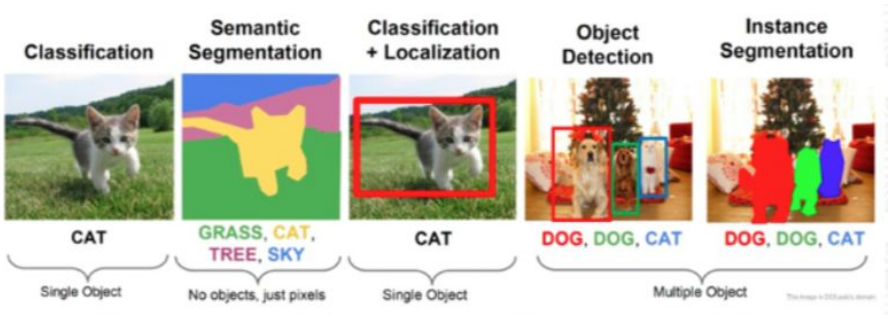
Forsyth and Ponce (2003): extracting descriptions of the world from pictures or sequences of pictures.

Interdisciplinary field which deals with image and video understanding

What is Computer Vision?

Popular computer vision tasks

- Image classification
- Image Segmentation
- Classification + Localization
- Object Detection
- Instance Segmentation
- Pose Estimation
- Scene Reconstruction
- Content-based Image Retrieval
- Optical Character Recognition (OCR)
- ...

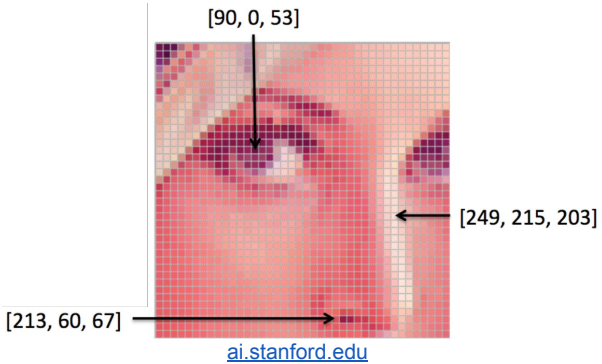


cs231n.stanford.edu

Gaining intuition about convolutions

When dealing with images we start from a high-dimensional space of pixel values

Typically, working with RGB images of 256x256 pixels, our input image $x \in \mathbb{R}^{3\times256\times256}$



When the dimensionality of the input is that *high* many statistical challenges arise due the *curse of dimensionality*

Gaining intuition about convolutions

Supervised learning is a high-dimensional interpolation problem. The objective is often to approximate a function $f(x)$ using K training points $(x_i, y_i)_{i \in [1...K]}$

Why previous classical machine learning algorithms don't do well in a such high-dimensional Computer Vision problem?

They rely on two basic assumptions:

1. local constancy of the approximate function
2. smoothness regularization

Going beyond *local template matching* and *local generalization* comes from training deep learning models

Gaining intuition about convolutions

Before introducing successful deep learning architectures, let's gain some intuition in the MNIST dataset.

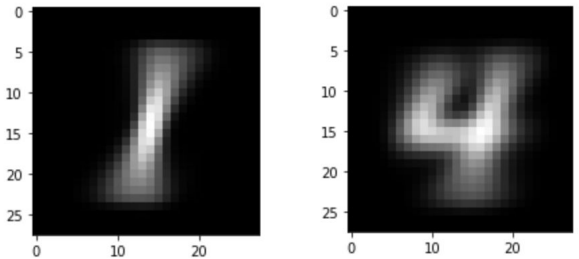
MNIST dataset: handwritten digits in grayscale 28x28 images.

Having a labelled dataset, can you think about a simple algorithm to classify digits?

Gaining intuition about convolutions

Solution 1:

- Flatten the input image, 2d grid, into a vector $x \in \mathbb{R}^{784}$
- Compute a canonical representation for each class (digit)
  - average of vectors for each class (centroid)



- Classify an unseen digit by choosing the class with the lowest L2-norm from test example to canonical representation

Gaining intuition about convolutions

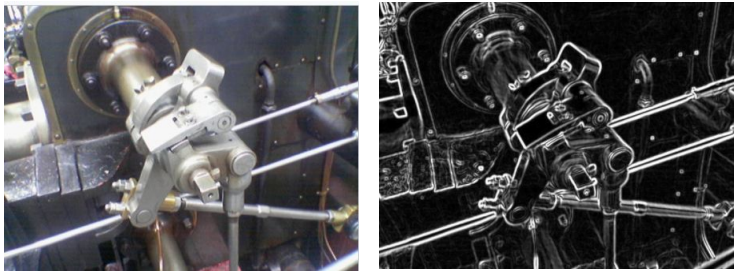
Can we do better than using a canonical digit representation?

Yes, we can use feature extraction techniques to get a better digit representation.

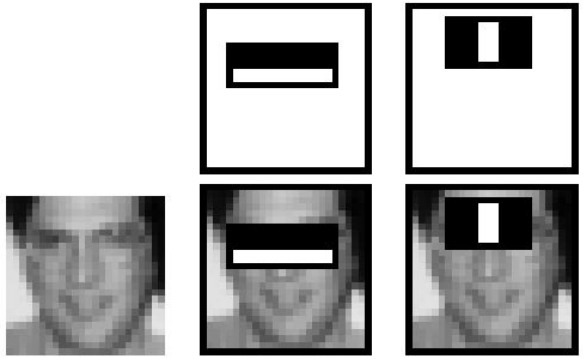
We can use filters!

Gaining intuition about convolutions

Filters are widely used in signal processing and image processing techniques.



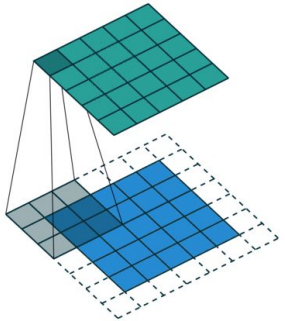
Sobel operator: edge detection



Viola-Jones face detection using Haar-like features

We can look to an image through the lens of a filter: feature map.

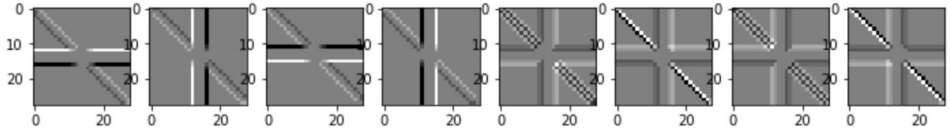
Filters can highlight a particular aspect of the image



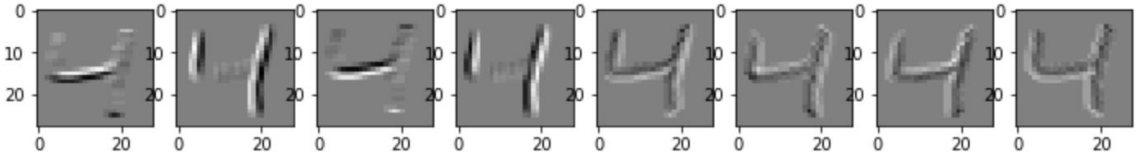
Gaining intuition about convolutions

Solution 2:

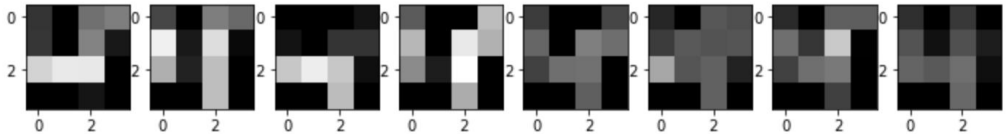
Define a fixed bank of filters:



Apply to an image :



Use a dimensionality reduction technique, such as max pooling to get our digit signatures



Finally, to classify an unseen digit, compare its signature based on convolutions to the signature / feature representation of each class.

We reduce our embedding size from 28x28 = 784 canonical representation to a 4x4x8 = 218 vector.

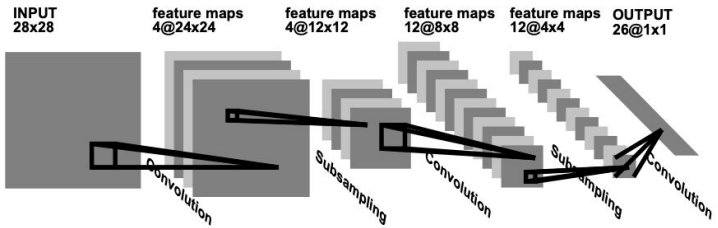
Colab: Deep Learning DIY

Gaining intuition about convolutions

Solution 3:

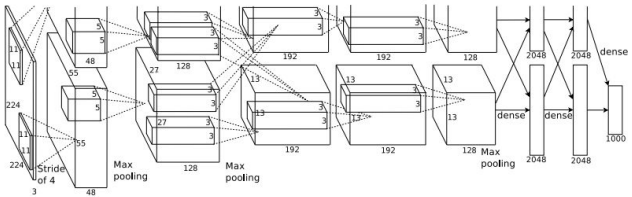
Can we do better? Yes, filters can be learned!

Inspired by classical work in Neuroscience by Hubel and Wiesel (1962) about the visual cortex which showed that brain neurons were organized in local receptive fields, Convolution Neural Networks were introduced in by Yann LeCun.



CNNs for Images, Speech & Time-series

A deeper architecture won ImageNet competition by a large margin in 2012. (10pp higher accuracy than previous SOTA)



AlexNet

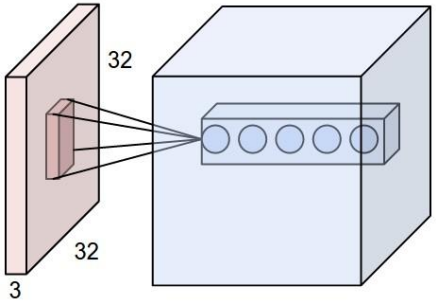
Gaining intuition about convolutions

CNNs: Composition of differentiable 3D-tensor transformations

convolution

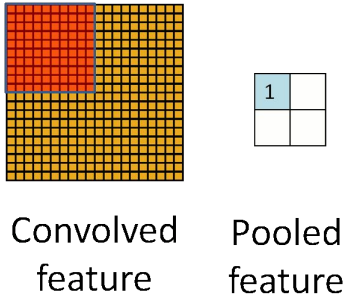
[1]

feature map stacking

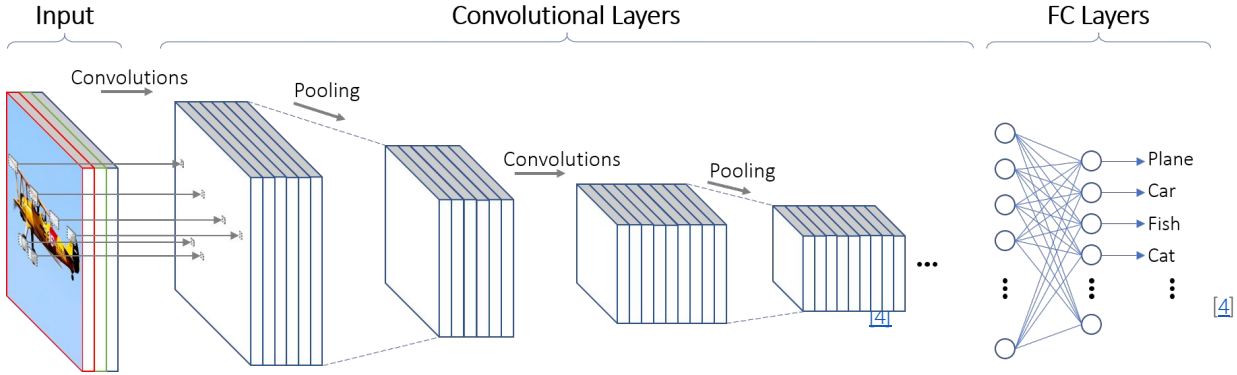


[2]

pooling



[3]



[4]

Gaining intuition about convolutions

What are the advantages of CNNs?

- Rely on the assumption that the data was generated by a composition of factors, or features, potentially interacting in multiple hierarchical levels
- Composition of layers implies a hierarchical processing bias: computations are performed in stages, allowing increasingly long-range interactions among information in the input image. Pooling layers increase the receptive field
- Use inductive biases to learn translation invariance functions. Avoid learning shift invariance from the data
- Weight sharing by learning small filters/kernels which are used in the entire feature map

CNNs effectively solved the curse of dimensionality

Inductive Biases

Google AI

Relational biases impose constraints on relationships and interactions among entities in a learning process."[1]

Relational biases can be viewed as a learning facilitator

Fully connected layers have week relational biases: allow all-to-all relationship

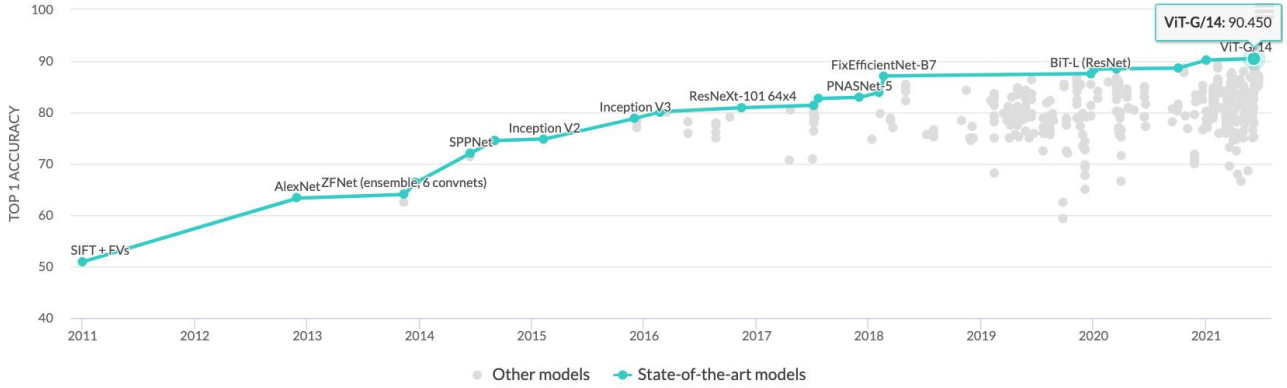
CNNs have sparser pixel interactions

- locality: only nearby pixels take part in the computation of convolution
- translation invariance: use the same filters across localities in the input signal

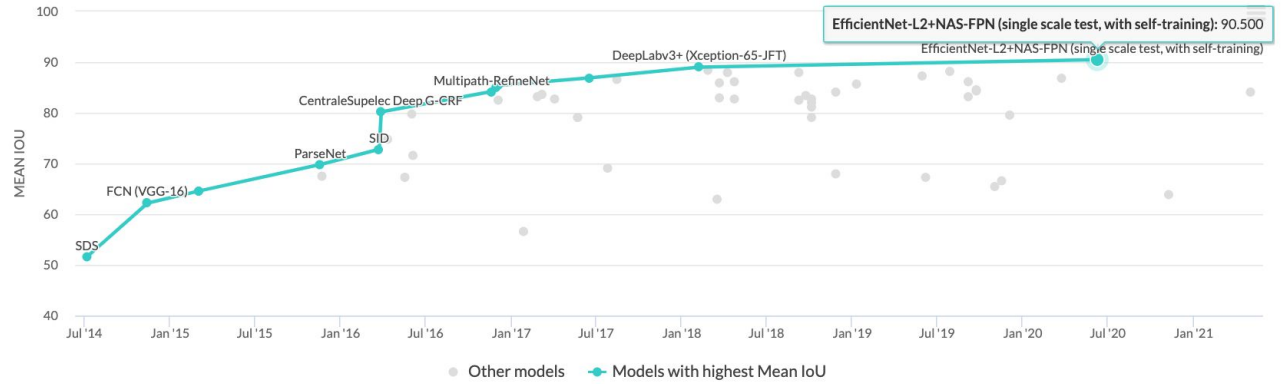
Transformer: permutation invariant using global attention leading to a more generic architecture

[1] Relational inductive biases, deep learning, and graph networks

Computer Vision Progress



ImageNet - Papers with code



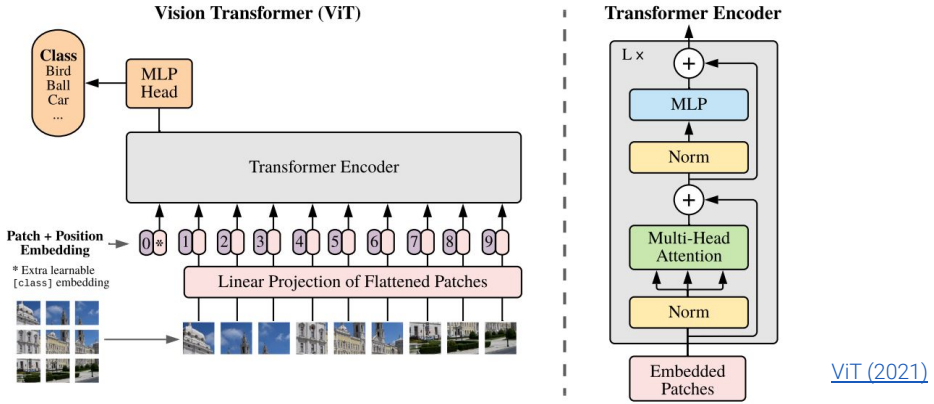
Pascal VOC 2012 - Papers with code

Recent Trends in Computer Vision

- **Transformer based architectures**
- **Self-supervised learning**
- Efficient-models
  - MobileNets, EfficientNet, RegNets
- Generative deep learning
  - GANs: thisxdoesnotexist, Implicit Neural Representation: CLIP
- Robust vision models

Transformer based-architectures

Is the inductive bias imposed by CNNs really necessary? Actually, no.



ViT (2021)

Can we be limiting model performance by imposing CNN constraints in the learning process?

ViT is scaling better than ResNets when pre-trained in larger datasets

Self-supervised Learning (SSL)

Strategy to go beyond supervised transfer learning and move away from specialized networks for each task

Learn good representations for images and videos and minimize the necessity of large labelled datasets

Language SSL vs. Image SSL: discrete vs. continuous prediction proxy tasks

Two families of SSL methods

1. Contrastive [SimCLR]
2. Non-contrastive [BYOL], [SwAV]

06

Multimodal Learning

Attention Bottlenecks for Multimodal Fusion: state-of-the-art audio-visual classifications

Google Research

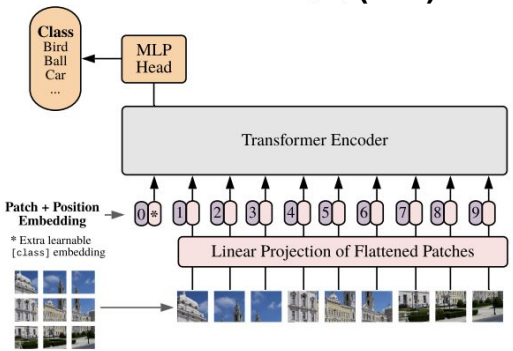
MBT: Attention Bottlenecks for Multimodal Fusion

Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, Chen Sun

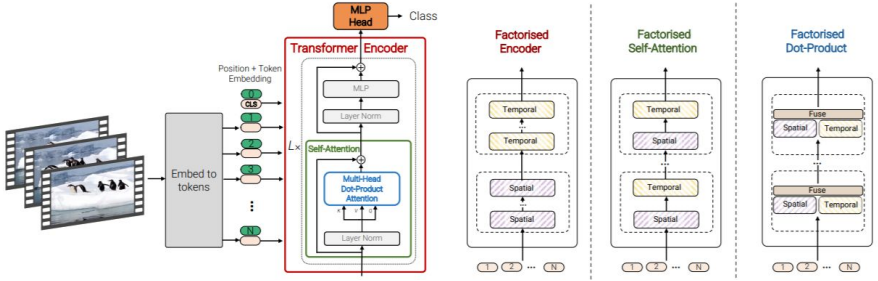
Motivation

- Transformers have had great success on different modalities individually
- Text (BERT), Images (ViT), Videos (ViViT, Timesformer), Audio (AST)

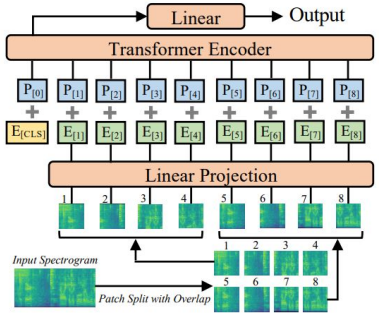
Vision Transformer (ViT)



Video Vision Transformer (ViViT)



Audio Spectrogram Transformer (AST)



Motivation

- Transformers have had great success on different modalities individually
- Text (BERT), Images (ViT), Videos (ViViT, Timesformer), Audio (AST)

However

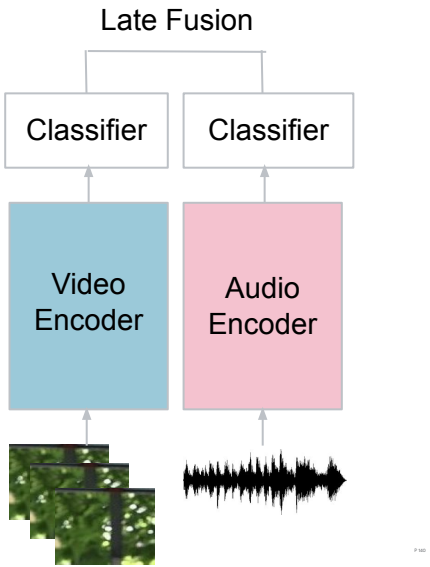
- Video is inherently multimodal - audio, vision, text etc
- Can we create a single multimodal transformer based model that is:
  - Robust
  - Efficient and Scalable
  - Can deal with variable length inputs

Picture source: rawpixel.com - freepik.com

Late Fusion

Multimodal Inputs

- Heterogeneity of inputs (RGB frames, audio spectrograms)
- Specialised architectures
- Different datasets and evaluation benchmarks



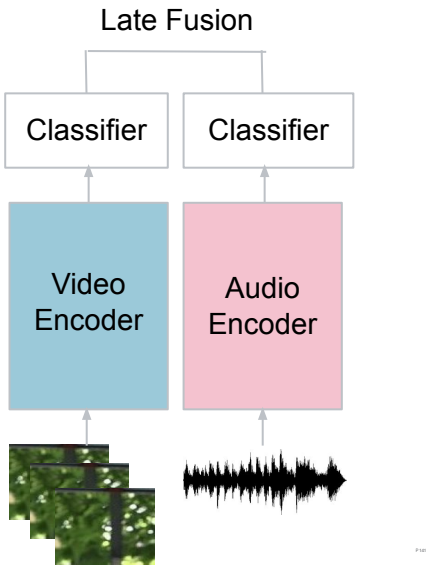
Late Fusion

Multimodal Inputs

- Heterogeneity of inputs (RGB frames, audio spectrograms)
- Specialised architectures
- Different datasets and evaluation benchmarks

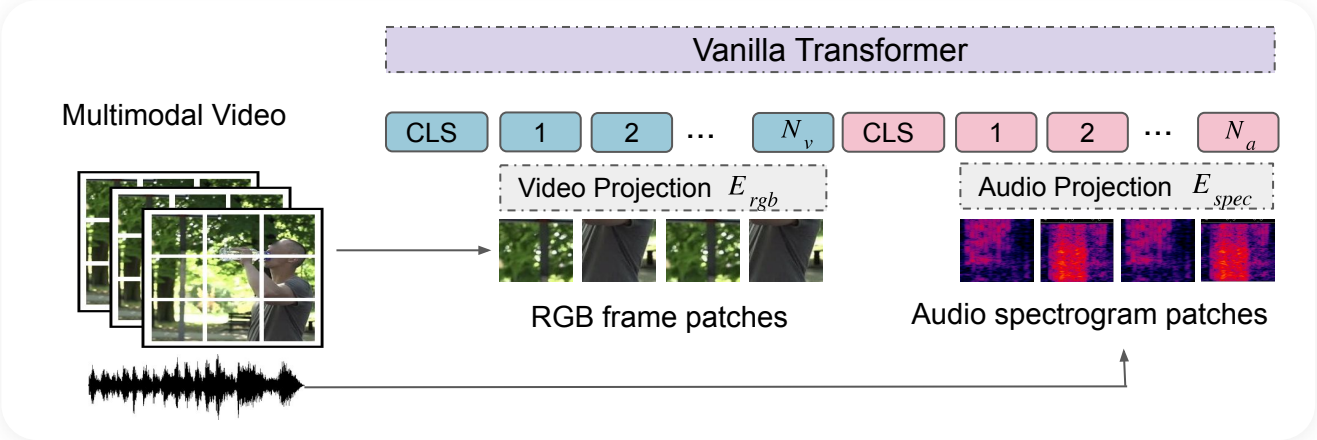
“The Dominant Paradigm”:

- Different encoders
- Output scores or representations are fused right at the end
- This is in contrast to human perception (early or mid fusion)



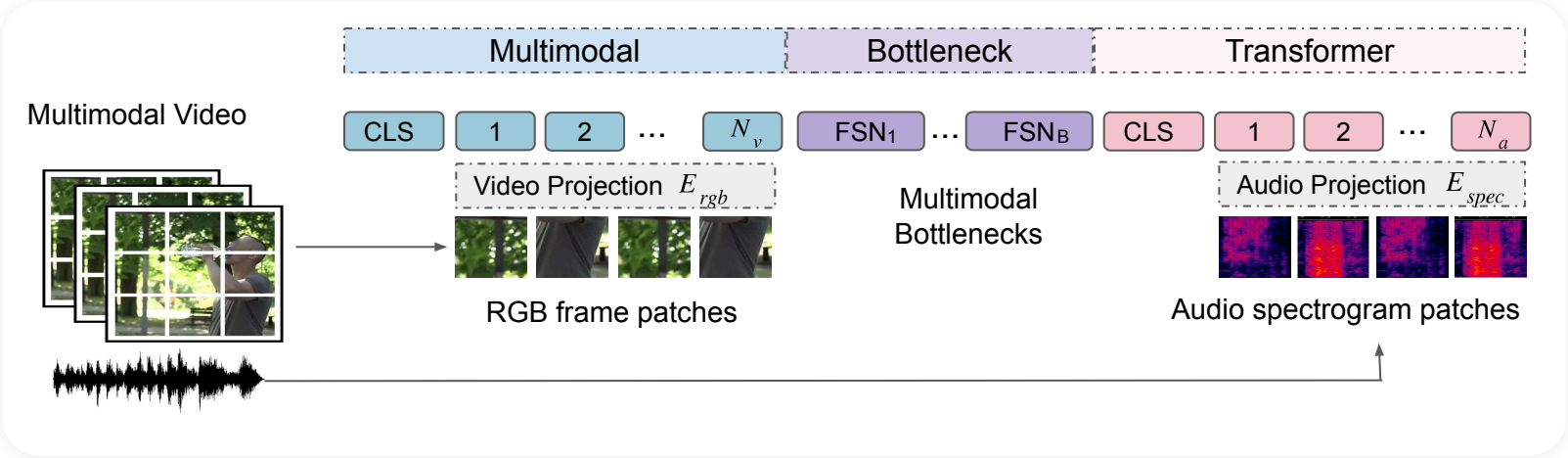
A Vanilla Multimodal Transformer

- Tokenize RGB frame and spectrogram patches
- Feed all tokens to a transformer
- Pairwise self-attention between all tokens, scales quadratically with sequence length



Multimodal Bottleneck Transformers (MBT)

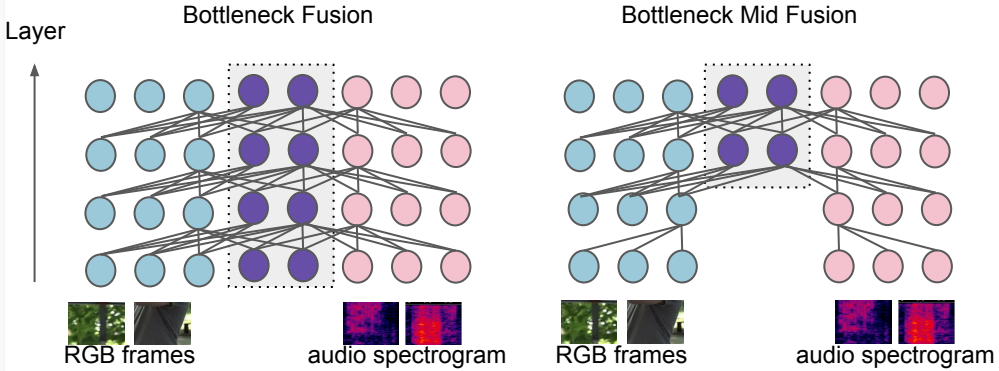
- Full pairwise self attention within a modality
- Introduce a small number of bottleneck tokens (B=4)
- Attention between the visual tokens and the bottleneck tokens
- Attention between the audio tokens and the bottleneck tokens

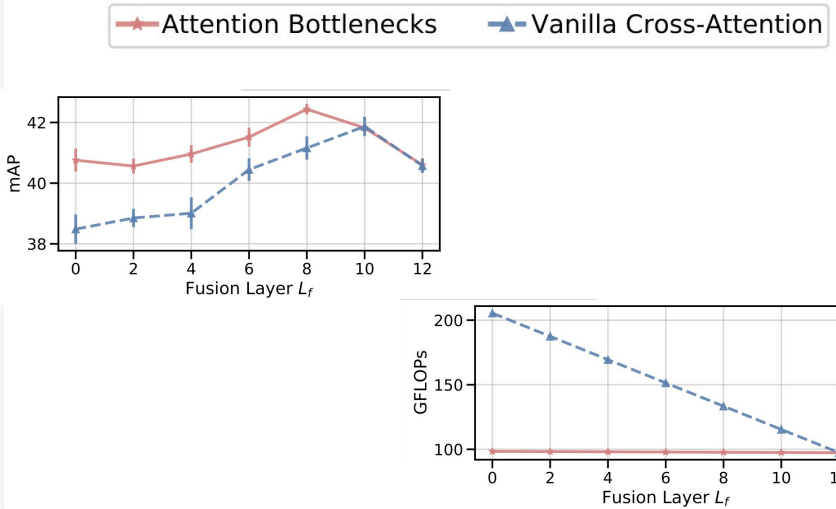


Google Research

Multimodal Bottleneck Transformers (MBT)

- Restrict cross-modal information flow via very few (B=4) bottleneck tokens
- Restrict cross-modal information to later layers (mid-fusion)





Type of token: Audio Video Bottleneck

Google Research

Results on 6 video classification datasets

SOTA across a number of tasks

Action Recognition

Epic-Kitchens

State-of-the-Art*	38.5
Late Fusion	37.9
MBT (ours)	**43.4**

Sound Event Classification

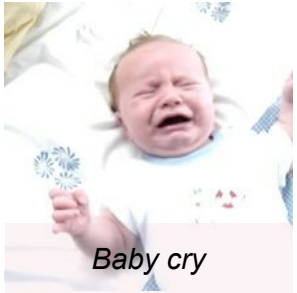
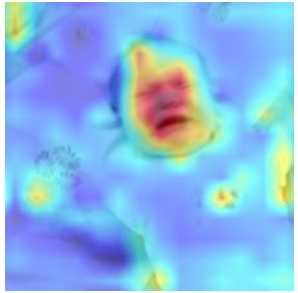
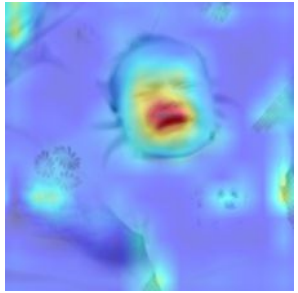
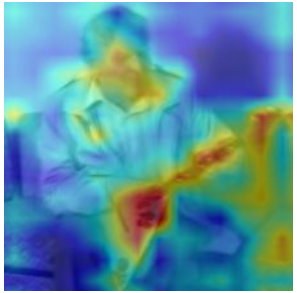
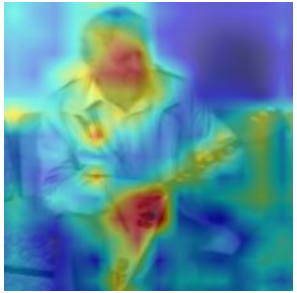
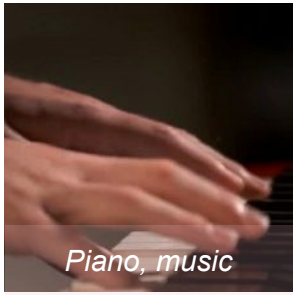
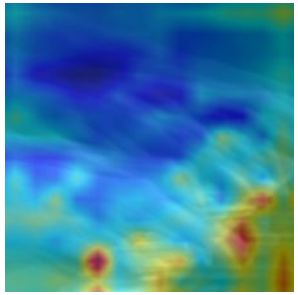
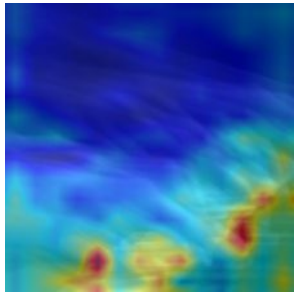
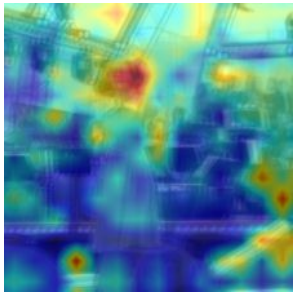
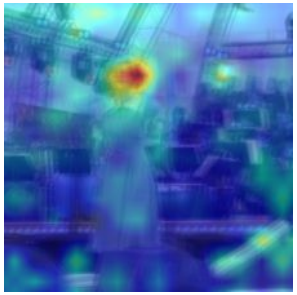
Audioset

State-of-the-Art*	47.3
Late Fusion	49.2
MBT (ours)	**52.1**

Perform ablations on number of bottleneck tokens, sequence length and synchronisation between modalities

Google Research

Attention Heatmaps

Mid Frame	Vanilla Fusion	MBT	Mid Frame	Vanilla Fusion	MBT
					
					

Resources

Code and
Models will
be released

Webpage

https://a-nagrani.github.io/mbt.html

Self-Supervised Multimodal Versatile Networks: visual, audio and language streams

Motivation

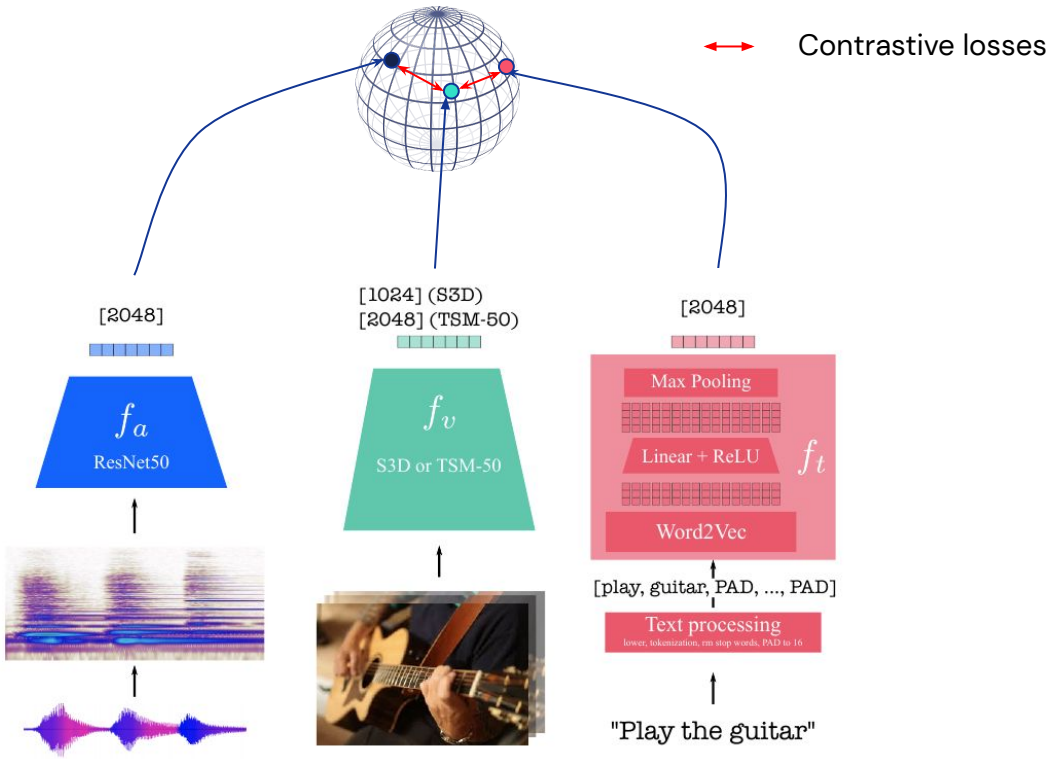
Self-supervised learning on modalities naturally present in videos:

Vision, Audio and Language

Research questions:

- Are three modalities better than two for downstream tasks?
- What architectural designs are going to give benefits beyond performance?

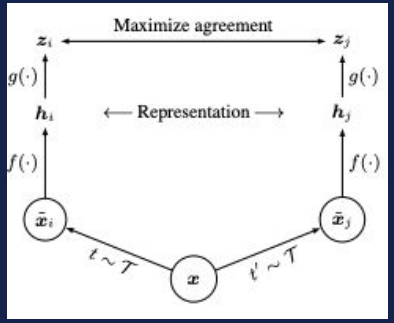
Main idea



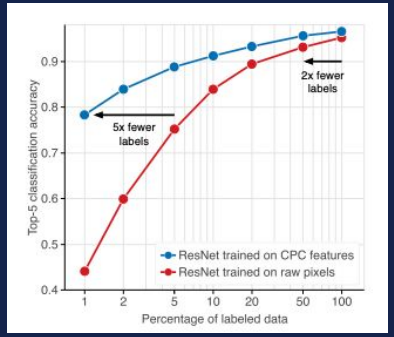
This is an “old” idea: *DeVise*, Frome et al. NeurIPS13 and *WSABIE, Weston et al.* IJCAI 2011.

Related Work

Vision

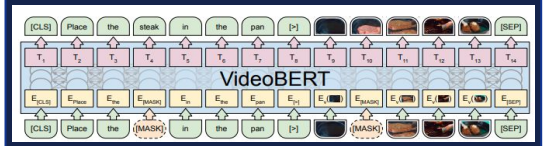


SimCLR: Chen et al, 2020

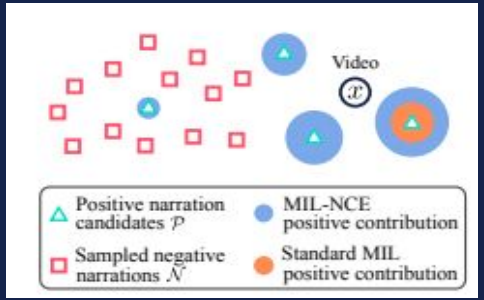


CPC: Oord et al 2018 / Henaf et al, 2019

Vision+Text

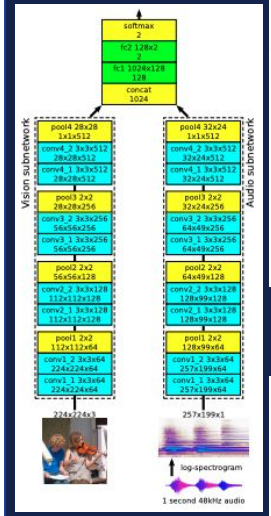


VideoBERT: Sun et al, 2019

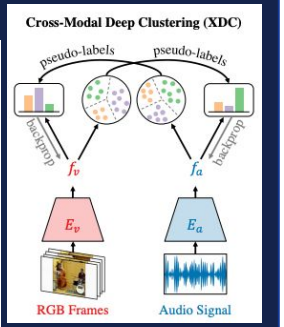


MIL-NCE: Miech, Alayrac et al, 2020

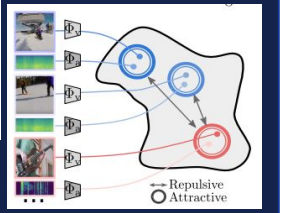
Vision+Audio



L3: Arandjelovic and Zisserman, 2017



XDC: Alwassel at al, 2020



GDT: Patrick at al, 2020

Architecture versatility checklist

(i) takes as input any of the three modalities

(ii) respects the specificity of modalities

(iii) enables the different modalities to be easily compared

(iv) efficiently applicable to visual data in the form of videos or images

How to obtain the embeddings for each modalities?

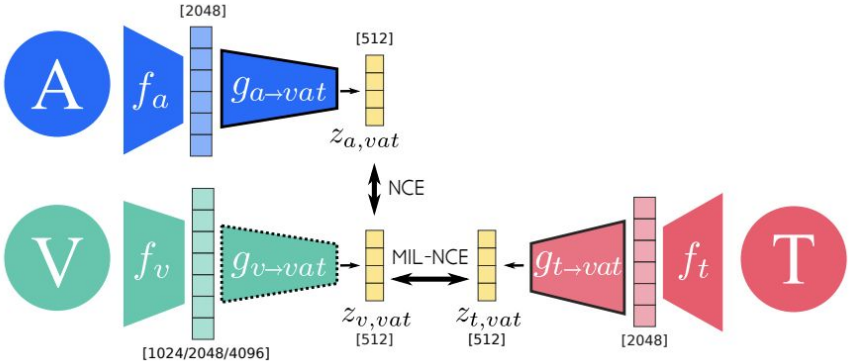
- Should all modalities be embedded in a same space?
- Should we instead have specific embedding spaces for different pairs of modalities?

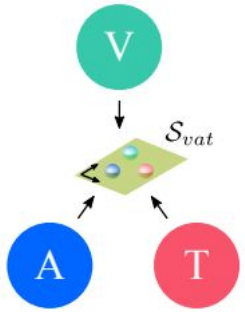
We explore these questions via network architecture options: **embedding graph design.**

Embedding graph design
Option I: Shared spaces

Embedding all the modalities in the same space.

- ✓ Enables the different modalities to be compared
- ✗ Treat all modalities as if they were equal



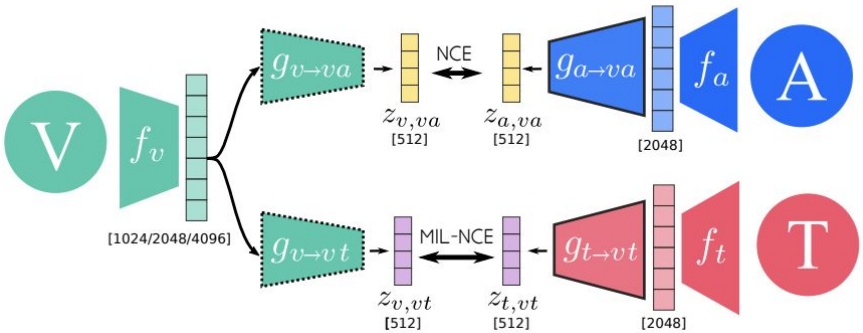


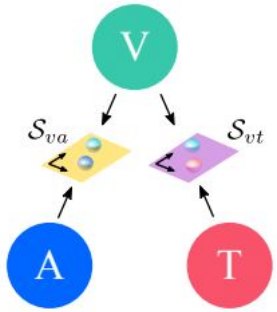
Shared spaces

Embedding graph design
Option II: Disjoint spaces

Separate spaces for visual-audio and visual-text

- ✓ Allows a specific treatment of modalities
- ✗ Not easy to compare modalities/hard to transfer from one modality to the other



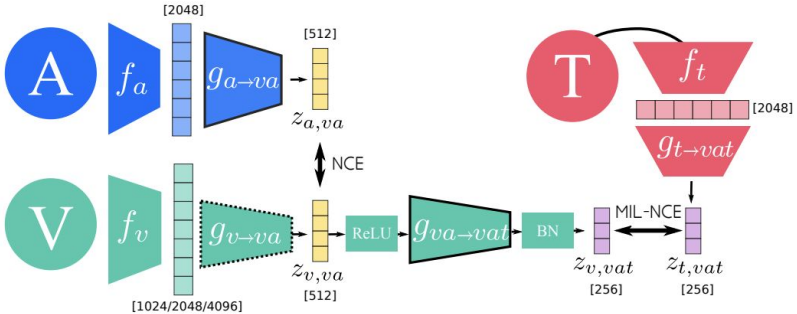


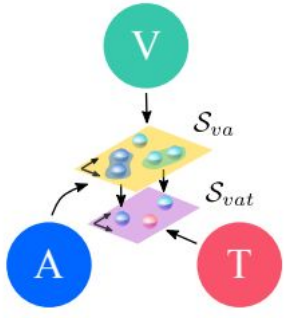
Disjoint spaces

Embedding graph design
Option III: Fine and Coarse

Intuition: audio is more **fine** grained (e.g., multiple sounds of guitar) whereas text is more **coarse** (a single word for guitar) ⇒ The Fine and Coarse design:

- ✓ respects the specificity of modalities
- ✓ enables the different modalities to be easily compared
- ✓ has the best results in several downstream tasks





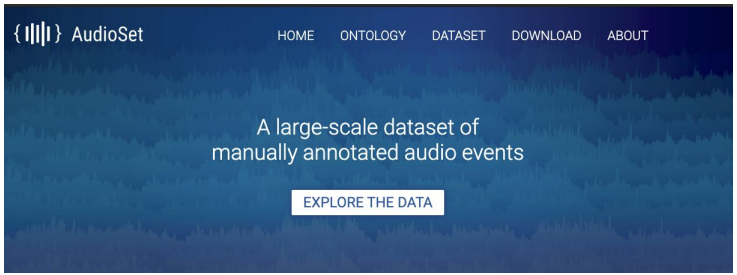
Fine and Coarse (FAC)

Which pretraining datasets?

HowTo100M: 1M videos, 100M clips, 20K tasks, *text obtained from ASR*.

HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips, Miech, Zhukov, Alayrac et al., ICCV19

AudioSet: 2M videos (with audio tracks), we do not extract text for this dataset.



Audio Set: An ontology and human-labeled dataset for audio events, Gemmeke et al. ICASSP 2017

Training loss

Contrastive loss for video – audio

$$\mathbf{NCE}(x_v, x_a) = -\log\left(\frac{\exp(z_{v,vat}^{\top} z_{a,vat}/\tau)}{\exp(z_{v,vat}^{\top} z_{a,vat}/\tau) + \sum_{z' \sim \mathcal{N}(x)} \exp(z'^{\top}_{v,vat} z'_{a,vat}/\tau)}\right)$$

Adaptation for video – text

$$\mathbf{MIL\text{-}NCE}(x_v, x_t) = -\log\left(\frac{\sum_{z \in \mathcal{P}(x)} \exp(z_{v,vat}^{\top} z_{t,vat}/\tau)}{\sum_{z \in \mathcal{P}(x)} \exp(z_{v,vat}^{\top} z_{t,vat}/\tau) + \sum_{z' \sim \mathcal{N}(x)} \exp(z'^{\top}_{v,vat} z'_{t,vat}/\tau)}\right)$$

Joint loss

$$\mathcal{L}(x) = \lambda_{va}\mathbf{NCE}(x_v, x_a) + \lambda_{vt}\mathbf{MIL\text{-}NCE}(x_v, x_t)$$

Audio to video

Rank 1

Rank 2

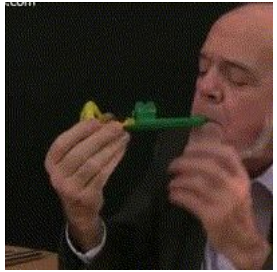
Rank 3

Audio to video

Rank 1

Rank 2

Rank 3



Text to video

Input text	Rank 1	Rank 2	Rank 3
“add fresh chopped tomatoes and stir”			

Text to video

Input text

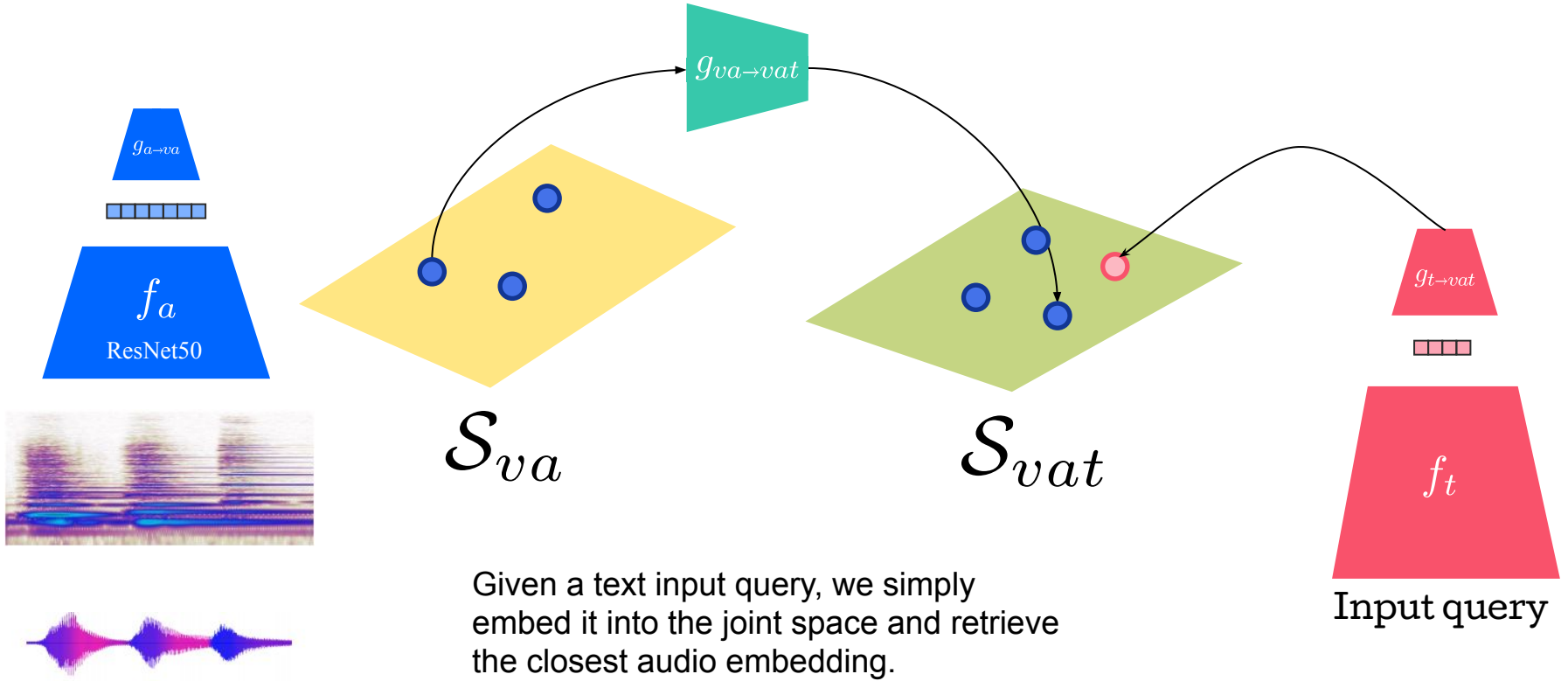
“pour some oil into a hot pan”

Rank 1

Rank 2

Rank 3

Text to audio retrieval in the coarse space



Given a text input query, we simply embed it into the joint space and retrieve the closest audio embedding.

Text to audio retrieval

Input text

“airplane”

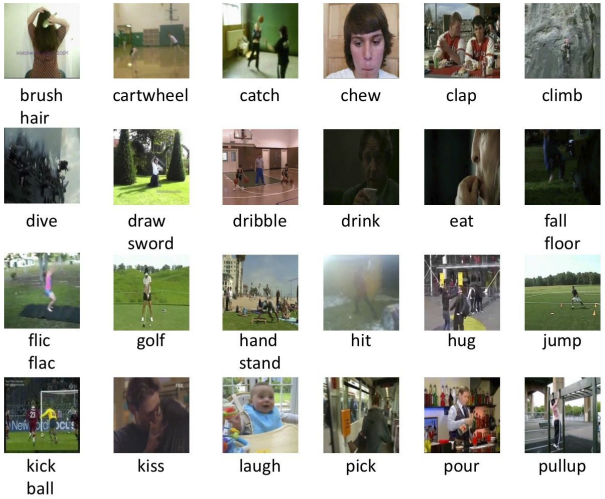
Rank 1

Text to audio retrieval

Input text	Rank 1
“chirping bird”	

Evaluating the representation

Video representation: Transfer learning to UCF101 and HMDB51 datasets (the two current standard downstream tasks for video action classification).

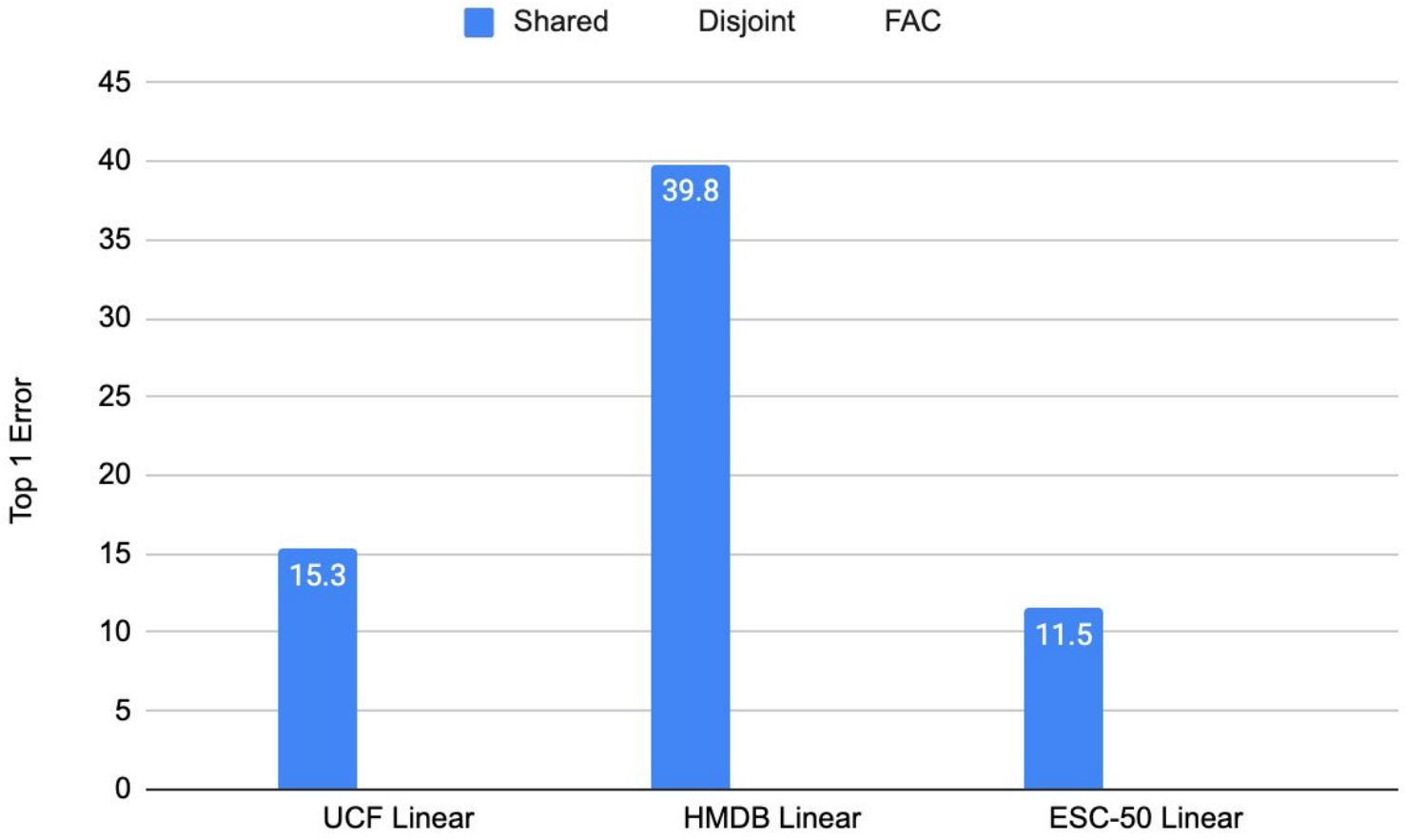


HMDB51: 51 classes, 6.7K clips total

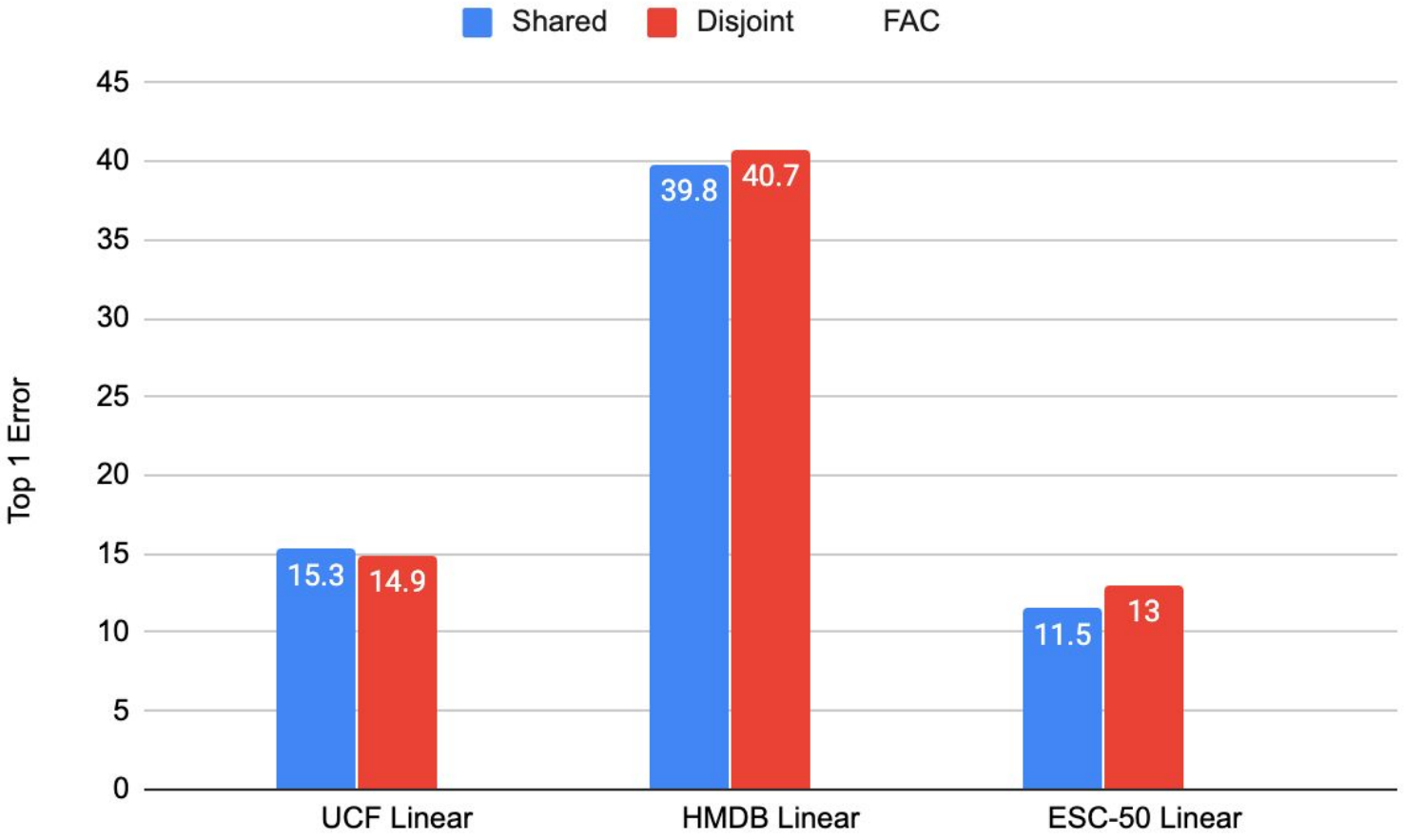
UCF101: 101 classes, 13K clips total

Audio representation: Transfer learning to ESC-50 (audio classification)

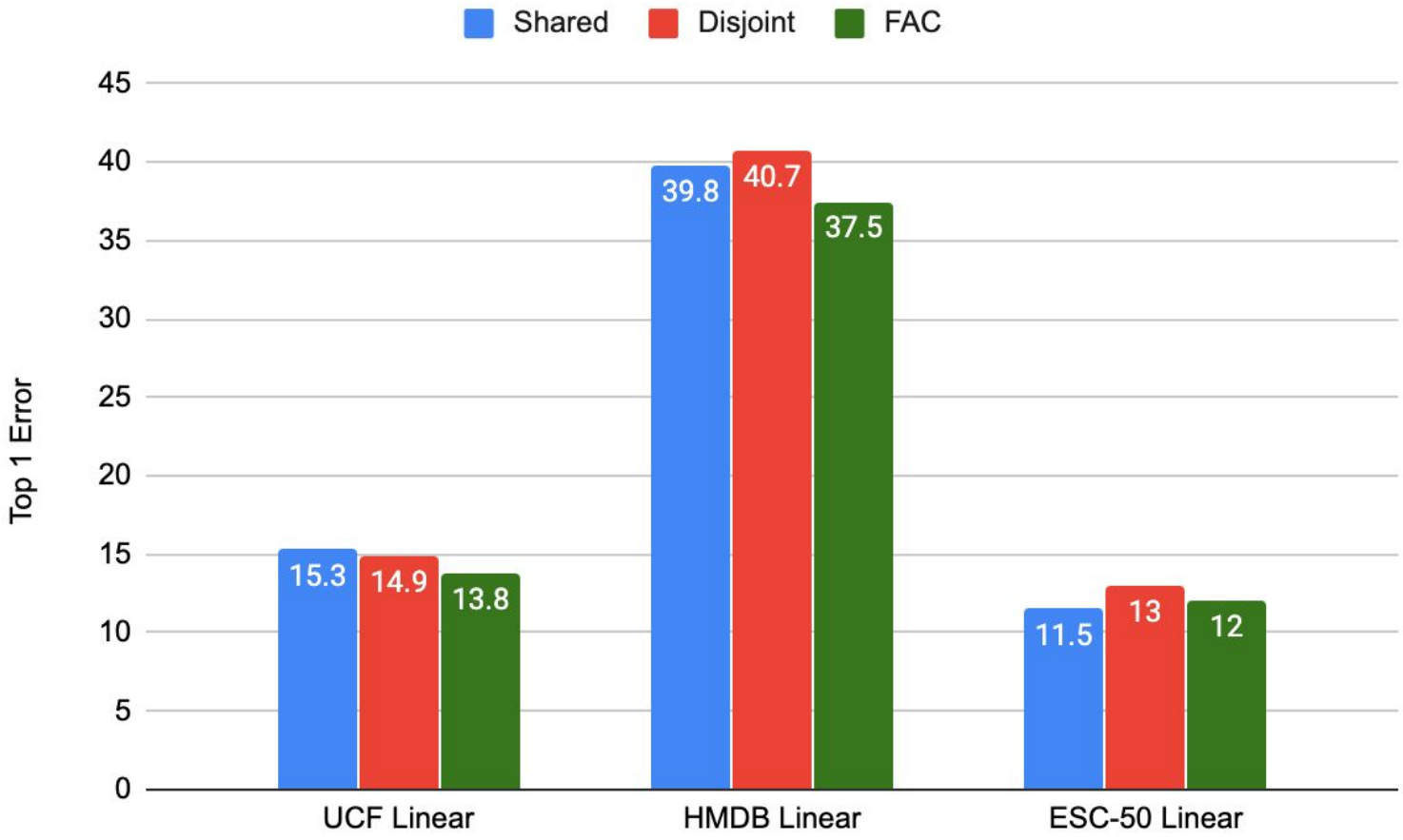
Embedding graph design



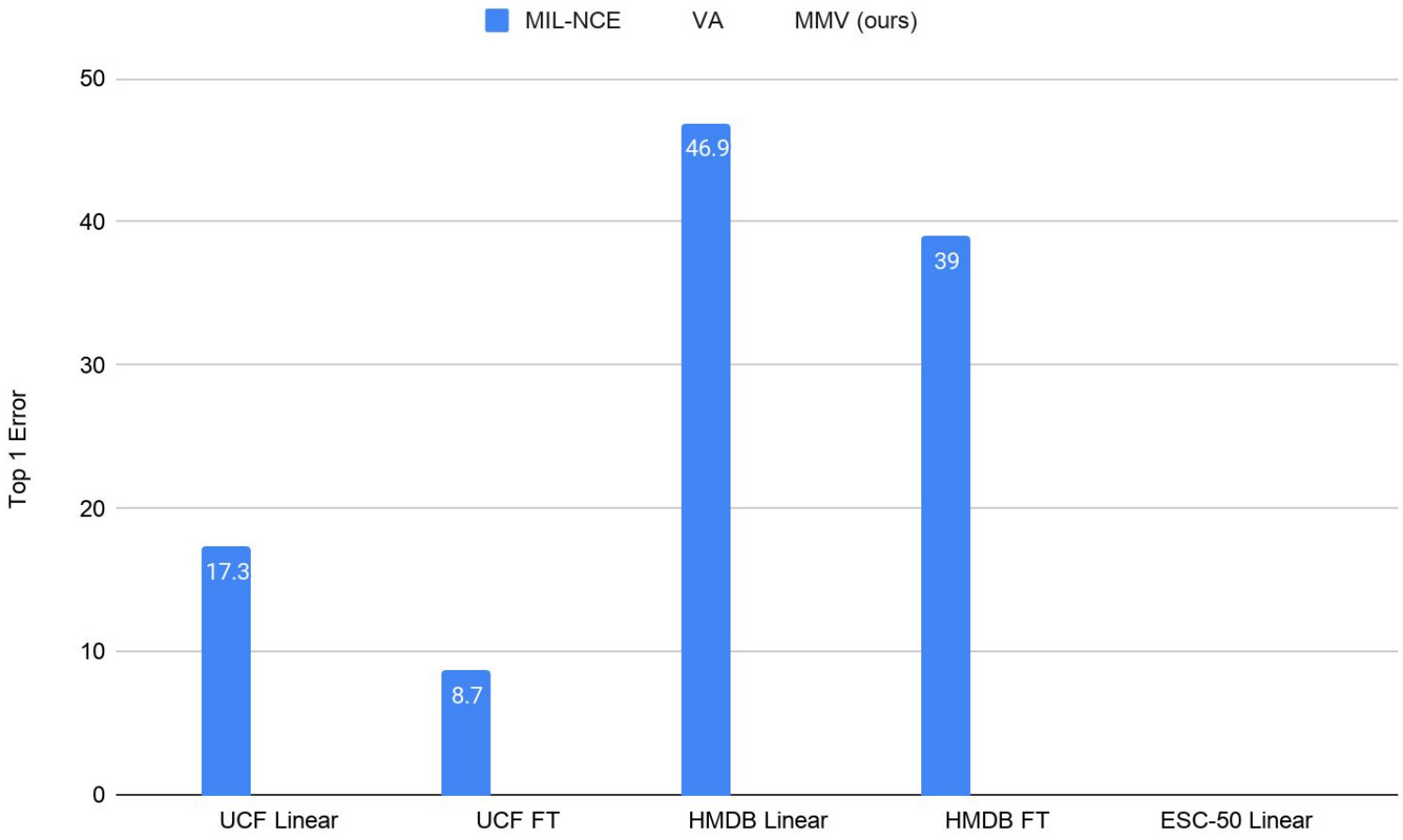
Embedding graph design



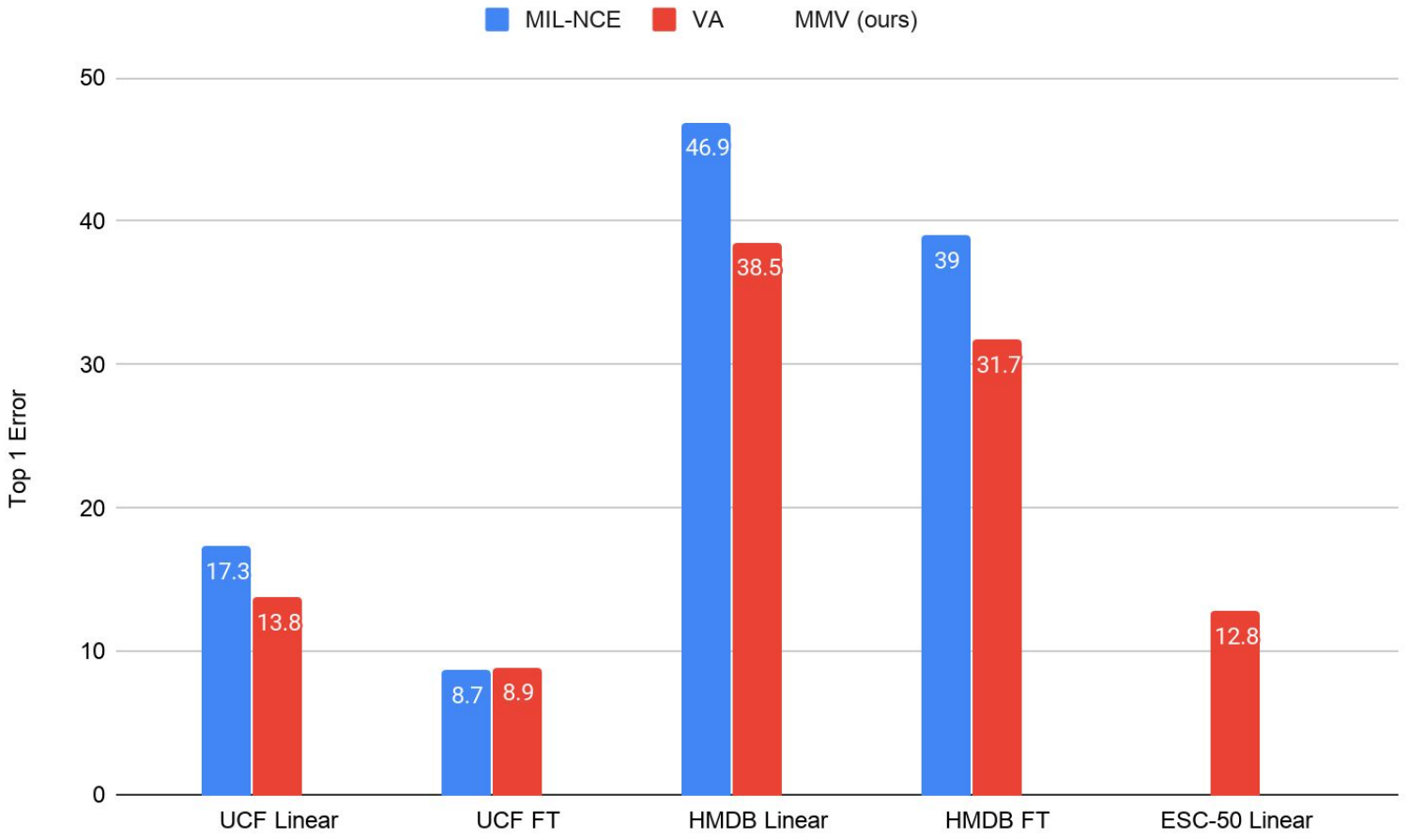
Embedding graph design



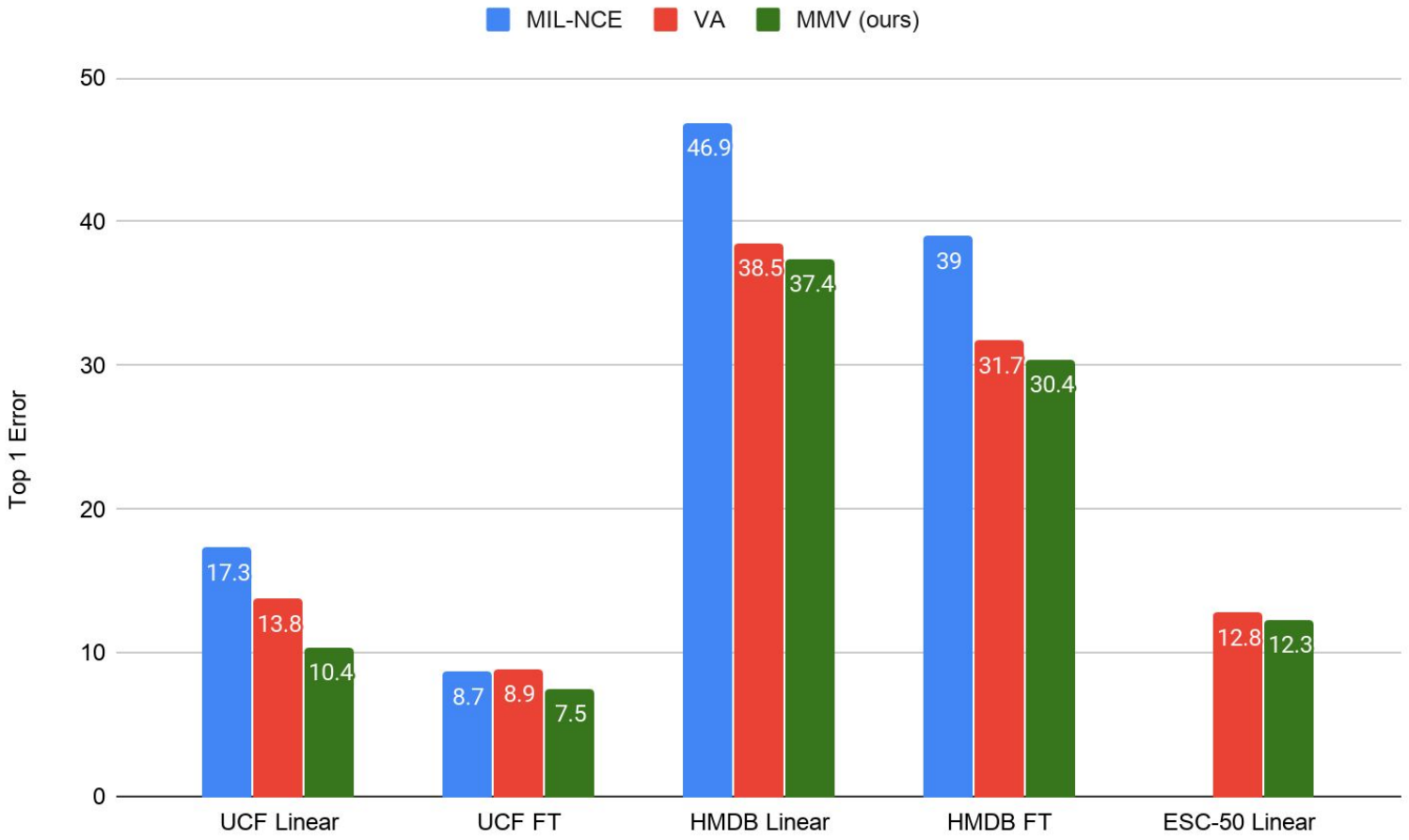
Do more modalities help?



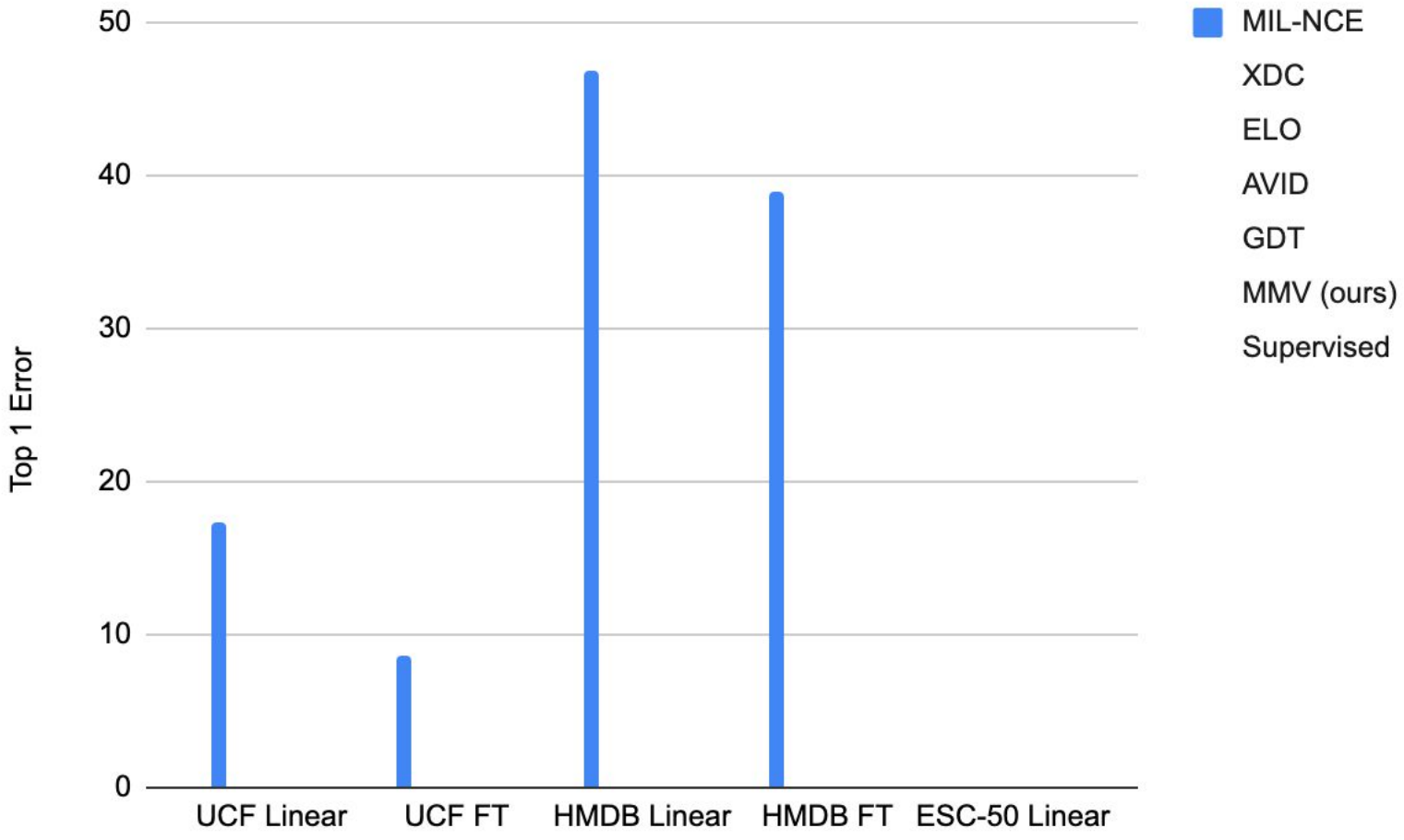
Do more modalities help?



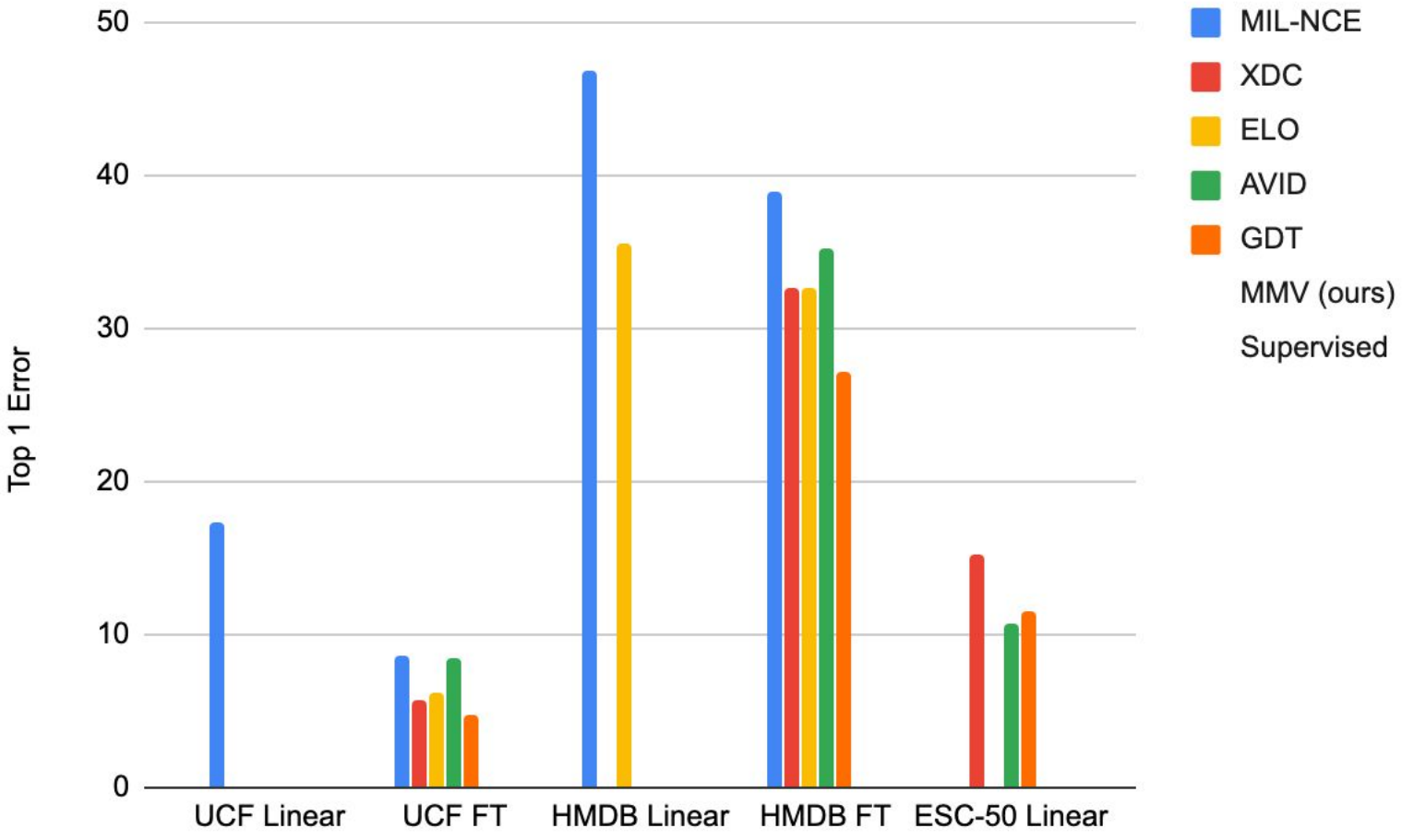
Do more modalities help?



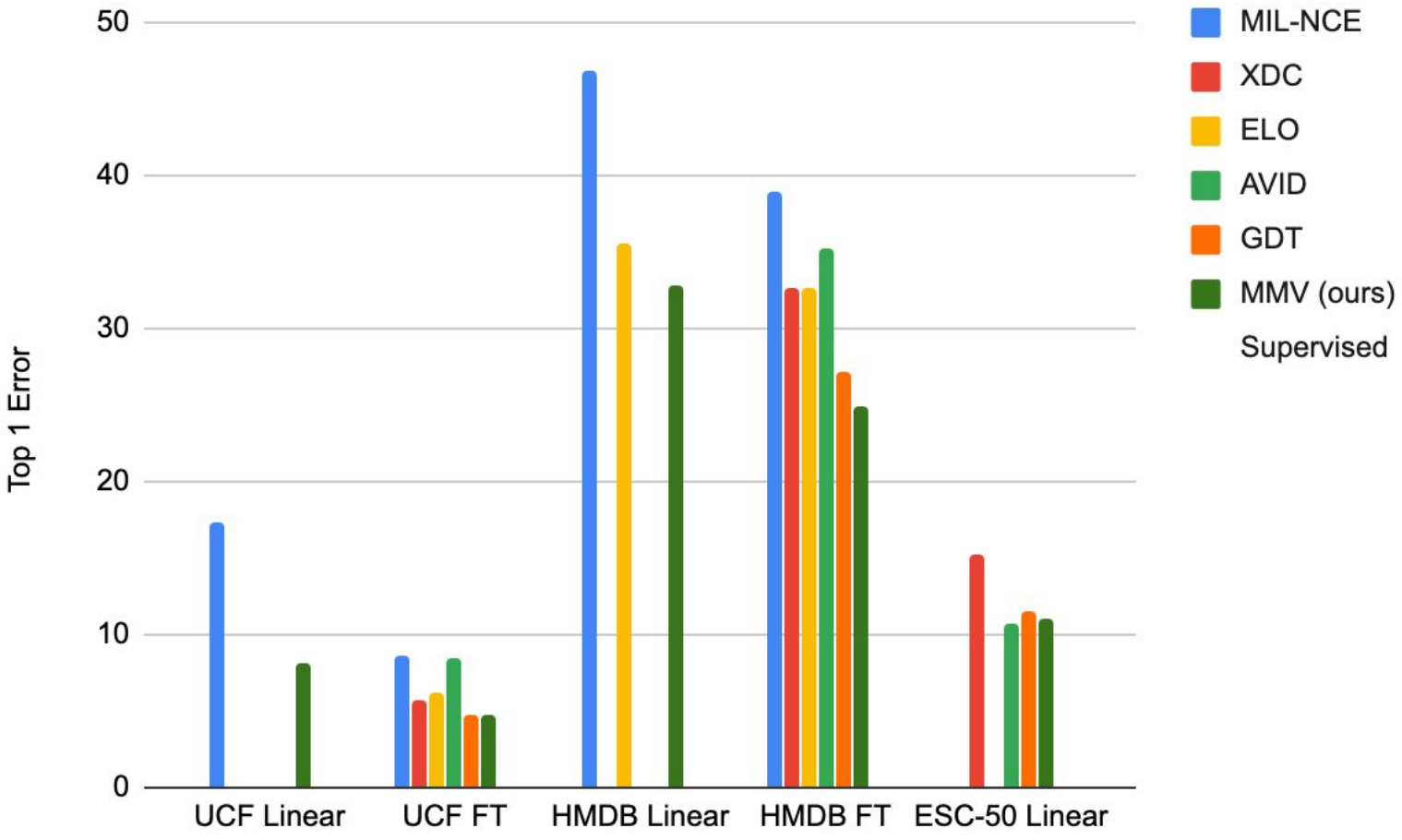
State-of-the-art comparison



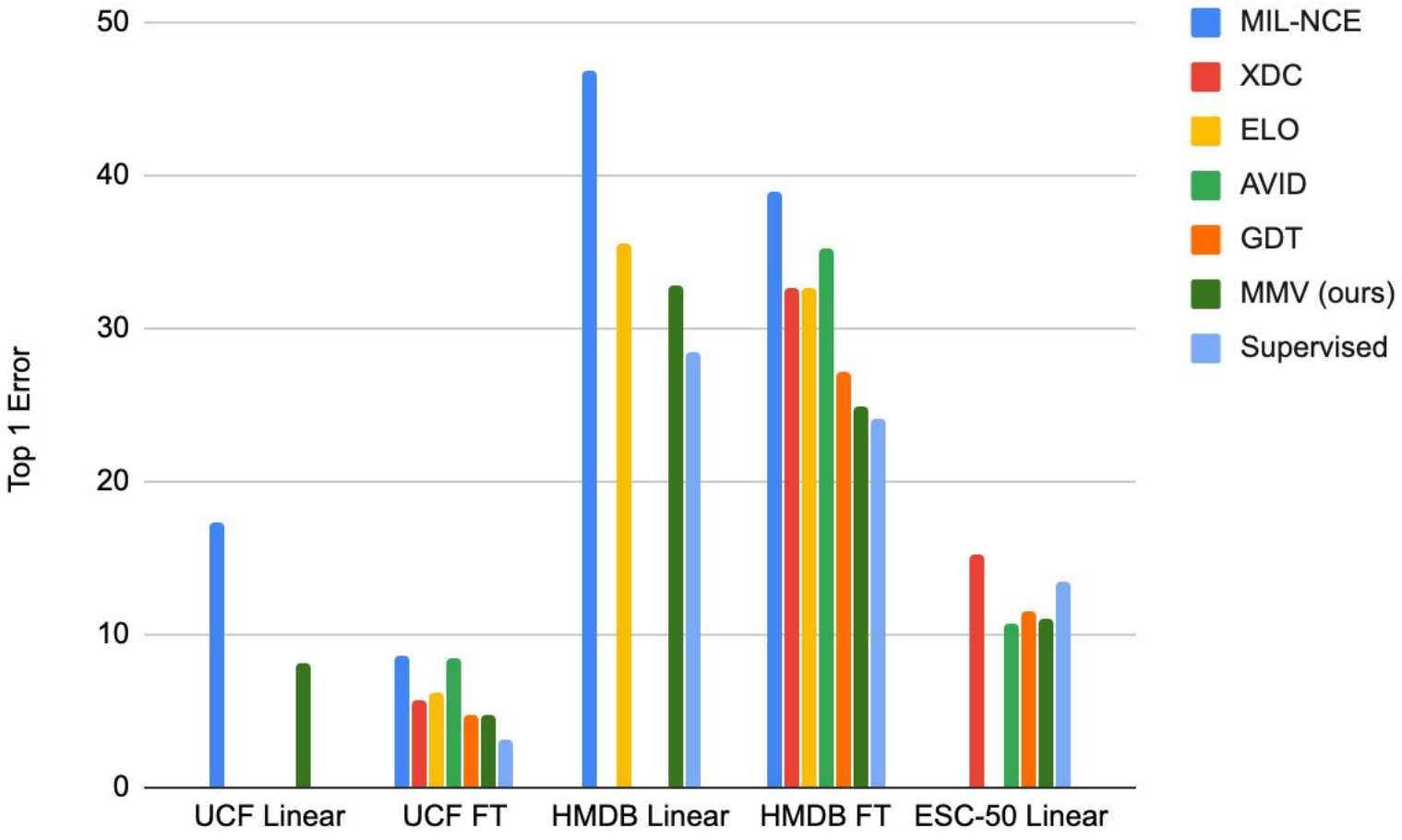
State-of-the-art comparison



State-of-the-art comparison



State-of-the-art comparison



Architecture versatility checklist

(i) takes as input any of the three modalities

(ii) respects the specificity of modalities

(iii) enables the different modalities to be easily compared

(iv) efficiently applicable to visual data in the form of videos or images

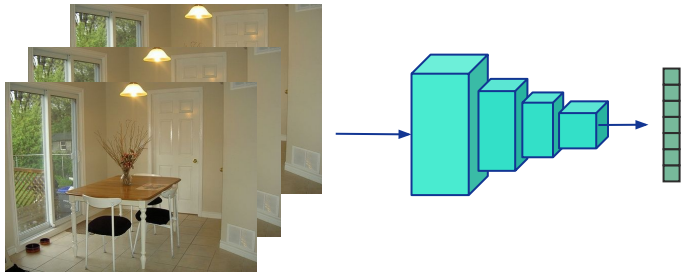
Network deflation

Motivation:

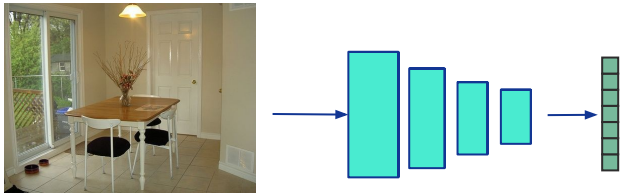
- Humans learn from continuous stream of data (~video).
- Most works consider learning first from images to apply models to video.

Desiderata:

- Our models to be applicable off-the-shelf on images
- Our models to be efficient when applied to images

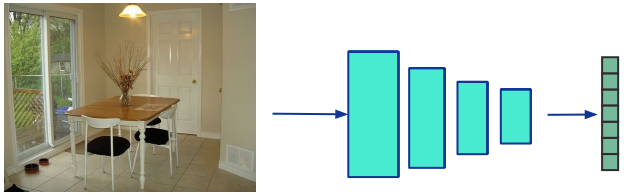


A standard solution: Inflated input



Proposed solution: Deflated network

Network deflation



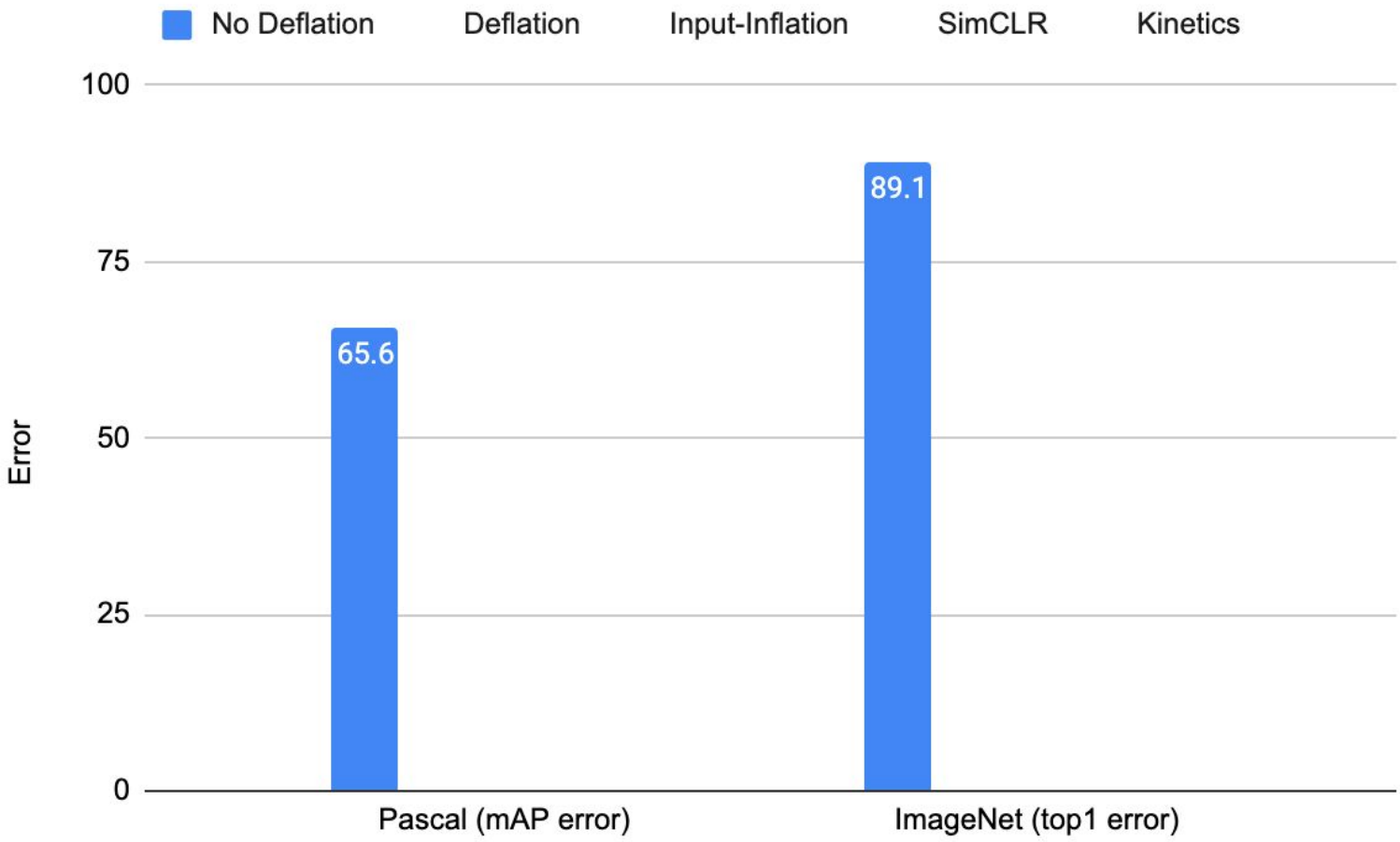
Network deflation: we transform 3D conv kernels into 2D by summing along temporal dimensions.

Challenge: this does not address the temporal padding effect.

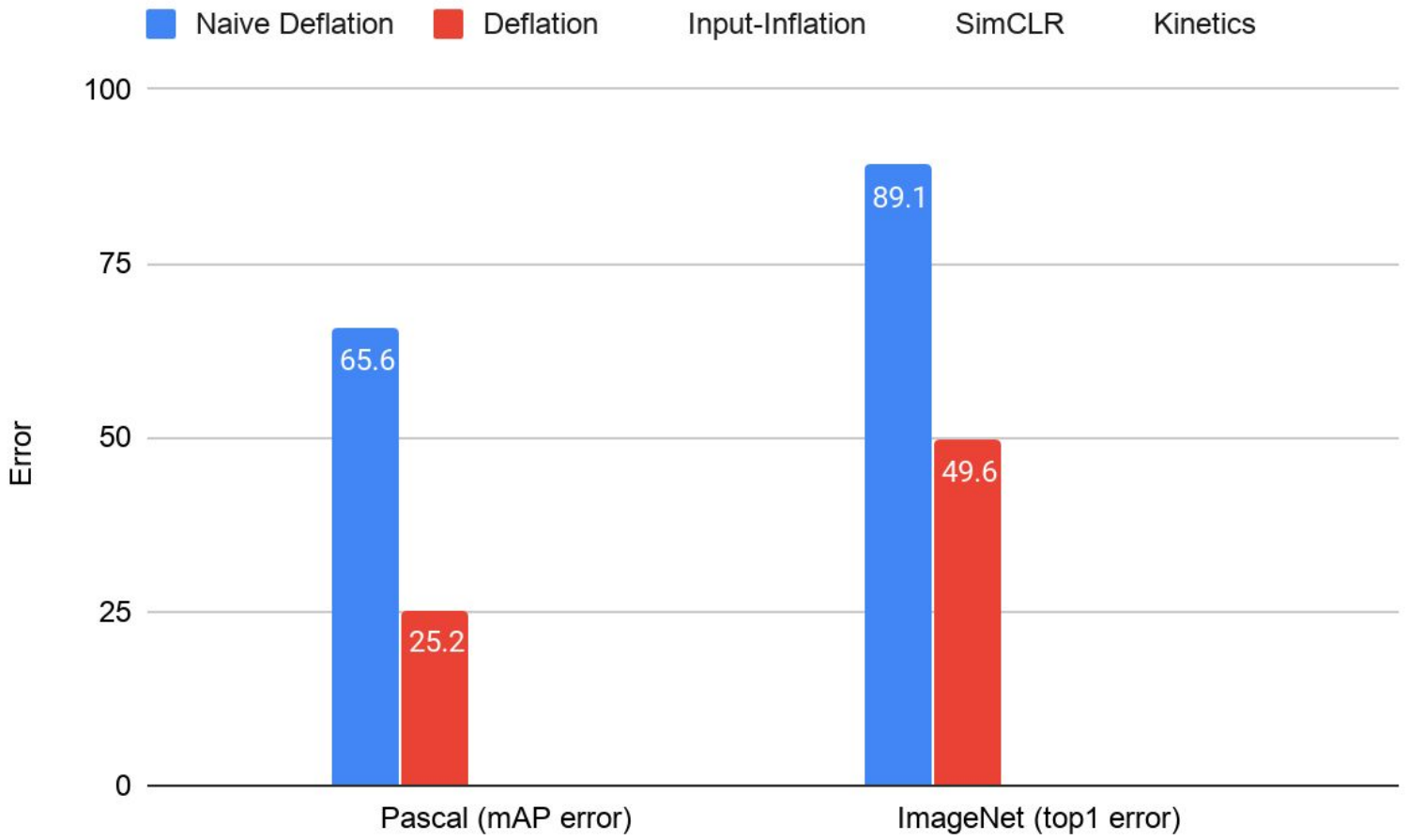
Solution: we retrain the batch normalization parameters (gamma and beta) to match the output of the input-inflated network from HowTo100M with an unsupervised reconstruction loss.

Train only the BN gamma and beta params

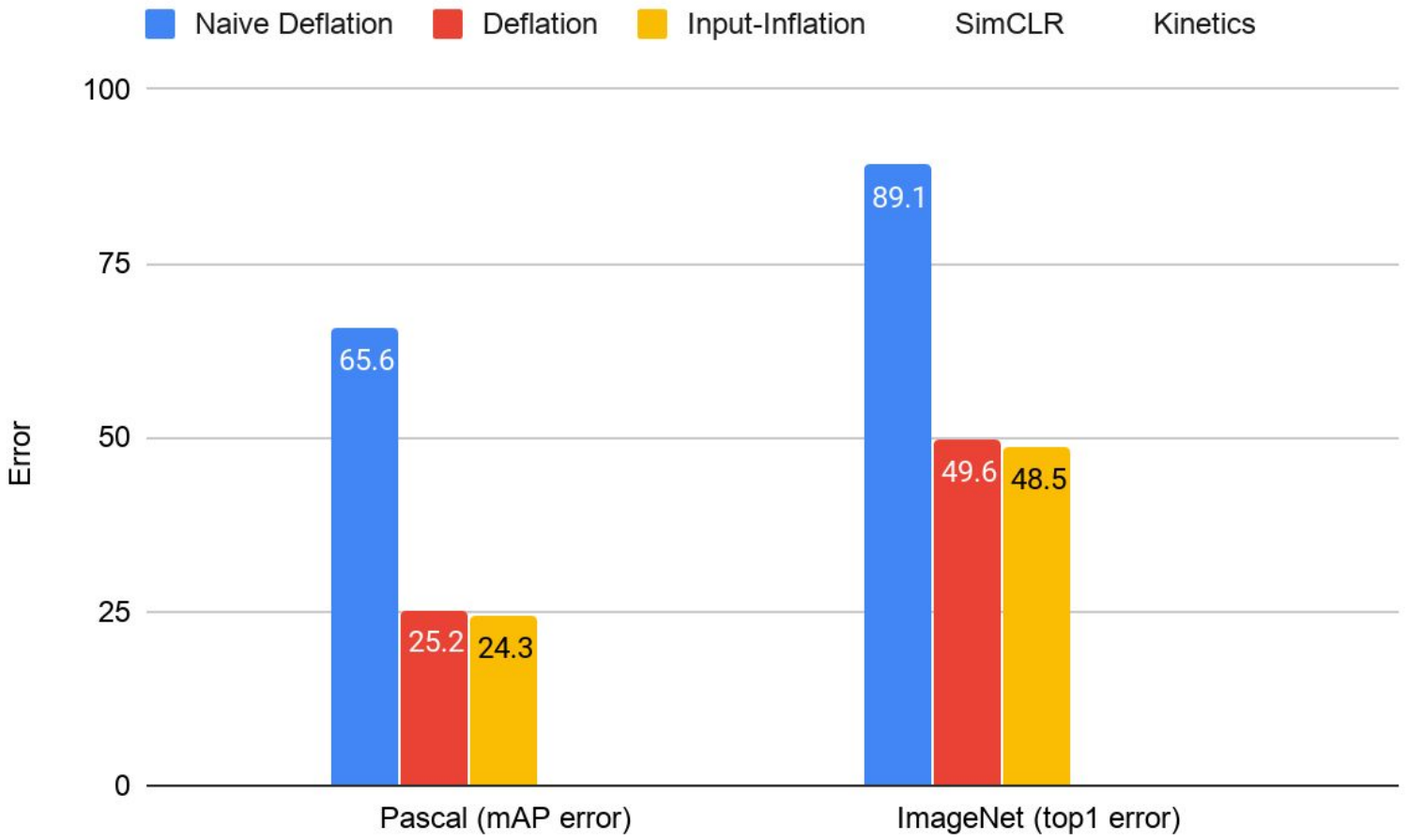
Transfer to image results



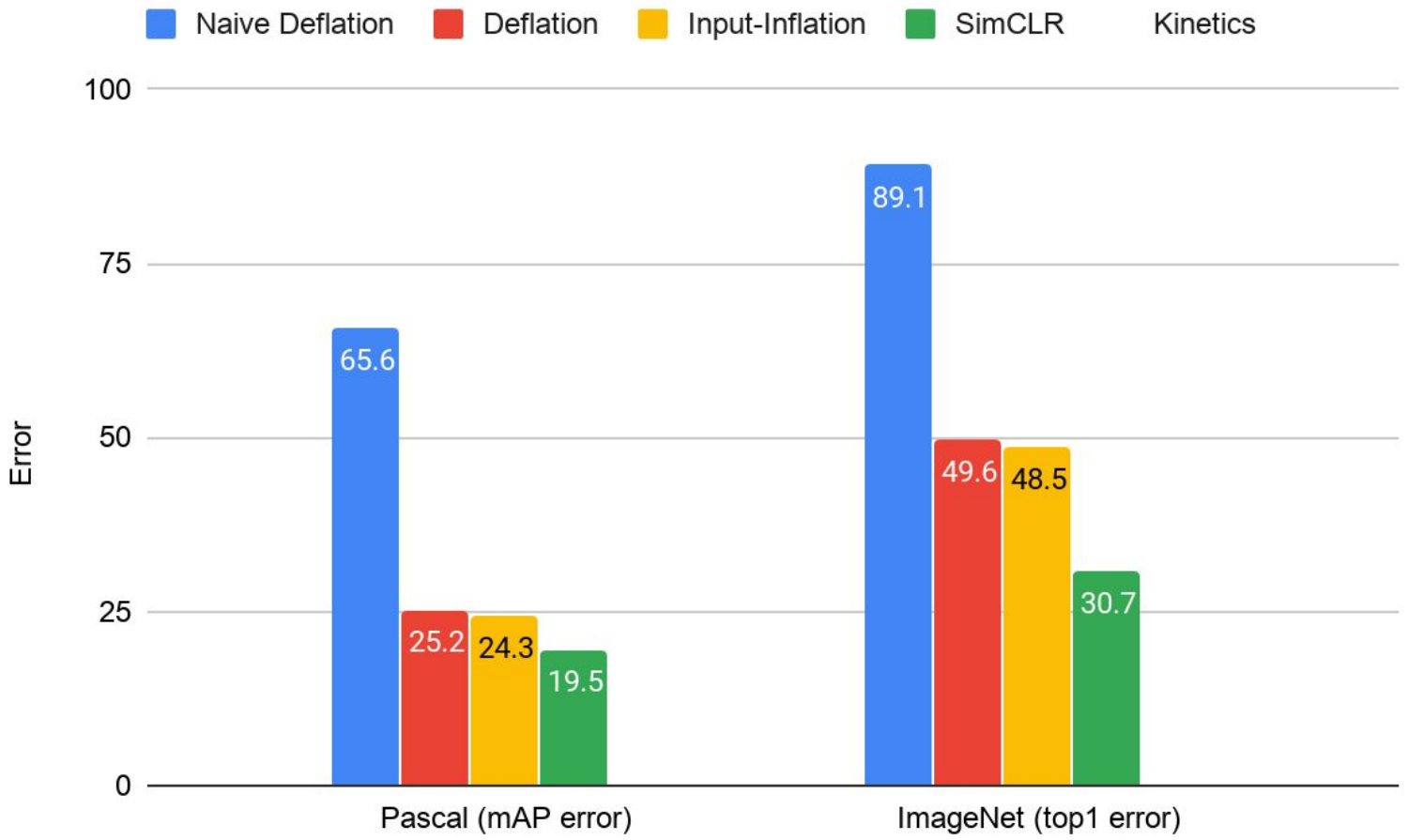
Transfer to image results



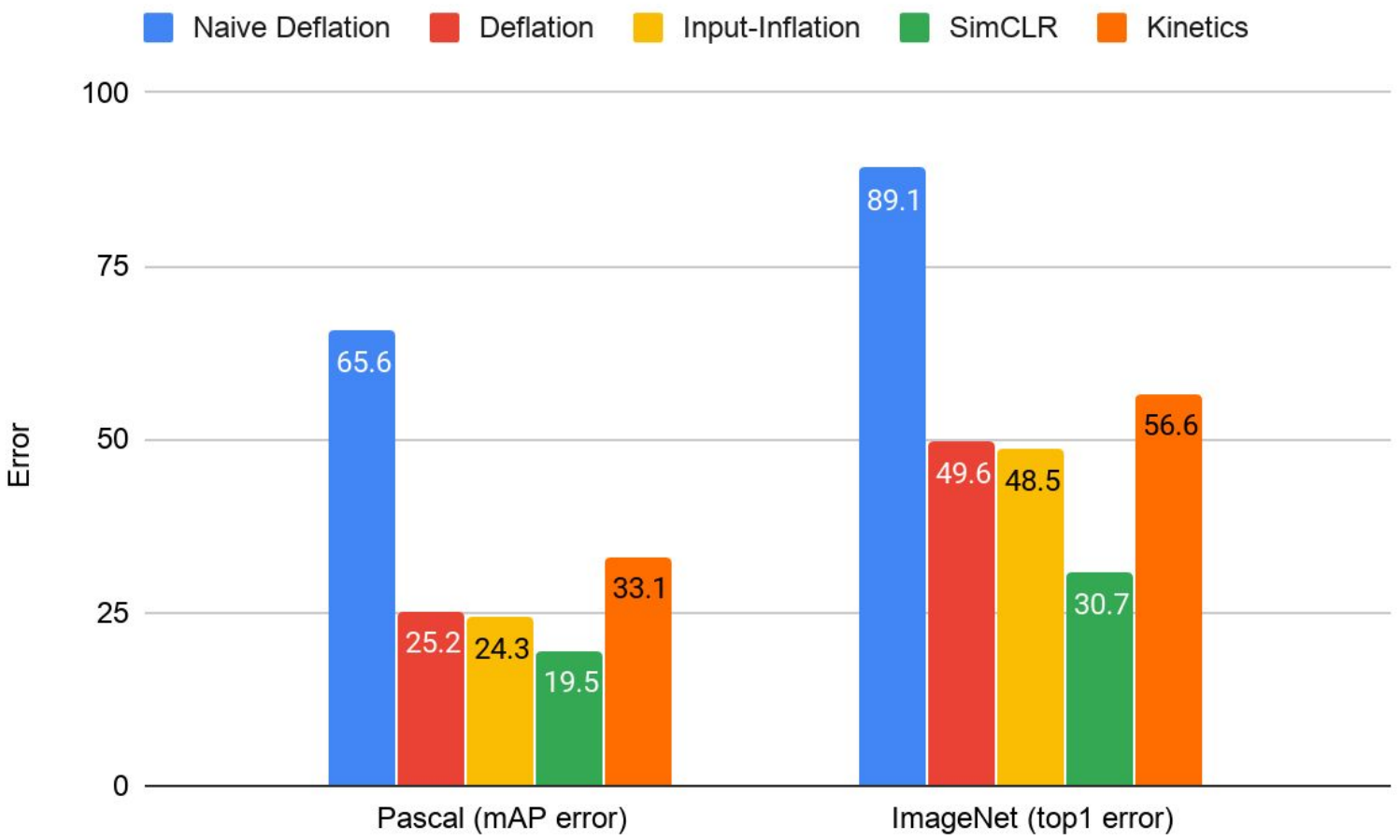
Transfer to image results



Transfer to image results



Transfer to image results



Summary and conclusion

- We proposed a general framework to learn representation by aligning modalities
- We achieve SoTA results for self-supervised learning in several downstream tasks
- One step towards transferring from video to images
- Potential new applications such as zero-shot Text-to-Audio retrieval

Limitations

- Better leverage of the modalities embedding spaces (zero shot transfer from modalities...).
- Very simple fusion strategy for now. We can use more complex fusion strategies (spatio temporal instead of global…)
- Make explicit differentiation between speech and sound:
  - Explore source separation techniques between speech and sound (are both signal necessary to learn?)
  - What kind of additional signal is contained in the speech vs ASR? How is that useful?

Case studies: cross-modal fine-grained reasoning

The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes

[Kiela et al., 2021]

Question: *How do we detect which images in the context of text could be harmful to users?*

Exploring Hate Speech Detection in Multimodal Publications

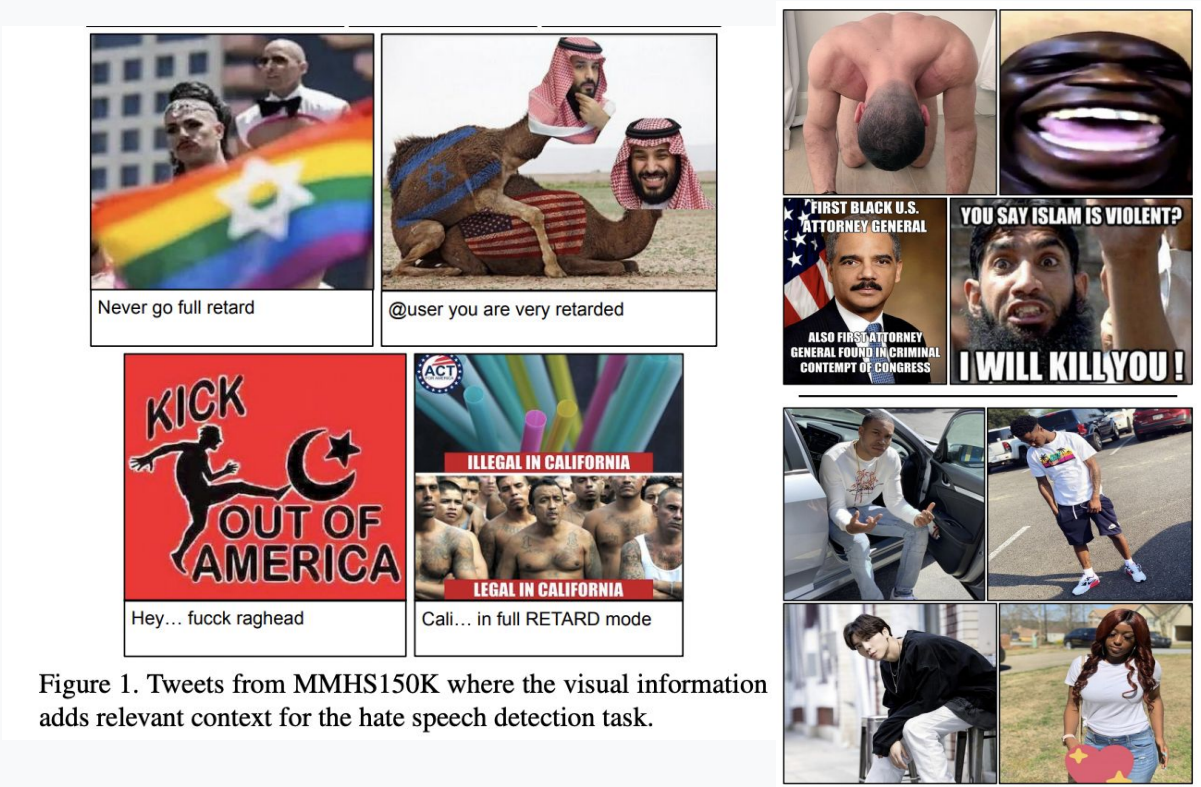


Figure 1. Tweets from MMHS150K where the visual information adds relevant context for the hate speech detection task.

[Gomez et al., 2019]

Question: *How do we detect which images in the context of text could be harmful to users?*

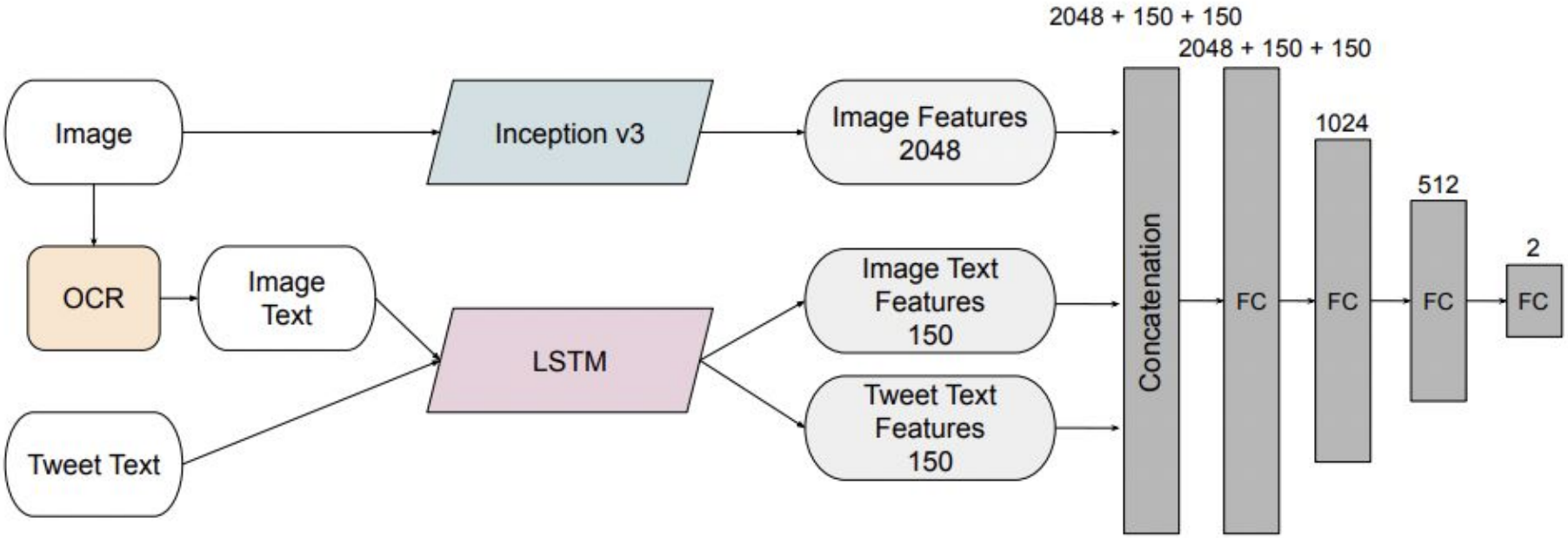
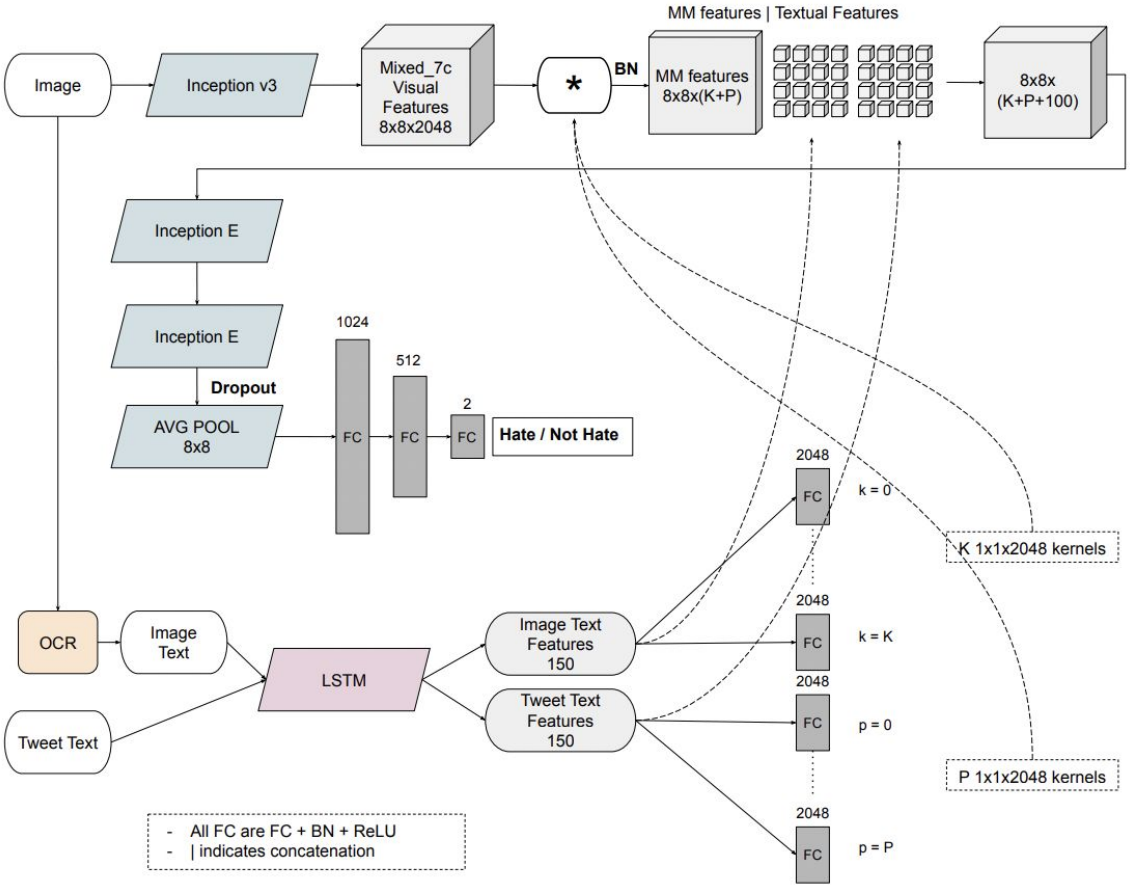
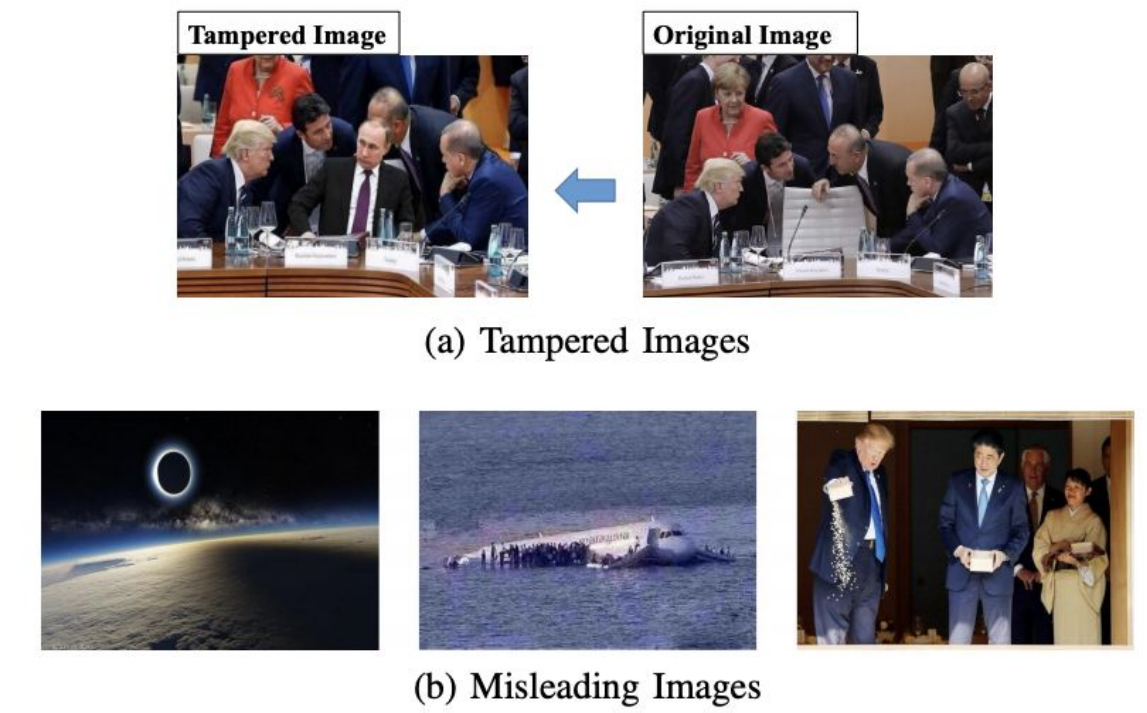


Image
Inception v3
Image Features
2048
OCR
Image
Text
LSTM
Image Text
Features
150
Tweet Text
Tweet Text
Features
150
Tweet Text
2048 + 150 + 150
2048 + 150 + 150
Concatenation
FC
1024
FC
512
FC
2
FC


MM features | Textual Features
Image
Inception v3
Mixed_7c
Visual
Features
8x8x2048
*
BN
MM features
8x8x(K+P)
8x8x
(K+P+100)
Inception E
Inception E
Dropout
AVG POOL
8x8
1024
512
2
FC
FC
FC
Hate / Not Hate
2048
FC
k = 0
K 1x1x2048 kernels
OCR
Image
Text
Tweet Text
LSTM
Image Text
Features
150
Tweet Text
Features
150
2048
FC
k = K
2048
FC
p = 0
P 1x1x2048 kernels
2048
FC
p = P
- All FC are FC + BN + ReLU
- | indicates concatenation

Exploiting Multi-domain Visual Information for Fake News Detection

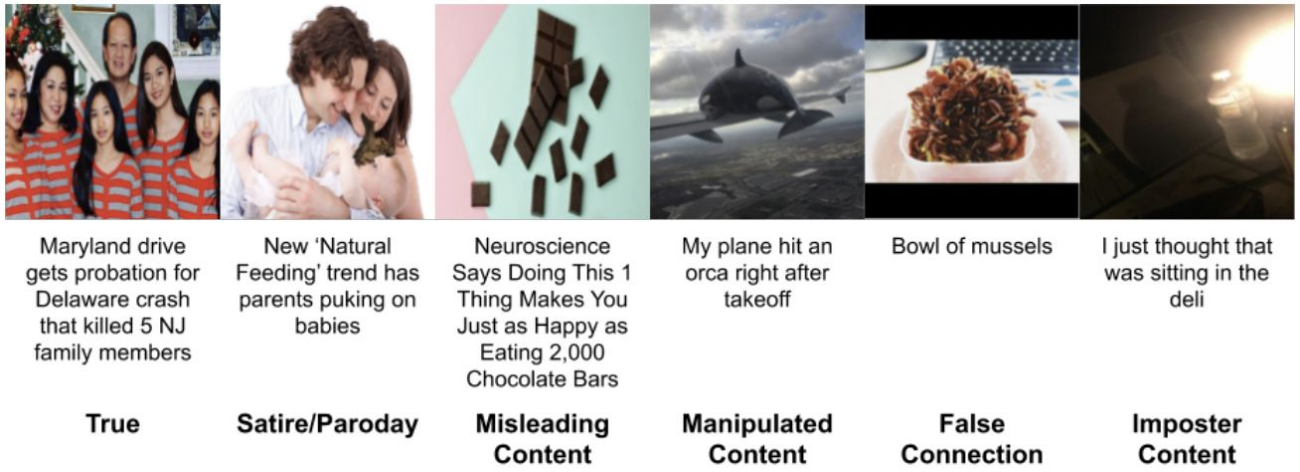


[Qi et al., 2019]

Question: *How can we use the context of one modality (text) to detect fake news and determine if images have been tampered with?*

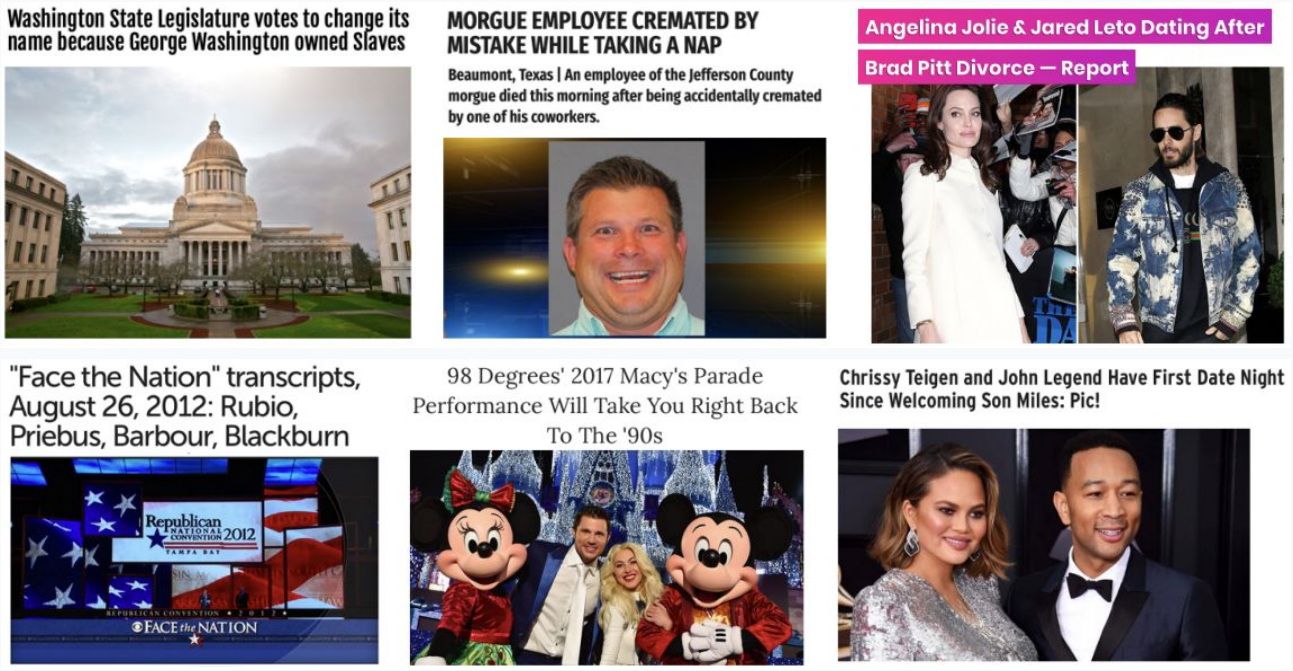
r/Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection

Nakamura et al., 2020



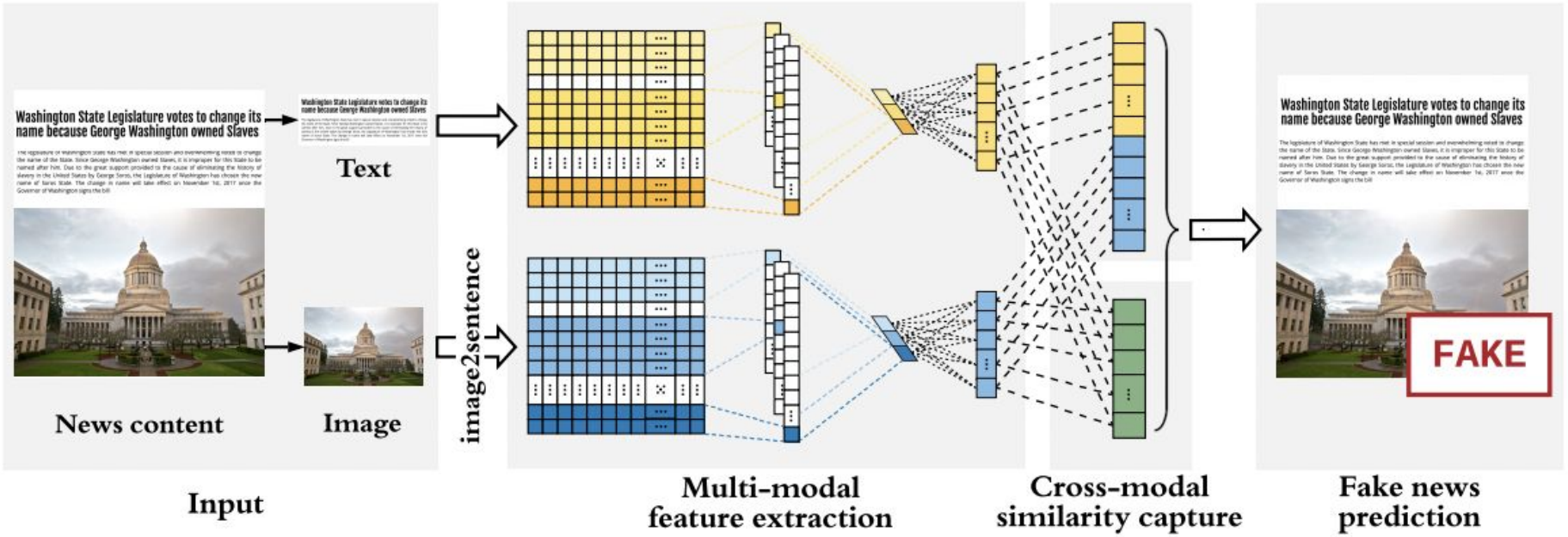
Question: *How can we use the context of one modality (text) to detect fake news and determine if images have been tampered with?*

SAFE: Similarity-Aware Multi-Modal Fake News Detection



[Zhou et al., 2020]

Question: *How can we use the context of one modality (text) to detect fake news and determine if images have been tampered with?*



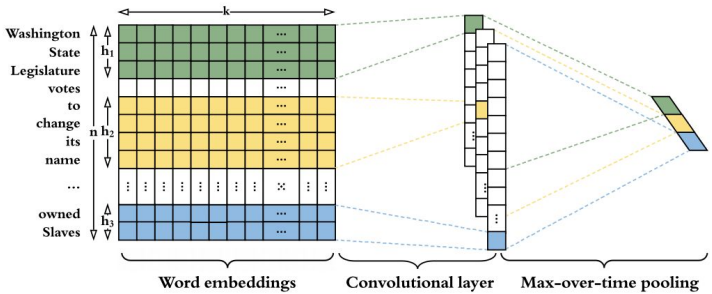


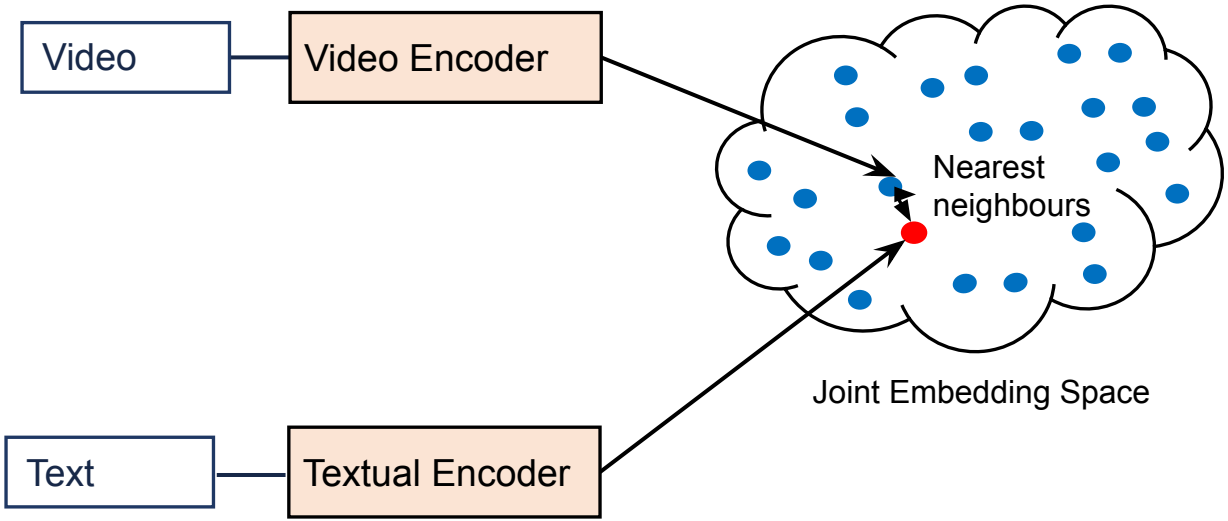
Fig. 2. Text-CNN Architecture

07

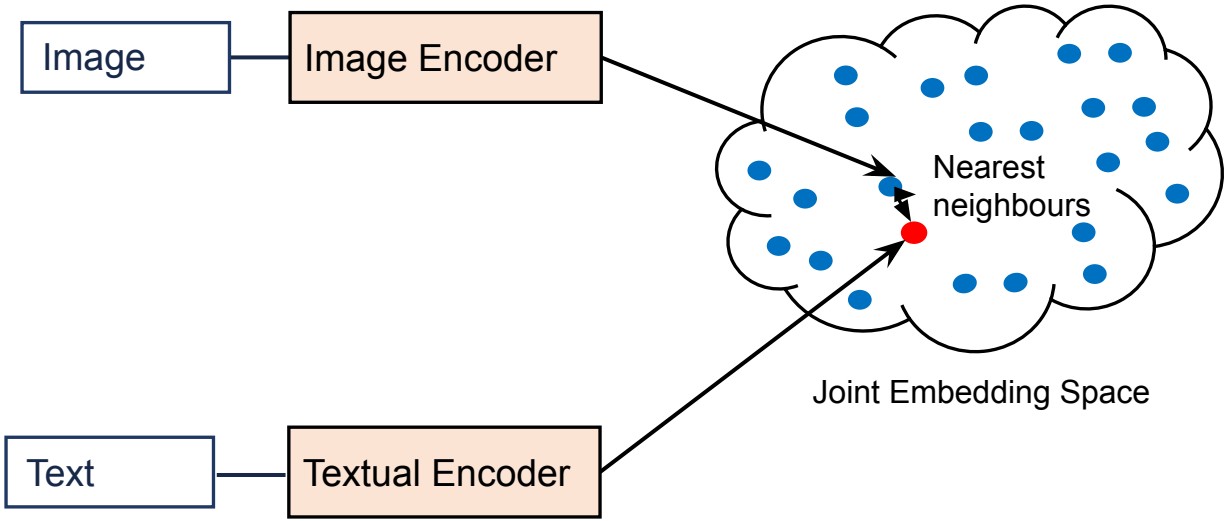
Multimodal entailment

Multimodal models for entailment inferences

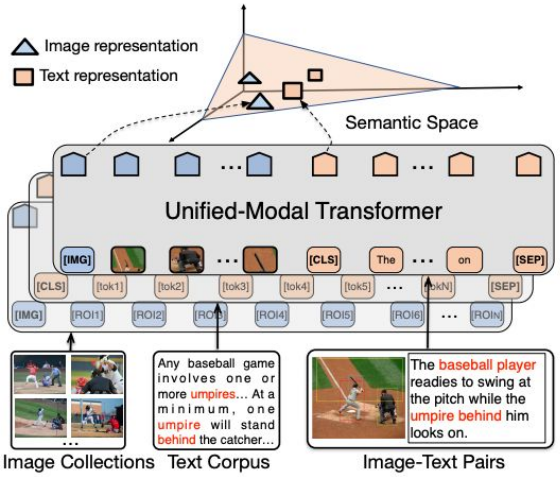
Joint video-text embedding

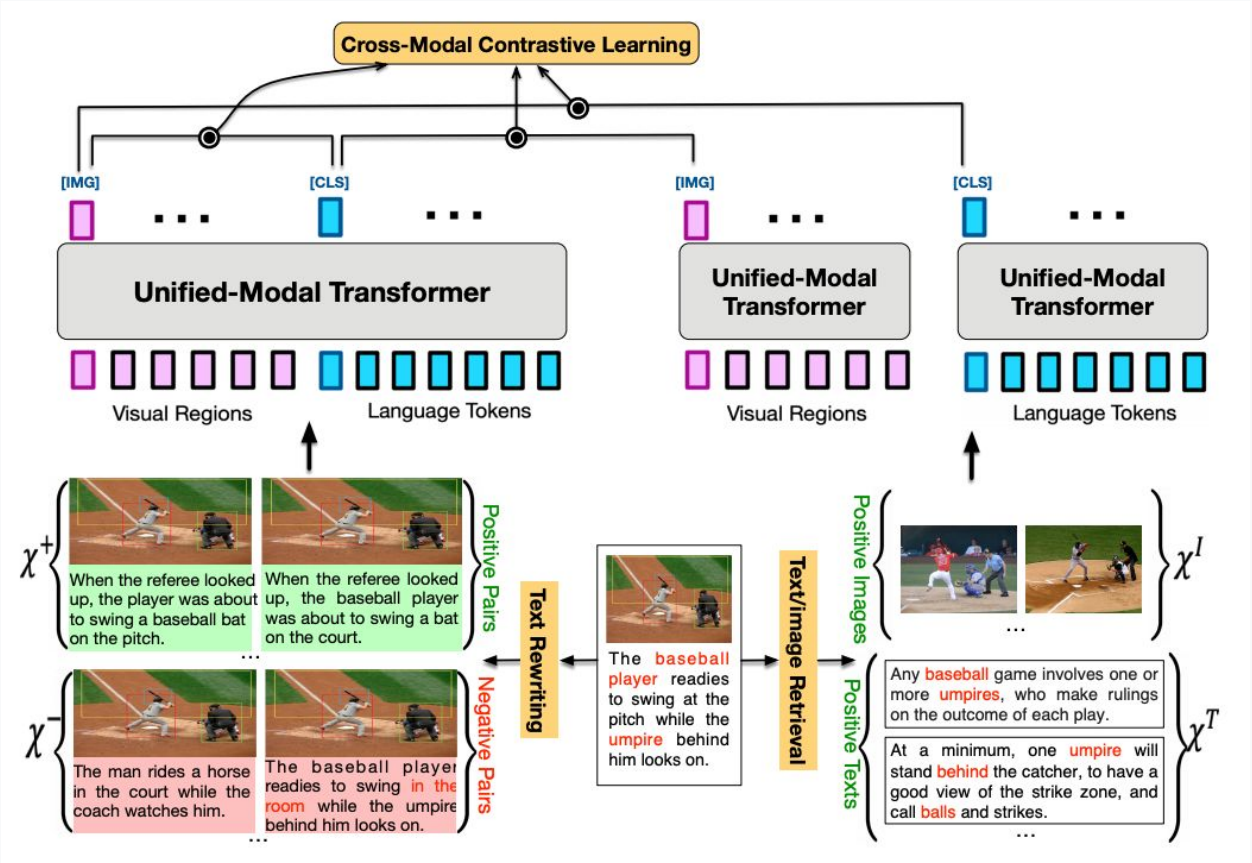


Joint image-text embedding



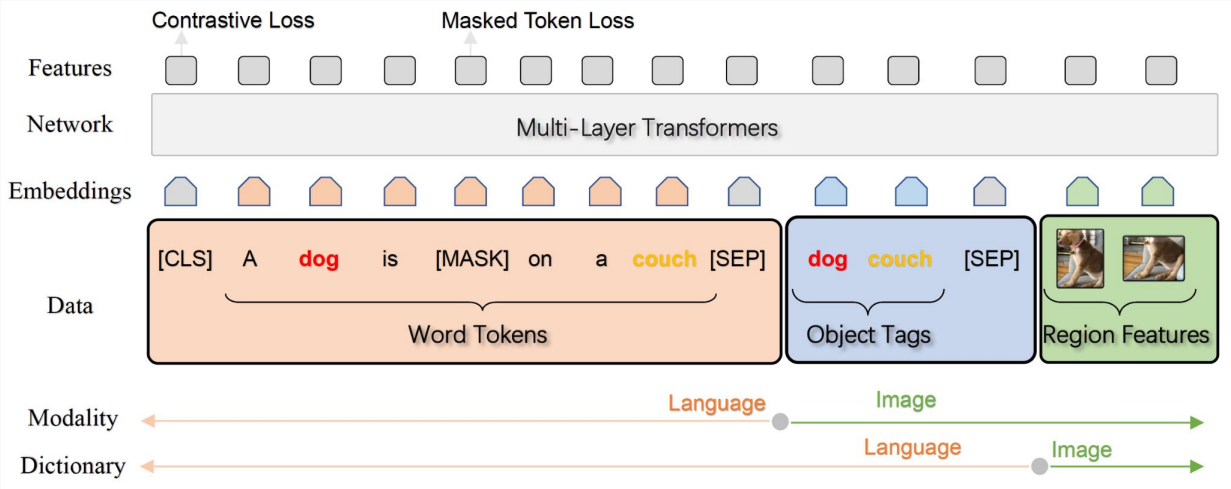
Contrastive Learning





UNIMO: Towards Unified-Modal Understanding and Generation via Cross-Modal Contrastive Learning, Li et al., 2021

Contrastive Learning

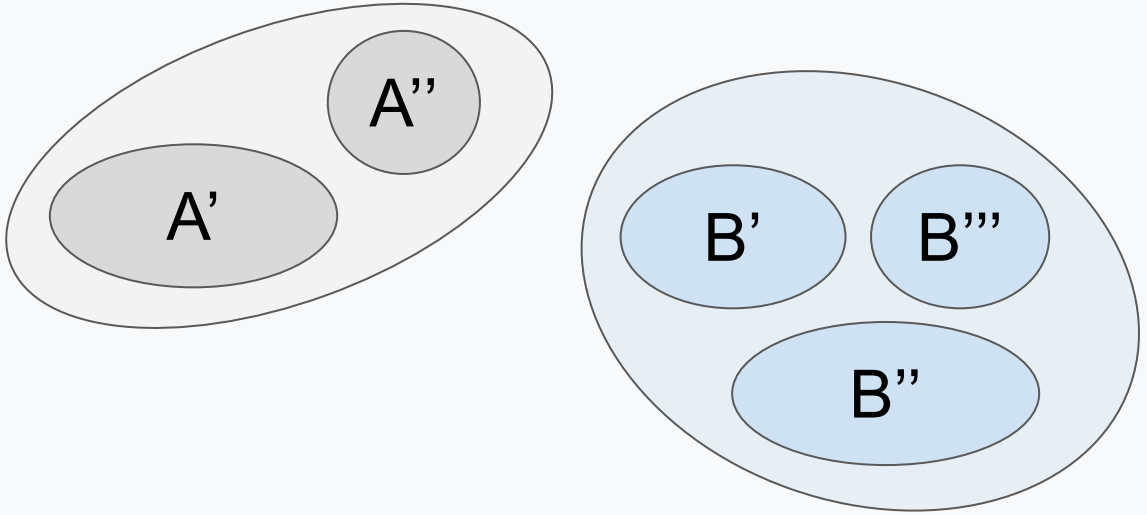


Loss w'/q'	$(w, q/q', v)$ All q's (OSCAR)	 q's from QA	$(w/w', q, v)$ All w's	3-way contrastive All (OSCAR+)	 q's from QA
VQA (vqa-dev)	**69.8**±0.08	**70.1**±0.08	69.5±0.05	**69.8**±0.06	**69.7**±0.06
COCO-IR	73.9±0.2	**75.0**±0.2	**75.0**±0.7	**78.3**±0.3	**77.7**±0.7

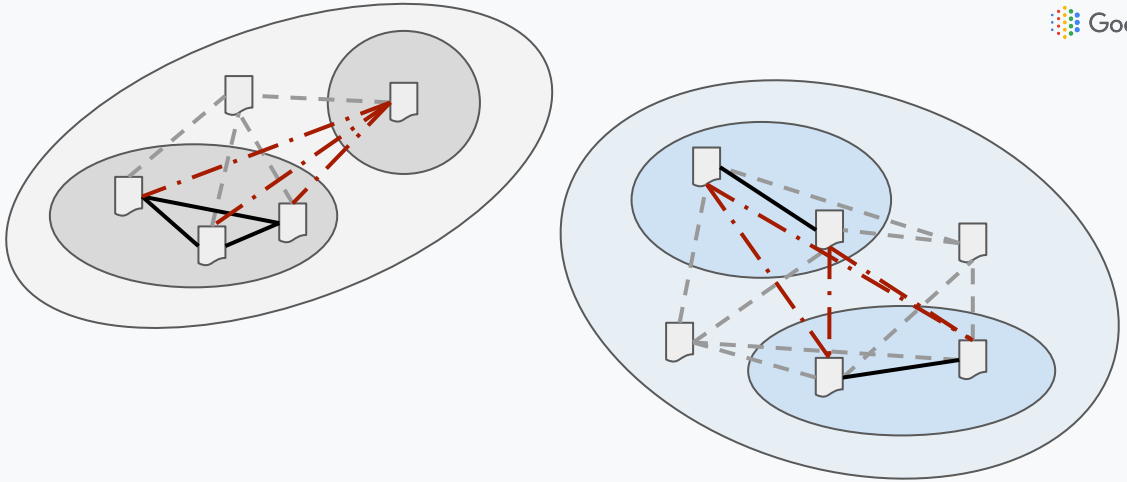
VinVL: Revisiting Visual Representations in Vision-Language Models, Zhang et al., 2021

Clustering & semantic alignments

Topic clusters can be sliced into facets according to their meanings. Subclusters can be created to be semantically cohesive (information support / entailment).



Clustering & semantic alignments

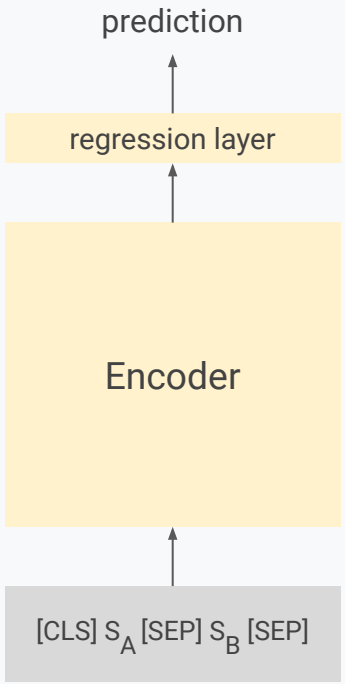


If clusters are fine-grained enough to be about a specific topic and small in size, pairwise comparisons intra-clusters are doable with a single-tower cross-attentional classifier.

Otherwise, sampling strategies can be applied, or the faceting can be learned as part of the latent representation.

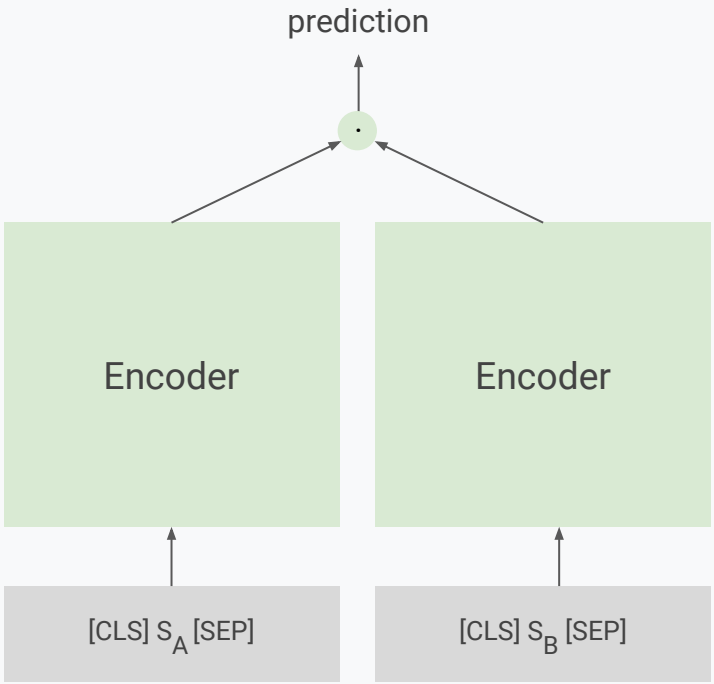
Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

[Reimers et al., 2019]



Cross-attentional

(Single tower)



Dual-encoder

(Two tower)

Efficient retrieval

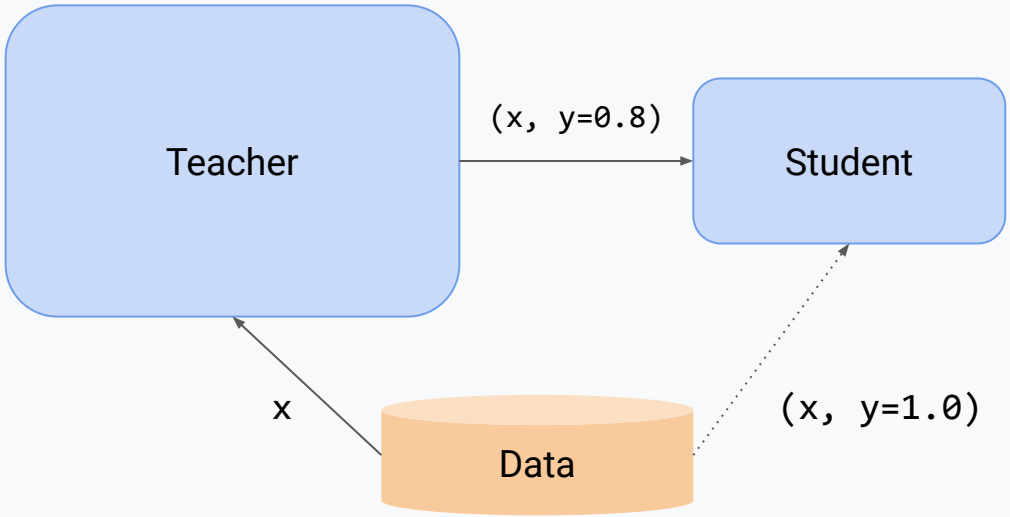
- Example from NLP: finding the most similar sentence in a collection of 10,000 sentences on a V100 GPU
  - BERT (cross-attention): 65 hours
  - SBERT (dual encoder): 5 seconds

- Can also be combined with Maximum Inner Product Search tools for sublinear scaling
  - https://github.com/google-research/google-research/tree/master/scann
  - https://github.com/facebookresearch/faiss
  - https://github.com/spotify/annoy

Knowledge Distillation

Hinton et al., 2015

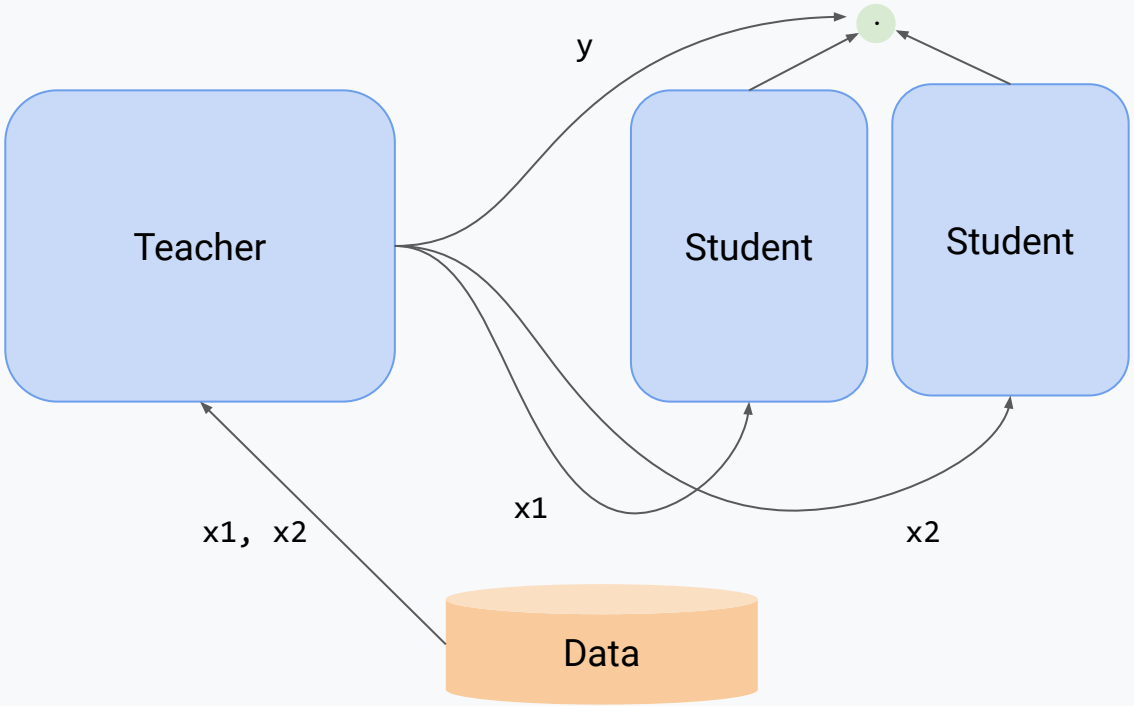
Distilling the Knowledge in a Neural Network



Knowledge Distillation

Hinton et al., 2015

Distilling the Knowledge in a Neural Network



Multimodal entailment dataset

Dataset and colab

https://colab.research.google.com/github/google-research-datasets/recognizing-multimodal-entailment/blob/main/dataset.ipynb

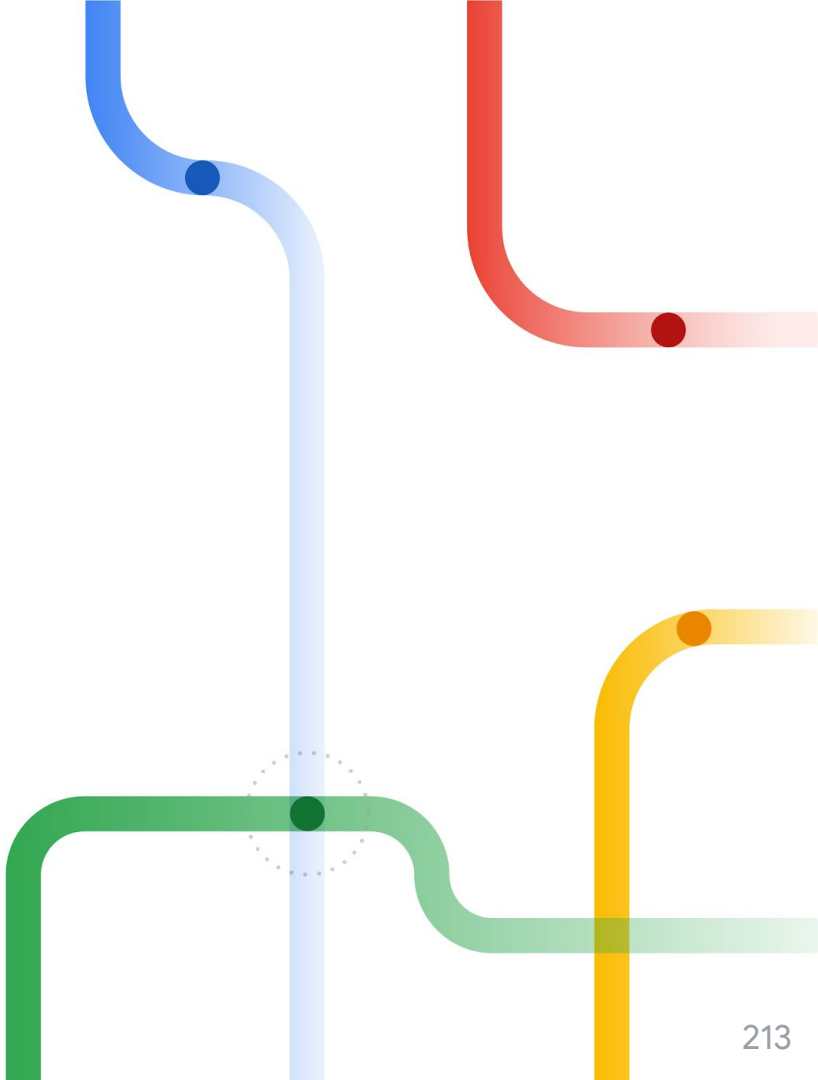
08

Final considerations

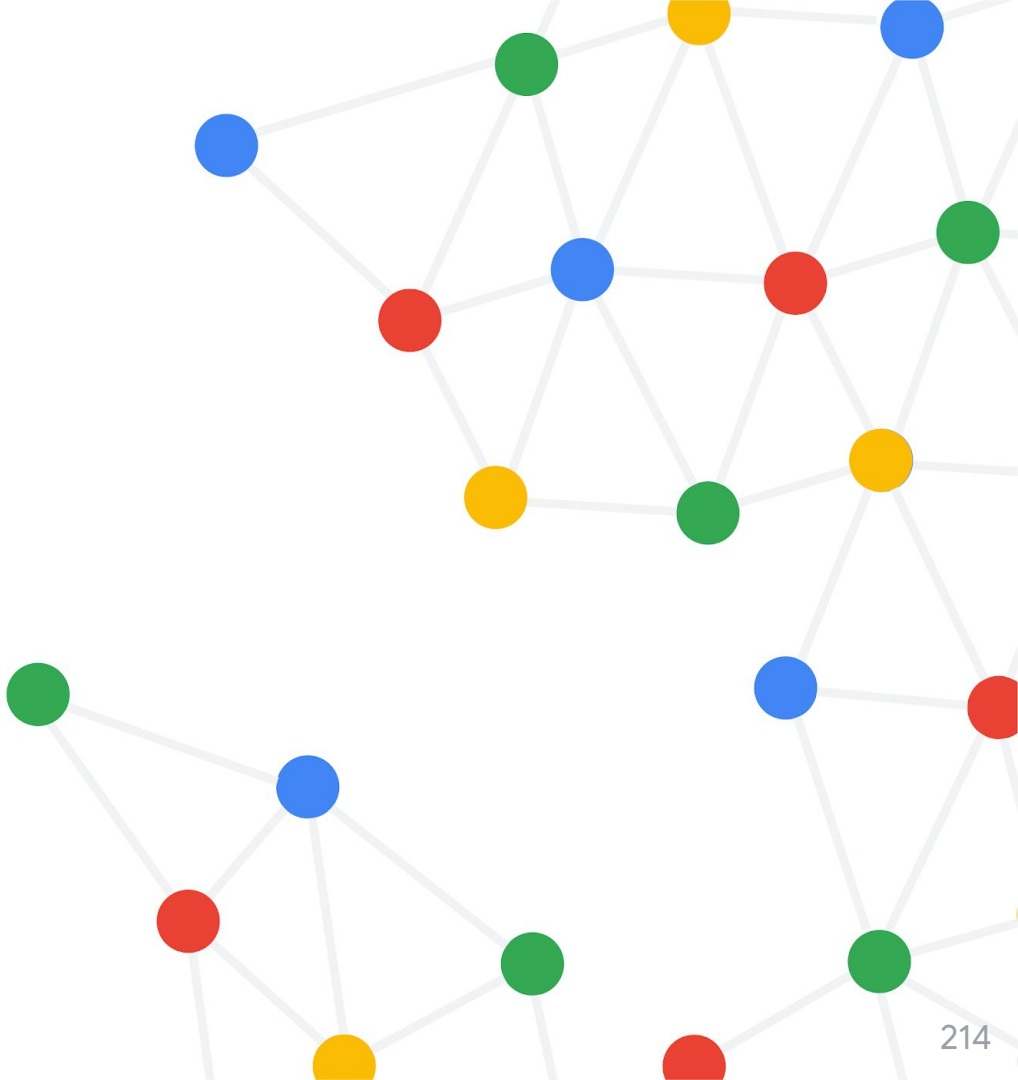
Let's go back to our original questions

- How can we generate representations of so many different kinds of content?
  - Language
  - Knowledge
  - Structured data
  - Vision
  - …
- How do we combine these representations / learn joint representations to provide true multimodal inference about content?
- How can we define the notion of entailment and agreement? How can we use the learned representations to classify whether e.g., two videos or a video and a text "match"?

We hope you enjoyed it and learned something new!

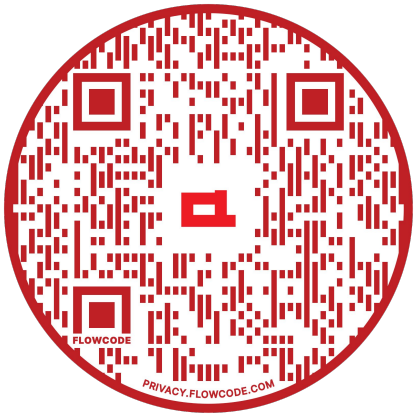


Q&A Time



Thank you!

Available at multimodal-entailment.github.io



BROWN

DeepMind

UNIVERSITY of WASHINGTON

Google Research

UWNLP

IST AUSTRIA
Institute of Science and Technology