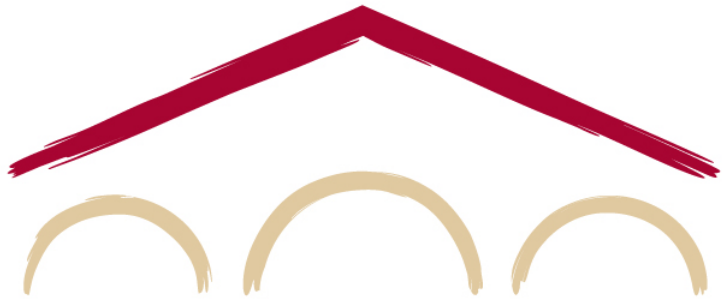
Natural Language Processing with Deep Learning CS224n



Lecture 18: The Future of NLP + Deep Learning

Shikhar Murty

Course Logistics

1. Guest lecture reactions: Due on Friday, March 12 at 11:59PM US-Pacific
2. Great job with the project milestone reports!
3. Finally, final project reports due on Tuesday, March 16 at **4:30 PM** US-Pacific
    1. Hard deadline with late days is Friday, March 19 4:30 PM US-Pacific
    2. For students doing the default final project, last date to submit on the leaderboard is Friday, March 19 4:30 PM US-Pacific.

Lecture Plan

1. Extremely large models and GPT3

Lecture Plan

1. Extremely large models and GPT3
2. Compositional Representations and Systematic Generalization
    1. Are neural representations compositional?
    2. Do neural NLP models generalize systematically?

Lecture Plan

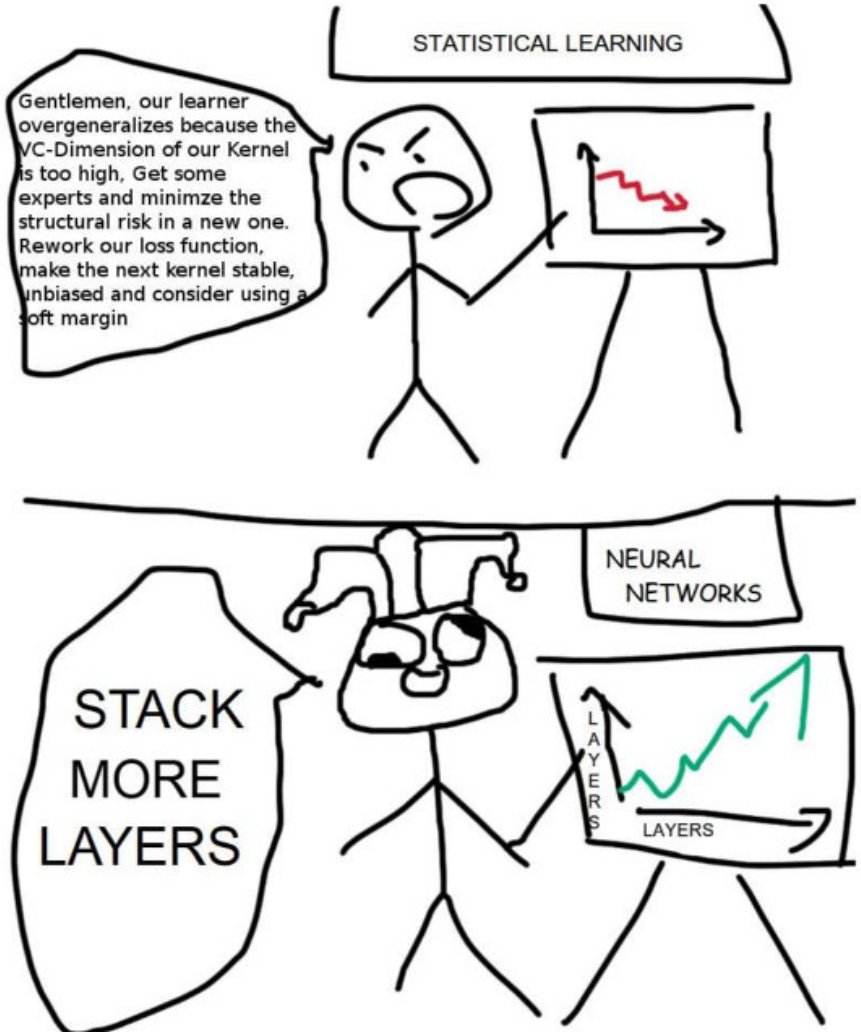
1. Extremely large models and GPT3
2. Compositional Representations and Systematic Generalization
    1. Are neural representations compositional?
    2. Do neural NLP models generalize systematically?
3. Improving how we evaluate models in NLP

Lecture Plan

1. Extremely large models and GPT3
2. Compositional Representations and Systematic Generalization
    1. Are neural representations compositional?
    2. Do neural NLP models generalize systematically?
3. Improving how we evaluate models in NLP
4. Grounding language to other modalities

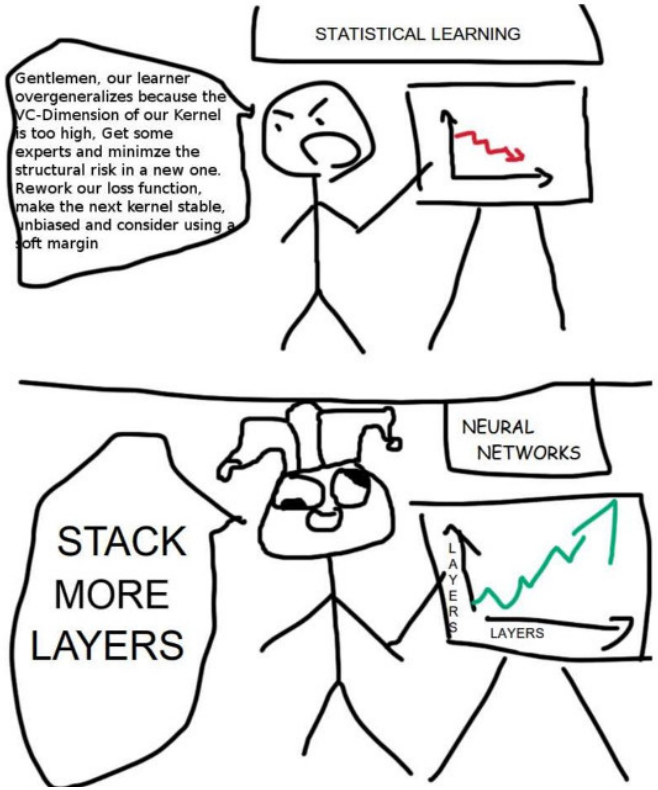
Lecture Plan

1. Extremely large models and GPT3
2. Compositional Representations and Systematic Generalization
    1. Are neural representations compositional?
    2. Do neural NLP models generalize systematically?
3. Improving how we evaluate models in NLP
4. Grounding language to other modalities
5. Getting involved with NLP ∩ Deep Learning Research!



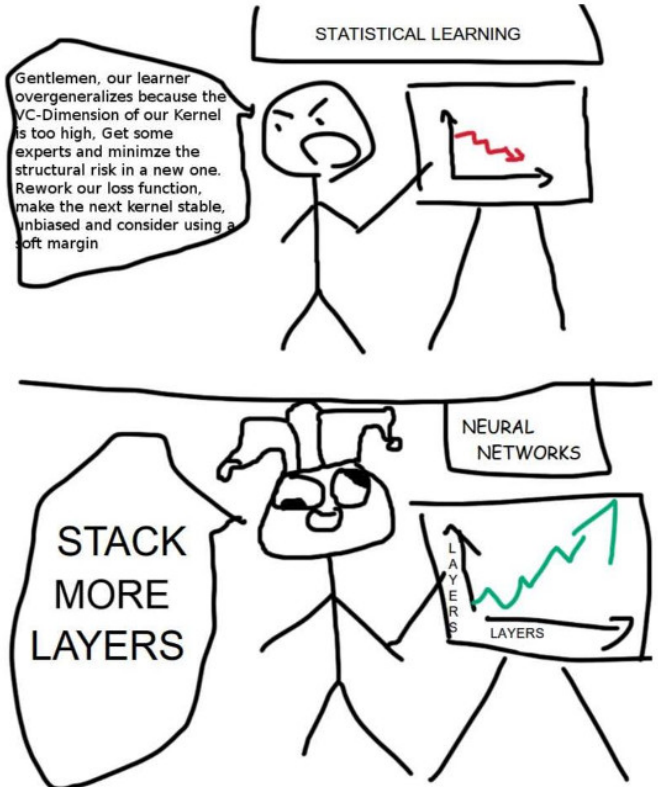
Story of last few years in the NLP ∩ Deep Learning space (almost)

Story of last few years in the Deep Learning + NLP space (almost)



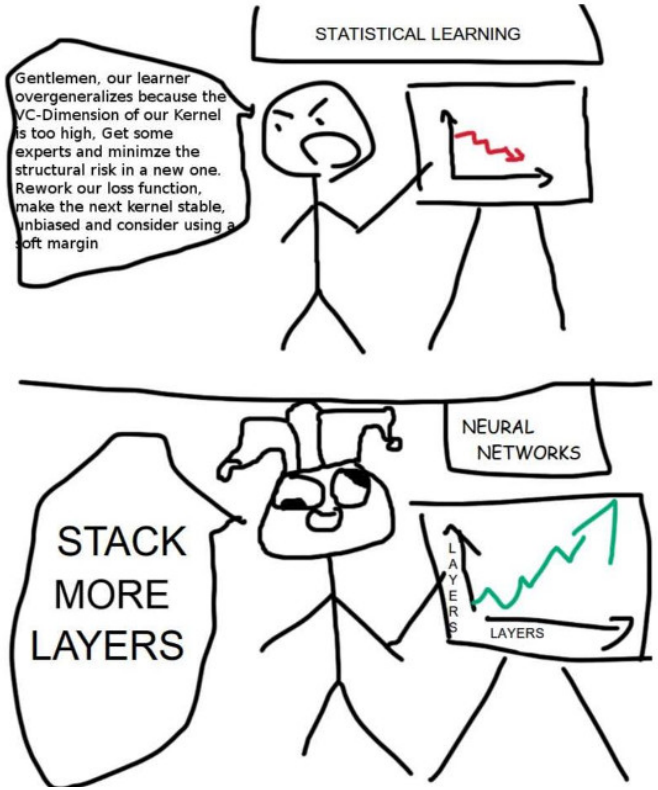
- General Representation Learning Recipe:
  - Convert your data (images, text, videos) into sequence of integers.

Story of last few years in the Deep Learning + NLP space (almost)



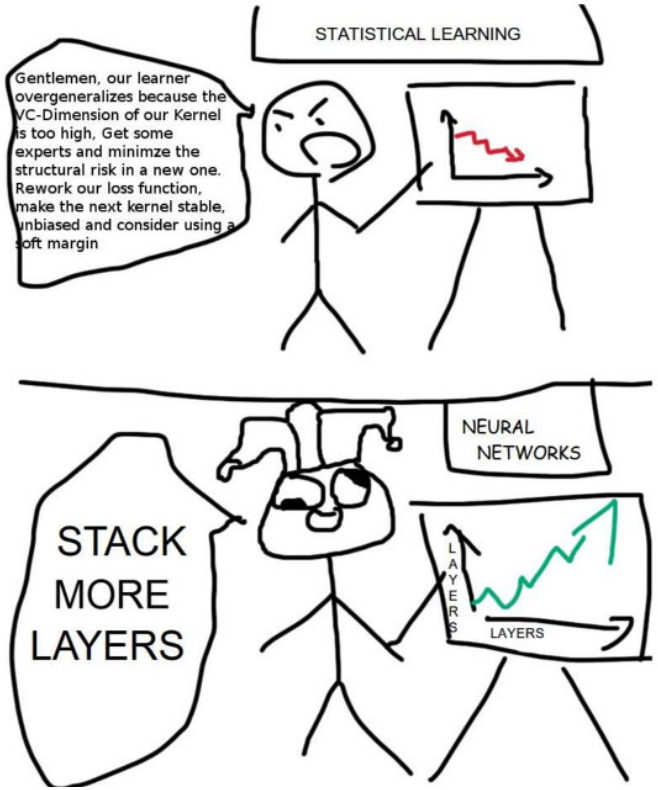
- General Representation Learning Recipe:
  - Convert your data (images, text, videos) into sequence of integers.
  - Define a loss function to maximize data likelihood or create a denoising auto encoder loss.

Story of last few years in the Deep Learning + NLP space (almost)



- General Representation Learning Recipe:
  - Convert your data (images, text, videos) into sequence of integers.
  - Define a loss function to maximize data likelihood or create a denoising auto encoder loss.
  - Train on ***lots*** of data

Story of last few years in the Deep Learning + NLP space (almost)



Certain properties emerge only when we scale up model size!

Q: She just turned 50
A: age

Q: She was going 50 on the highway
A: speed

Prompt

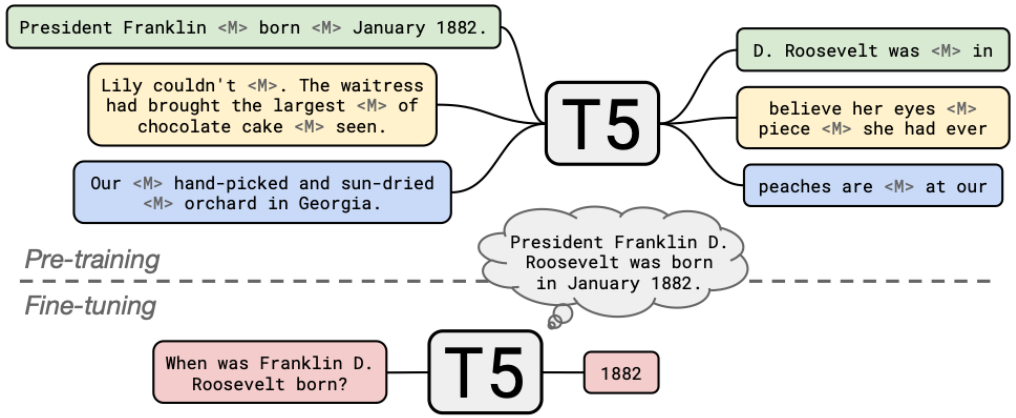
Q: She paid 50 for them.
A: price
Q: Nice donuts! I'll take 50.
A: quantity
Q: I can give you 50 for these, but not more
A: price
Q: I'll take 50 of these, but not more.
A: quantity
Q: I can give you 50 of these, but not more.
A: quantity
Q: I'll take these at 50, but not more.
A: price
Q: I can give you these for 50
A: price
Q: Taking this turn at 50 is dangerous
A: speed

Generated Output

- Examples:
  - GPT-3 (as seen in Lecture-10): Few-shot learning via “in-context” learning

Source: https://twitter.com/yoavgo/status/1284343206474403840/photo/1

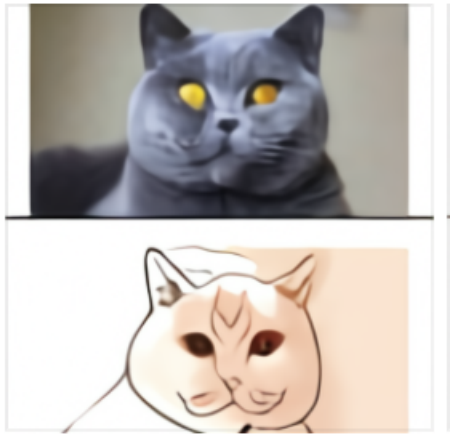
- Examples:
  - GPT-3 (as seen in Lecture-10): Few-shot learning via “in-context” learning
  - T5 (as seen in Lecture-14): Effective closed book QA by storing “knowledge” in its parameters.



Source: https://arxiv.org/abs/2002.08910

TEXT AND IMAGE PROMPT

the exact same cat on the top as a sketch on the bottom



TEXT PROMPT

an armchair in the shape of an avocado

- Examples:
  - GPT-3 (as seen in Lecture-10): Few-shot learning via “in-context” learning
  - T5 (as seen in Lecture-14): Effective closed book QA by storing “knowledge” in its parameters.
  - DALL-E (not covered in this class): Text-to-image model with impressive zero shot generalization

Source: https://openai.com/blog/dall-e/

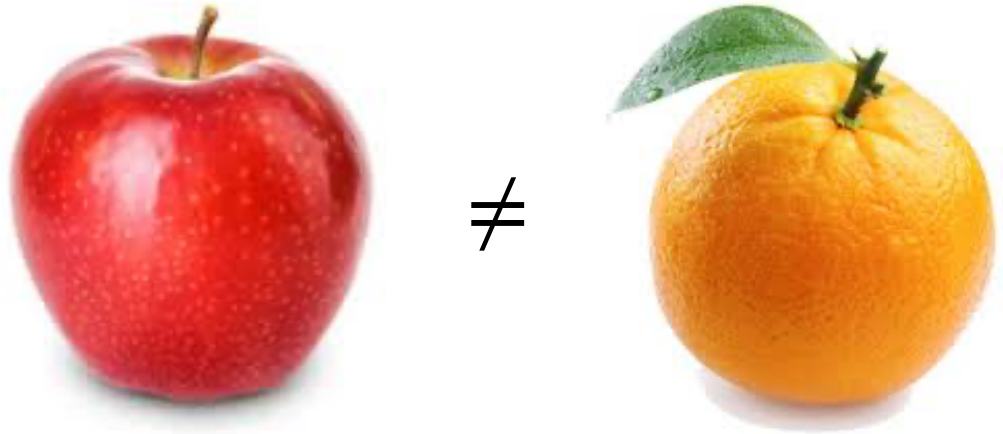
Large Language Models and GPT-3

Large Language Models and GPT-3

Model	# Parameters
Medium-sized LSTM	10M
ELMo	90M
GPT	110M
BERT-Large	320M
GPT-2	1.5B
Honey Bee Brain	~1B synapses
GPT-3	175B
Cat	~10 Trillion synapses
Human	~100 Trillion synapses

Large Language Models and GPT-3

Model	# Parameters
Medium-sized LSTM	10M
ELMo	90M
GPT	110M
BERT-Large	320M
GPT-2	1.5B
Honey Bee Brain	~1B synapses
GPT-3	175B
Cat	~10 Trillion synapses
Human	~100 Trillion synapses



Large Language Models and GPT-3

- 175 BILLION parameters ($n_{layers} = 96$, $n_{heads} = 96$, $d_{heads} = 128$)

Large Language Models and GPT-3

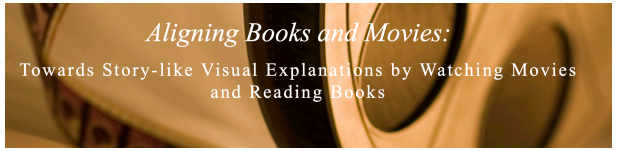
- 175 BILLION parameters ($n_{layers} = 96$, $n_{heads} = 96$, $d_{heads} = 128$)
- Same architecture as GPT-2 with the exception of locally banded sparse attention patterns

Large Language Models and GPT-3

- 175 BILLION parameters ($n_{layers} = 96$, $n_{heads} = 96$, $d_{heads} = 128$)
- Same architecture as GPT-2 with the exception of locally banded sparse attention patterns
- Training details:
  - Trained on 500 billion tokens from:

WIKIPEDIA
The Free Encyclopedia

Common Crawl



What's new about GPT-3?

- Better than other models at language modeling and related tasks such as story completion

Model	PTB (ppl)	LAMBADA (ppl)	LAMBADA (acc)
GPT-2	35.8	8.6	68.0
GPT-3	20.5	3.0	76.2
GPT-3 Few-Shot	_	1.92	86.4

- Lambada involves completing a story by filling in the last word, and a language model doesn't know this:

✓ Alice was friends with Bob. Alice went to visit her friend Bob

✗ Alice was friends with Bob. Alice went to visit her friend and took a…

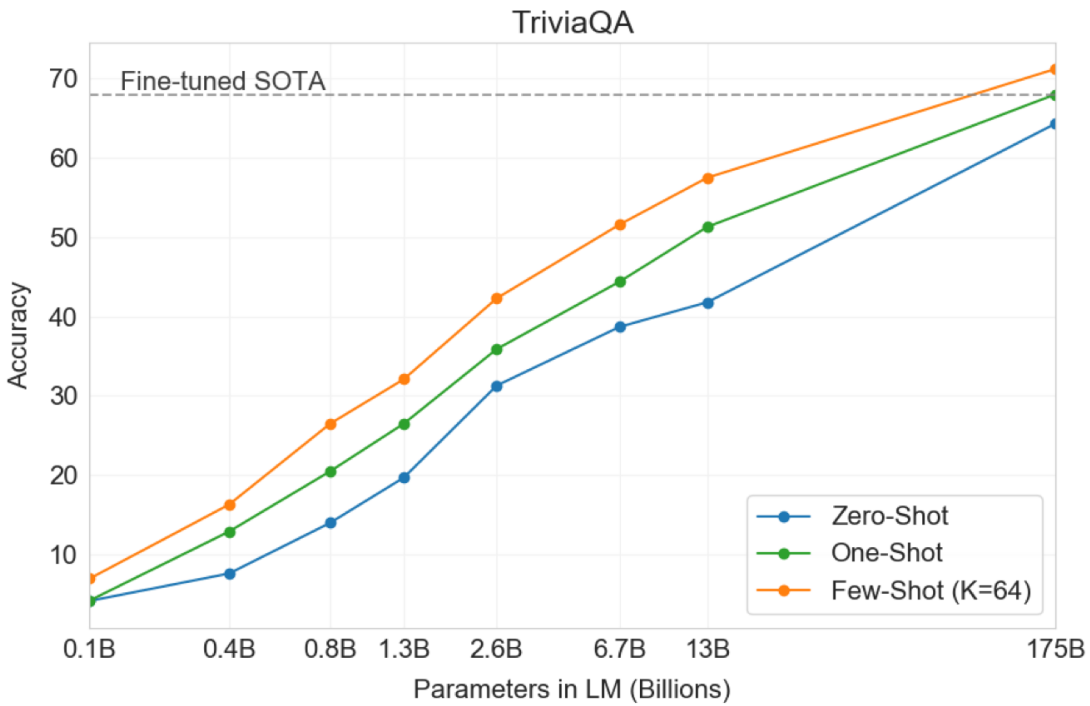
- In the few-shot setting, we can clarify this by providing an example as a prompt:

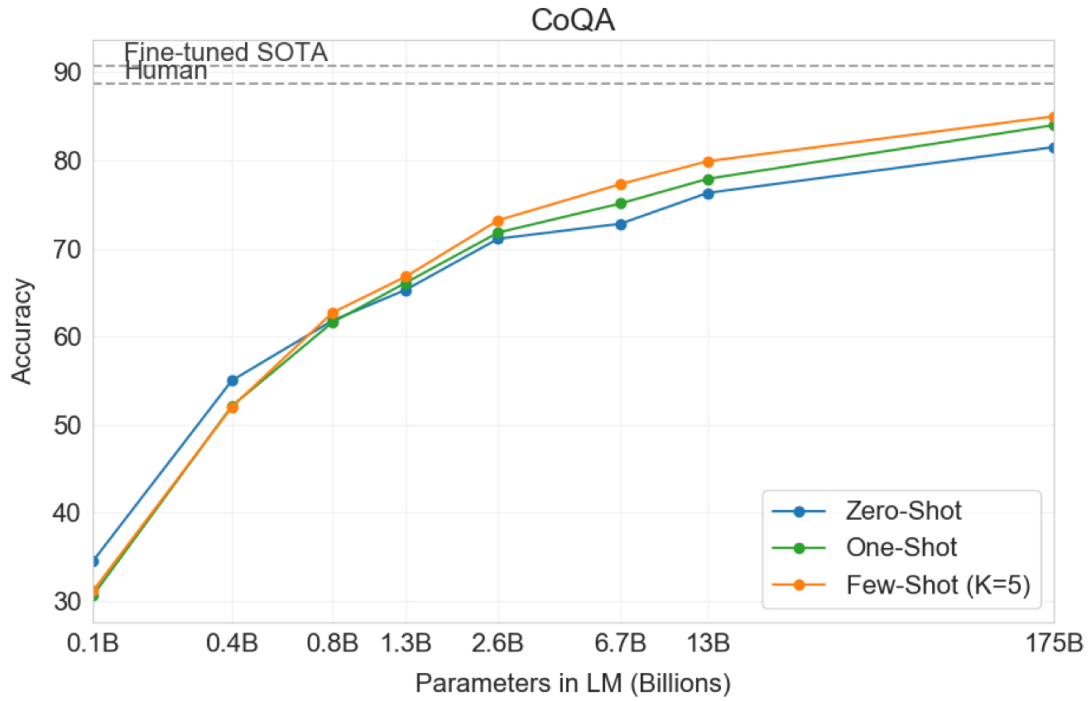
The man went to the park. He was happy to leave his ___ . → house

Alice was friends with Bob. Alice went to visit her friend ___ . →

What's new about GPT-3?

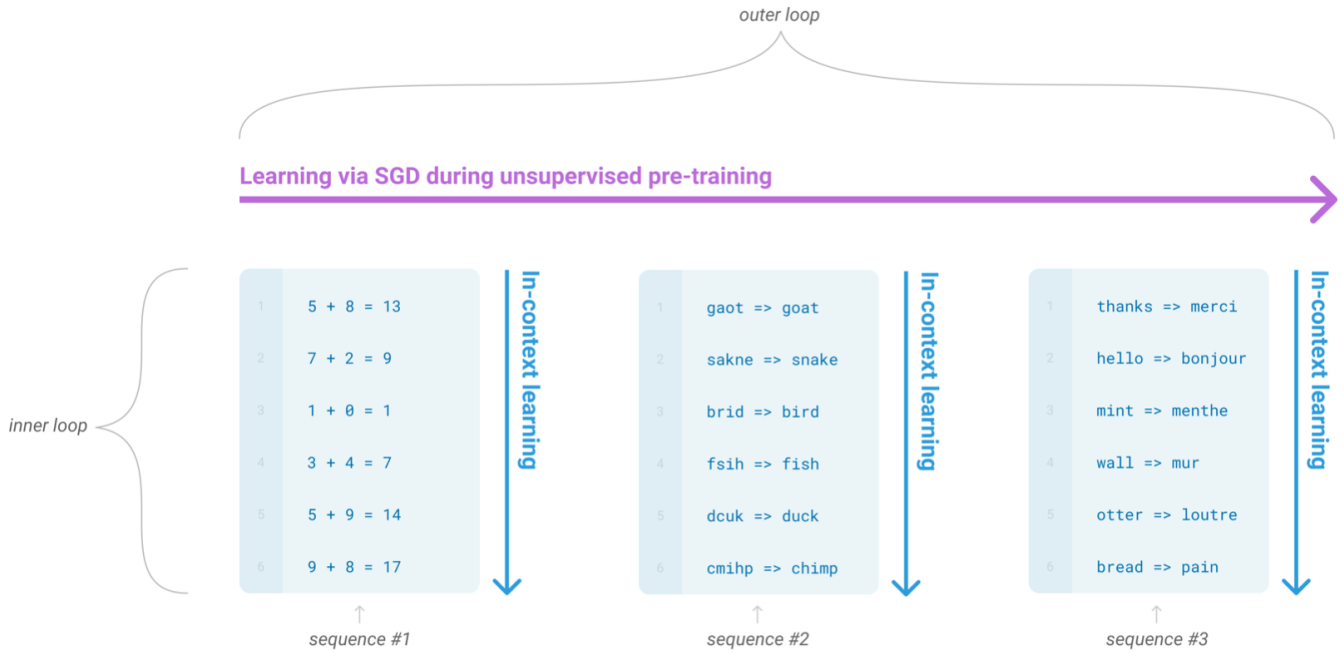
- Apart from better language modeling, scaling also helps with knowledge intensive tasks and reading comprehension





What's new about GPT-3: Flexible "in-context" learning

- GPT-3 demonstrates some level of fast adaptation to completely new tasks. This happens via "in-context" learning
- The language model training (outer loop) is learning how to learn from the context [inner loop]



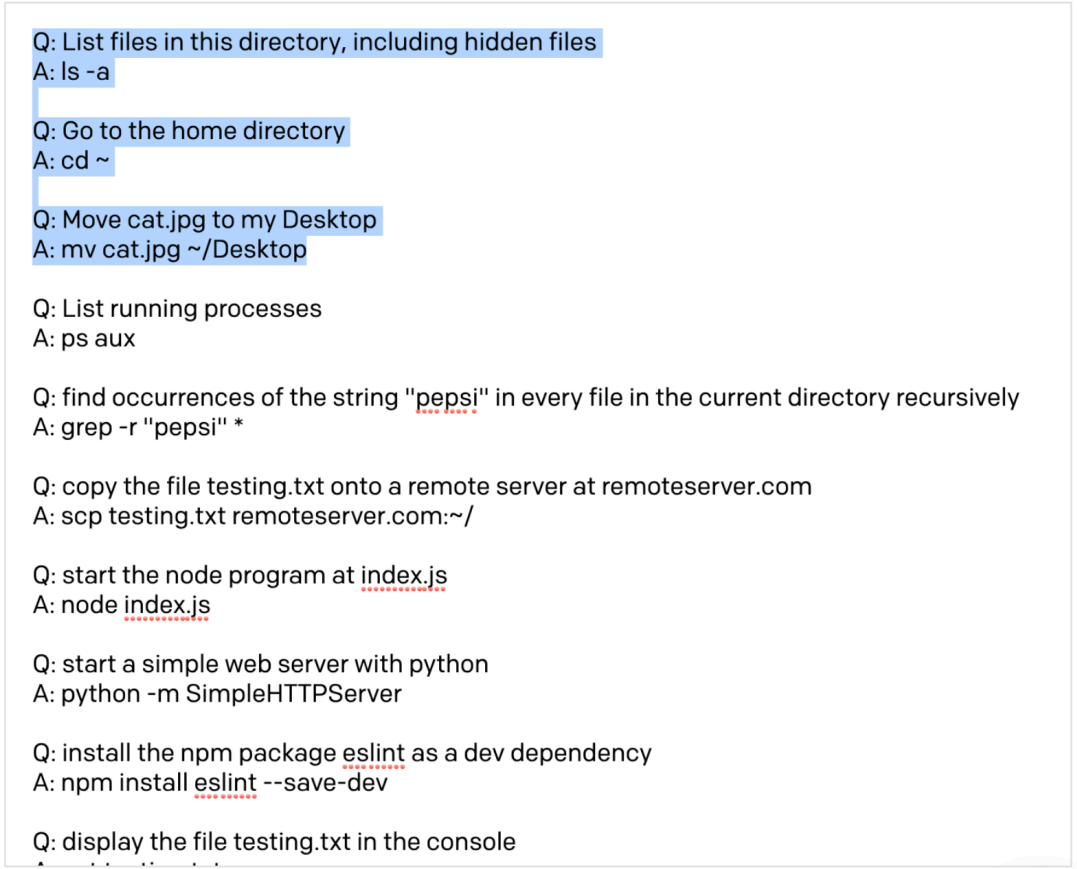
Source: https://arxiv.org/pdf/2005.14165.pdf

Playing with GPT-3!

GPT-3 is currently released only via an *inference* API. Since it's release, there have been many interesting applications that demonstrate its flexible few-shot learning abilities

Playing with GPT-3: Natural Language to Bash

- Using GPT-3 to convert natural language commands to bash one-liners



Q: List files in this directory, including hidden files
A: ls -a

Q: Go to the home directory
A: cd ~

Q: Move cat.jpg to my Desktop
A: mv cat.jpg ~/Desktop

Q: List running processes
A: ps aux

Q: find occurrences of the string "pepsi" in every file in the current directory recursively
A: grep -r "pepsi" *

Q: copy the file testing.txt onto a remote server at remoteserver.com
A: scp testing.txt remoteserver.com:~/

Q: start the node program at index.js
A: node index.js

Q: start a simple web server with python
A: python -m SimpleHTTPServer

Q: install the npm package eslint as a dev dependency
A: npm install eslint --save-dev

Q: display the file testing.txt in the console

Source: https://twitter.com/harlandduman/status/1282132804034150400/photo/1

Playing with GPT-3: Databases in Natural Language?

- Using GPT-3 to convert natural language commands to bash one-liners

The database begins knowing nothing.
The database knows everything that is added to it.
The database does not know anything else.
When asked a question, if the answer has been added to the database the database says the answer.
When asked a question, if the answer has not been added the database says it does not know.

Q: Does the database know "What is 2+2?"
A: The database does not know.

Q: Does the database know "What is the capital of France?"
A: The database does not know.

"""Tom is 20 years old""" is added to the database.
Nothing else about Tom is added to the database.

Q: Does the database know where Tom lives?
A: The database does not know.

Q: How does the database respond when Tom's age?
A: The database says "Tom is 20 years old."

Q: How does the database response when asked "What's my age?"
A: The database says "You are not in the database."

Source: This excellent post from Gwern

Playing with GPT-3: Blending Concepts

- Using GPT-3 to blend concepts i.e. find a word *between* 2 given words

Conceptual blending is where two or more existing concepts are blended together to form a new concept. The blending of the concepts is done in a way that is non-arbitrary. A new concept is formed by taking the meaning of one concept and the form of another. For example, the concept of a car and the concept of a plane are blended to form the concept of a carplane. The concept of a carplane is a non-arbitrary combination of the two concepts. The concept of a carplane is a new concept, with a new meaning, and a new form.

Conceptual blending theory is a theory of how concepts are formed. The theory was developed by Fauconnier and Turner (2002) and was inspired by the theory of conceptual metaphor developed by Lakoff and Johnson (1980). Conceptual blending theory is an extension of conceptual metaphor theory, and conceptual metaphor theory is an extension of the theory of categorization developed by Rosch (1975).

Here are some examples of conceptual blending (where sometimes 2 seemingly disparate ideas are blended):

Idea 1: Airplane
Idea 2: Car
Blended Idea: Flying Car: A car that can fly.

Idea 1: Hierarchy
Idea 2: Attire
Blended Idea: Hierarchical attire: In a workplace, a CEO may be wearing a different kind of attire (more expensive) than a director in the company. And a director would probably be wearing more expensive attire than a janitor in the company. So you have a kind of hierarchical attire.

Prompt

Source: This excellent post from Gwern

Playing with GPT-3: Blending Concepts

Idea 1: Two-Dimensional Space
Idea 2: Three-Dimensional Space
Blended Idea: 2.5-Dimensional Space

Idea 1: Old
Idea 2: New
Blended Idea: Recycled

Idea 1: Geology
Idea 2: Neurology Blended Idea: Sediment neurology: When sediments are buried deep under the ground, they can act like a soft layer of neurological tissue in that they will be compressed over time and can get folded and distorted. In neurology, this is called cortical folding. In geology, this is called tectonic folding.

Idea 1: Mathematics
Idea 2: Fecal matter
Blended Idea: Brownian motion: Brownian motion is a type of random motion that happens when you observe the movements of fecal matter (such as protozoans) under a microscope.

Idea 1: More
Idea 2: Less
Blended Idea: Just-Right

Idea 1: Triangle
Idea 2: Square
Blended Idea: Trapezoid

Some generated samples

Source: This excellent post from Gwern

Playing with GPT-3: Copycat analogy problems

- Copycat analogy problems - maybe it works?

Q: If a b c changes to a b d what does p q r change to?
A: p q r changes to p q s

Q: If a b c changes to a b d what does p q r s change to?
A: p q r s changes to p q r t

Q: If a b c changes to a b d what does p p q q r r change to?
A: p p q q r r changes to p p q q s s

Q: If a b c changes to a b d what does p p p q q q r r r change to?
A: p p p q q q r r r changes to p p p q q q s s s

Q: if a b c changes to a b d what does i j k change to?
A: i j k changes to i j l

Q: If a b c changes to a b d what does i i j j k k change to?
A: i i j j k k changes to i i j j l l

Q: If a b c changes to a b d what does x y z change to?
A: x y z changes to x z

GPT-3 output in red

Source: https://medium.com/@melaniemitchell.me/can-gpt-3-make-analogies-16436605c446

Playing with GPT-3: Copycat analogy problems

- Not so fast!

Q: If a b c changes to a b d , what does i i j j k k change to?
A: i i j j l l

Q: If a b c changes to a b d , what does m m n n o o p p change to?
A: m m n n o o q q

Q: If a b c changes to a b d , what does e e f f change to?
A: e e g g

Q: if a b c changes to a b d , what does r r r r s s s s t t t t change to?

Source: https://medium.com/@melaniemitchell.me/can-gpt-3-make-analogies-16436605c446

Playing with GPT-3: Copycat analogy problems

- Not so fast!

Q: If a b c changes to a b d , what does i i j j k k change to?
A: i i j j l l

Q: If a b c changes to a b d , what does m m n n o o p p change to?
A: m m n n o o q q

Q: If a b c changes to a b d , what does e e f f change to?
A: e e g g

Q: if a b c changes to a b d , what does r r r r s s s s t t t t change to?

GPT-3's answers:

r r r s s s t t
r r r s s t t
r r r s s t t t
r r s s t t
r r r r s s t t t t

Source: https://medium.com/@melaniemitchell.me/can-gpt-3-make-analogies-16436605c446

GPT-3: Limitations and Open questions

- Seems to do poorly on more structured problems that involve decomposing into atomic / primitive skills:
  - RTE / arithmetic / word problems / analogy making

GPT-3: Limitations and Open questions

- Seems to do poorly on more structured problems that involve decomposing into atomic / primitive skills:
  - RTE / arithmetic / word problems / analogy making
- Performing permanent knowledge updates interactively is not well studied.

GPT-3: Limitations and Open questions

- Seems to do poorly on more structured problems that involve decomposing into atomic / primitive skills:
  - RTE / arithmetic / word problems / analogy making
- Performing permanent knowledge updates interactively is not well studied.
- Doesn’t seem to exhibit human like generalization (systematicity).

GPT-3: Limitations and Open questions

- Seems to do poorly on more structured problems that involve decomposing into atomic / primitive skills:
  - RTE / arithmetic / word problems / analogy making
- Performing permanent knowledge updates interactively is not well studied.
- Doesn’t seem to exhibit human like generalization (systematicity).
- Language is situated and GPT-3 is merely learning from text without being exposed to other modalities.

GPT-3: Limitations and Open questions

- Seems to do poorly on more structured problems that involve decomposing into atomic / primitive skills:
  - RTE / arithmetic / word problems / analogy making
- Performing permanent knowledge updates interactively is not well studied.
- **Doesn’t seem to exhibit human like generalization (systematicity).**
- **Language is situated and GPT-3 is merely learning from text without being exposed to other modalities.**

Compositional Representations and Systematic Generalization

Compositional Representations and Systematic Generalization

- Systematicity: The ability to produce/understand some sentences is intrinsically connected to the ability to produce / understand certain others. This means there is a “definite and predictable pattern among the sentences we understand”
- E.g. any speaker that understands the sentence “John loves Mary” should be able to understand “Mary loves John”.

Source: [Stanford Encyclopedia of Philosophy]

Compositional Representations and Systematic Generalization

Compositionality: closely related to the idea of systematicity is the principle of compositionality.

Rough Definition:

"The meaning of an expression is a function of the meaning of its parts"

More concrete definition (Montague):

A homomorphism from syntax (structure) to semantics (meaning). That is, meaning of the whole is a function of immediate constituents (as determined by syntax)

Are human languages compositional?

Brown Cow = $Brown\ objects \cap Cows$

Red Rabbit = $Red\ objects \cap Rabbits$

Kicked the Ball = $Kicked(Ball, Agent)$

Are human languages compositional?

Brown Cow = *Brown objects ∩ Cows*

Red Rabbit = *Red objects ∩ Rabbits*

Kicked the Ball = *Kicked(Ball, Agent)*

Red Herring ≠ *Red things ∩ Herring*

Kicked the bucket ≠ *Kicked(Bucket, Agent)*

Are human languages compositional?

- Nevertheless, compositionality of *representations* is a helpful prior that could lead to systematicity in *behavior.*

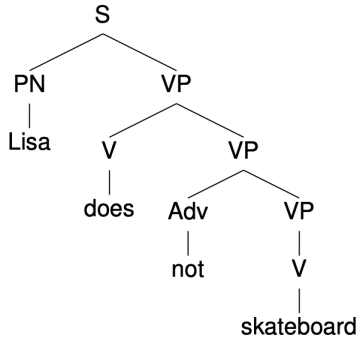
Are human languages compositional?

- Nevertheless, compositionality of *representations* is a helpful prior that could lead to systematicity in *behavior.*
- Questions:
  1. Are neural representations compositional?
  2. Do neural networks generalize systematically?

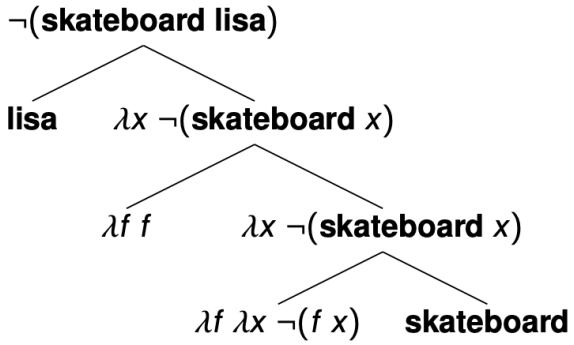
How do we measure if representations from a certain neural network exhibit compositionality?

- According to Montague, Compositionality is about the existence of a homomorphism from syntax to semantics:

Lisa does not skateboard =
⟨Lisa, ⟨does, ⟨not, skateboard⟩⟩⟩



m(Lisa does not skateboard) =
⟨m(Lisa), ⟨m(does), ⟨m(not), m(skateboard)⟩⟩⟩



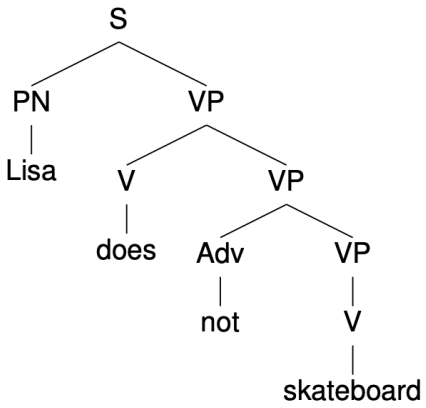
- Tree Reconstruction Error (TRE) [Andreas 2019]: Compositionality of **representations** is about how well the representation approximates an explicitly homomorphic function in *a learnt representation space*

Source: Chris Potts' symsys100 lecture on Montague semantics

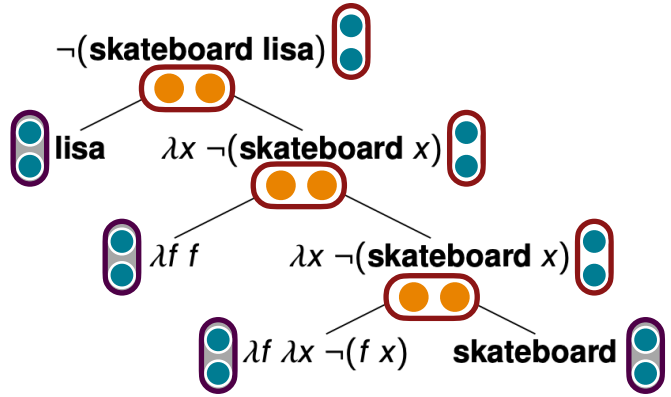
Are neural representations compositional?

- TRE [Andreas 2019]: Compositionality of representations is about how well the representation approximates an explicitly homomorphic function in a learnt representation space

Lisa does not skateboard =
⟨Lisa, ⟨does, ⟨not, skateboard⟩⟩⟩

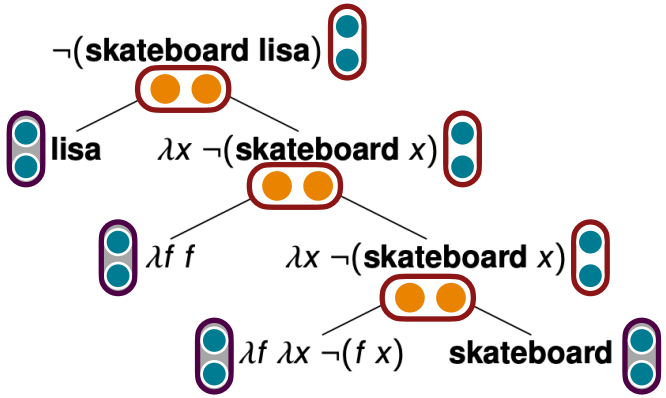


NN(Lisa does not skateboard) ≈
f(v(Lisa), f(v(does), f(v(not), v(skateboard))))



Are neural representations compositional? [Andreas 2019]

NN(Lisa does not skateboard) ≈
f(v(Lisa), f(v(does), f(v(not), v(skateboard))))



leaf vectors as well as the composition operator are *learnt by TRE*

Tree Reconstruction Error (TRE)

First choose :

- a distance function $\delta : \Theta \times \Theta \to [0, \infty)$ satisfying $\delta(\theta, \theta') = 0 \Leftrightarrow \theta = \theta'$
- a composition function $* : \Theta \times \Theta \to \Theta$

Define $\hat{f}_\eta(d)$, a *compositional approximation to* f with parameters η, as:

$$\hat{f}_\eta(d_i) = \eta_i \qquad \text{for } d_i \in \mathcal{D}_0$$
$$\hat{f}_\eta(\langle d, d' \rangle) = \hat{f}_\eta(d) * \hat{f}_\eta(d') \qquad \text{for all other } d$$

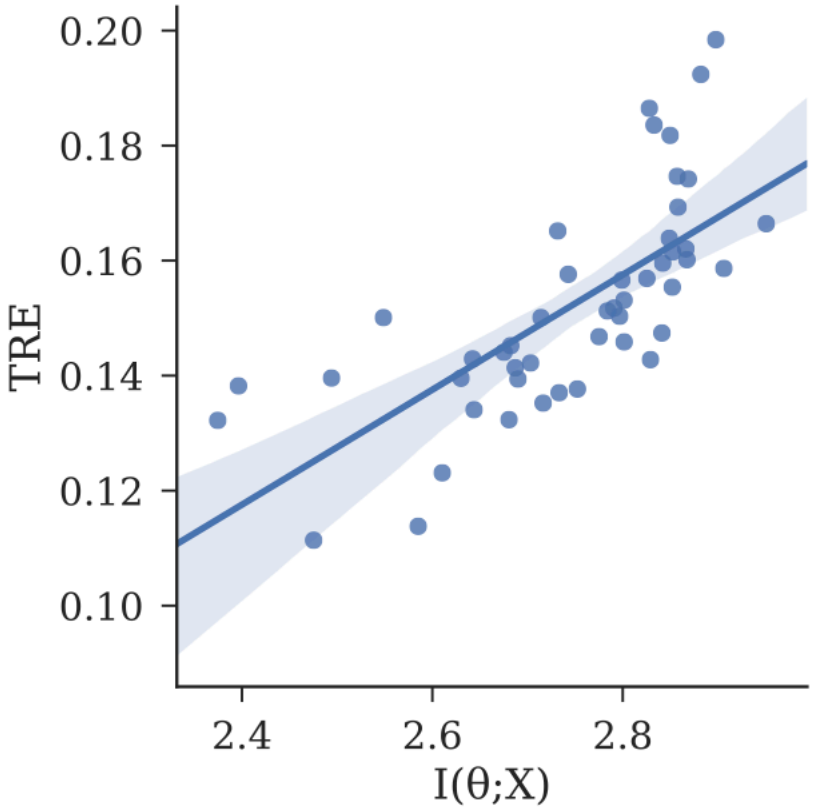
Given a dataset $\mathcal{X}$ of inputs x_i with derivations $d_i = D(x_i)$, compute:

$$\eta^* = \arg\min_\eta \sum_i \delta\big(f(x_i), \hat{f}_\eta(d_i)\big)$$

Then we can define datum- and dataset-level evaluation metrics:

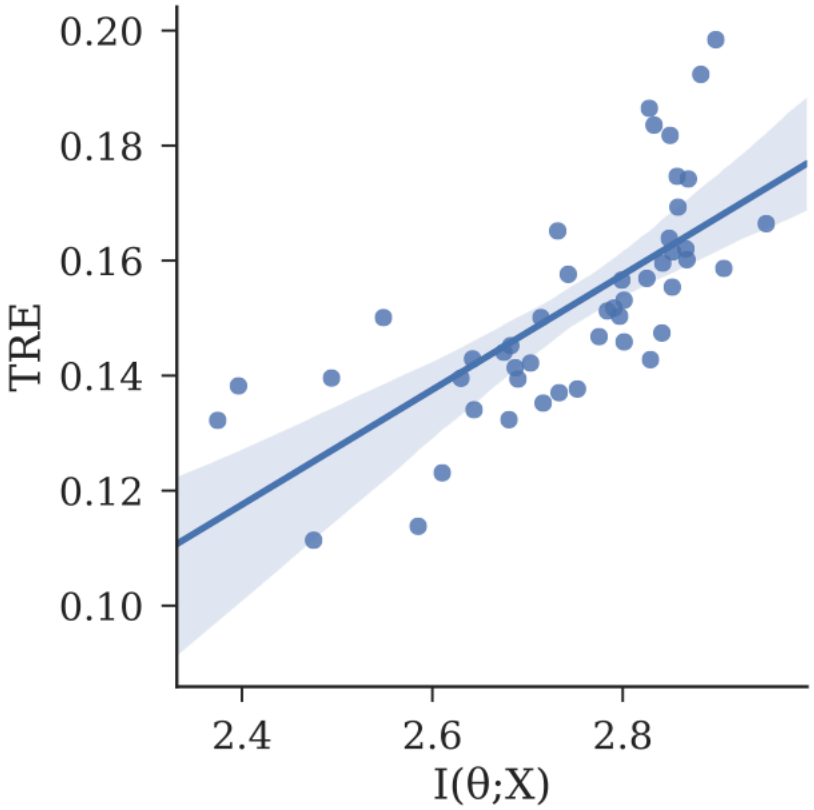
$$\text{TRE}(x) = \delta\big(f(x), \hat{f}_{\eta^*}(d)\big)$$
$$\text{TRE}(\mathcal{X}) = \frac{1}{n} \sum_i \text{TRE}(x_i)$$

Are neural representations compositional? [Andreas 2019]



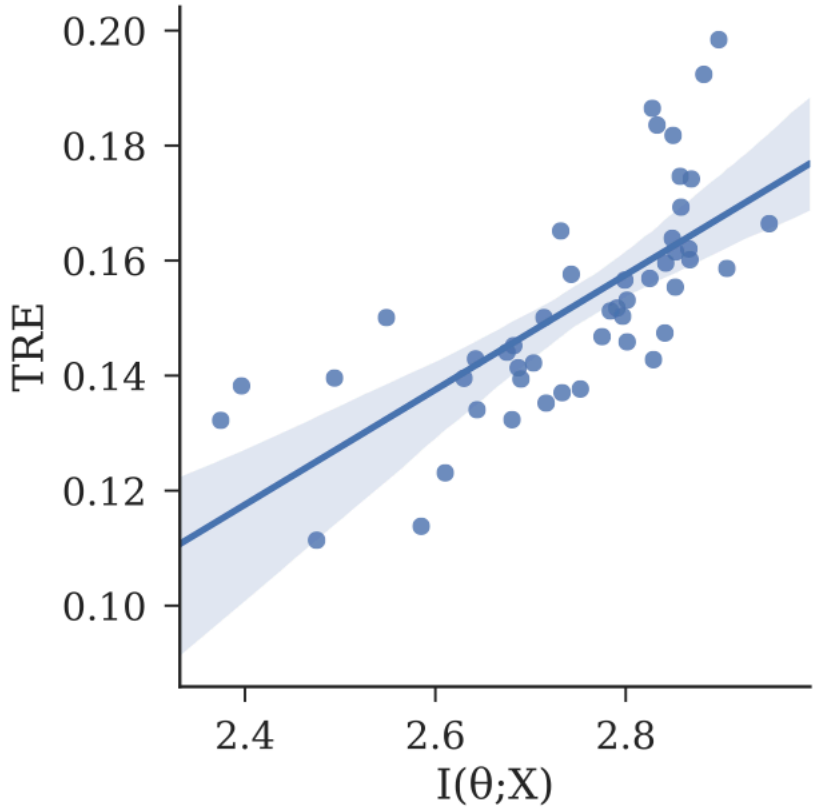
- This graph plots the mutual information between the input and the representation I(θ; X) against TRE.

Are neural representations compositional? [Andreas 2019]



- This graph plots the mutual information between the input and the representation I(θ; X) against TRE.
- As the model learns (characterized by decreasing mutual information), we notice that the representations become more compositional!

Are neural representations compositional? [Andreas 2019]



- This graph plots the mutual information between the input and the representation I(θ; X) against TRE.
- As the model learns (characterized by decreasing mutual information), we notice that the representations become more compositional!
- Overall, we observe that learning is correlated with increased compositionality as measured by TRE!

Do neural networks generalize systematically?

- Do neural networks (including large transformers) generalize systematically on challenging benchmarks involving *realistic language*?
- Can we create a dataset split that explicitly tests for this kind of generalization?

Systematicity and Compositional Generalization: Do neural networks generalize systematically?

- Maximize *compound divergence* to create challenging train / test splits!
  - **Atoms**: primitive elements (entity words, predicates)
  - **Compounds**: compositions of primitive elements.

Train:
Did Christopher Nolan produce Goldfinger?
Who directed inception?
Test:
Did Christopher Nolan direct Goldfinger?
Who produced Goldfinger?

Atoms:
produce
direct
inception
goldfinger
Christopher Nolan
Who [predicate] [y]?
Did [x] [predicate] [y]?

Compounds:
Did Christopher Nolan [predicate] Goldfinger?
Who directed [entity]?

Do neural networks generalize systematically?

Train:
- Did Christopher Nolan produce Goldfinger?
- Who directed inception?

Test:
- Did Christopher Nolan direct Goldfinger?
- Who produced Goldfinger?

Atoms:
- produce
- direct
- inception
- goldfinger
- Christopher Nolan
- Who [predicate] [y]?
- Did [x] [predicate] [y]?

Compounds:
- Did Christopher Nolan [predicate] Goldfinger?
- Who directed [entity]?

- Basic Machinery for producing compositionally challenging splits:

Let $\mathcal{F}_A(\text{data}) \equiv$ normalized frequency distribution of atoms

Let $\mathcal{F}_C(\text{data}) \equiv$ normalized frequency distribution of compounds

Define atom and compound divergence as:

Do neural networks generalize systematically?

Train:
- Did Christopher Nolan produce Goldfinger?
- Who directed inception?

Test:
- Did Christopher Nolan direct Goldfinger?
- Who produced Goldfinger?

Atoms:
- produce
- direct
- inception
- goldfinger
- Christopher Nolan
- Who [predicate] [y]?
- Did [x] [predicate] [y]?

Compounds:
- Did Christopher Nolan [predicate] Goldfinger?
- Who directed [entity]?

- Basic Machinery for producing compositionally challenging splits:

Let $\mathcal{F}_A(\text{data}) \equiv$ normalized frequency distribution of atoms

Let $\mathcal{F}_C(\text{data}) \equiv$ normalized frequency distribution of compounds

Define atom and compound divergence as:

$$\mathcal{D}_A(\text{train}||\text{test}) = 1 - C_{0.5}(\mathcal{F}_A(\text{train})||\mathcal{F}_A(\text{test}))$$
$$\mathcal{D}_C(\text{train}||\text{test}) = 1 - C_{0.1}(\mathcal{F}_C(\text{train})||\mathcal{F}_C(\text{test}))$$

Do neural networks generalize systematically?

Train:
- Did Christopher Nolan produce Goldfinger?
- Who directed inception?

Test:
- Did Christopher Nolan direct Goldfinger?
- Who produced Goldfinger?

Atoms:
- produce
- direct
- inception
- goldfinger
- Christopher Nolan
- Who [predicate] [y]?
- Did [x] [predicate] [y]?

Compounds:
- Did Christopher Nolan [predicate] Goldfinger?
- Who directed [entity]?

- Basic Machinery for producing compositionally challenging splits:

Let $\mathcal{F}_A(\text{data}) \equiv$ normalized frequency distribution of atoms

Let $\mathcal{F}_C(\text{data}) \equiv$ normalized frequency distribution of compounds

Define atom and compound divergence as:

$$\mathcal{D}_A\big(\text{train}\,||\,\text{test}\big) = 1 - C_{0.5}\big(\mathcal{F}_A(\text{train})\,||\,\mathcal{F}_A(\text{test})\big)$$
$$\mathcal{D}_C\big(\text{train}\,||\,\text{test}\big) = 1 - C_{0.1}\big(\mathcal{F}_C(\text{train})\,||\,\mathcal{F}_C(\text{test})\big)$$

where,

$$C_\alpha(P\,||\,Q) = \sum_k p_k^\alpha q_k^{1-\alpha}$$

is the chernoff coefficient between two categorical distributions that measures similarity.

Do neural networks generalize systematically?

Train:
- Did Christopher Nolan produce Goldfinger?
- Who directed inception?

Test:
- Did Christopher Nolan direct Goldfinger?
- Who produced Goldfinger?

Atoms:
- produce
- direct
- inception
- goldfinger
- Christopher Nolan
- Who [predicate] [y]?
- Did [x] [predicate] [y]?

Compounds:
- Did Christopher Nolan [predicate] Goldfinger?
- Who directed [entity]?

- Basic Machinery for producing compositionally challenging splits:

Let $\mathcal{F}_A(\text{data}) \equiv$ normalized frequency distribution of atoms

Let $\mathcal{F}_C(\text{data}) \equiv$ normalized frequency distribution of compounds

Define atom and compound divergence as:

$$\mathcal{D}_A\big(\text{train}\,||\,\text{test}\big) = 1 - C_{0.5}\big(\mathcal{F}_A(\text{train})\,||\,\mathcal{F}_A(\text{test})\big)$$
$$\mathcal{D}_C\big(\text{train}\,||\,\text{test}\big) = 1 - C_{0.1}\big(\mathcal{F}_C(\text{train})\,||\,\mathcal{F}_C(\text{test})\big)$$

where,

$$C_\alpha(P\,||\,Q) = \sum_k p_k^\alpha q_k^{1-\alpha}$$

is the chernoff coefficient between two categorical distributions that measures similarity.

Goal:

Split data into train / test such that compound divergence is maximized and atom divergence is minimized!

Do neural networks generalize systematically?

- So do neural networks generalize systematically?
- Furrer 2020: “Pre-training helps for compositional generalization, but doesn’t solve it”

Model	*CFQ (Maximum Compound divergence)*
T5-small (no pretraining)	21.4
T5-small	28.0
T5-base	31.2
T5-large	34.8
T5-3B	40.2
T5-11B	40.9
T5-11B-mod	42.1

Increasing #parameters

Source: Results from Furrer 2020 “Compositional Generalization in Semantic Parsing: Pre-training vs. Specialized Architectures”

Improving how we evaluate models in NLP

Improving how we evaluate models in NLP

While we are making progress in terms of performance on benchmarks, it's unclear if the gains are coming from spurious correlations or real task understanding.

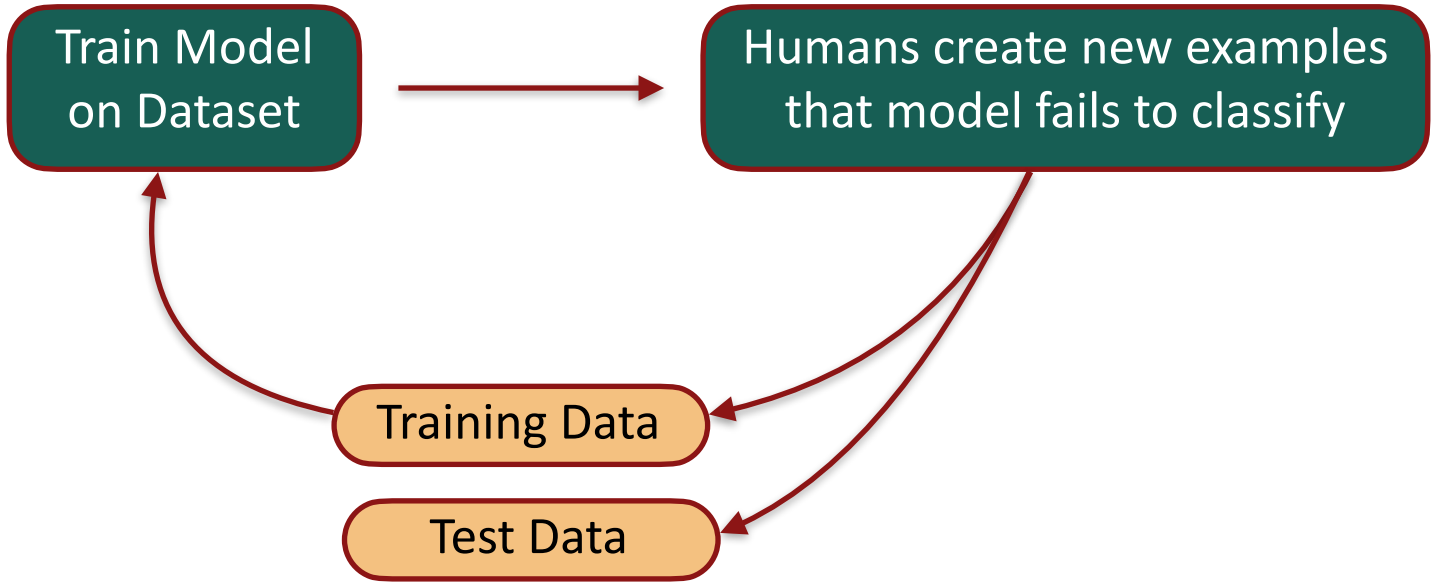
Improving how we evaluate models in NLP

This is because of models exhibiting poor generalization to out-of-distribution samples. How do we ensure that we are accurately measuring task understanding and not overestimating model performance?

Improving how we evaluate models in NLP: Dynamic Benchmarks

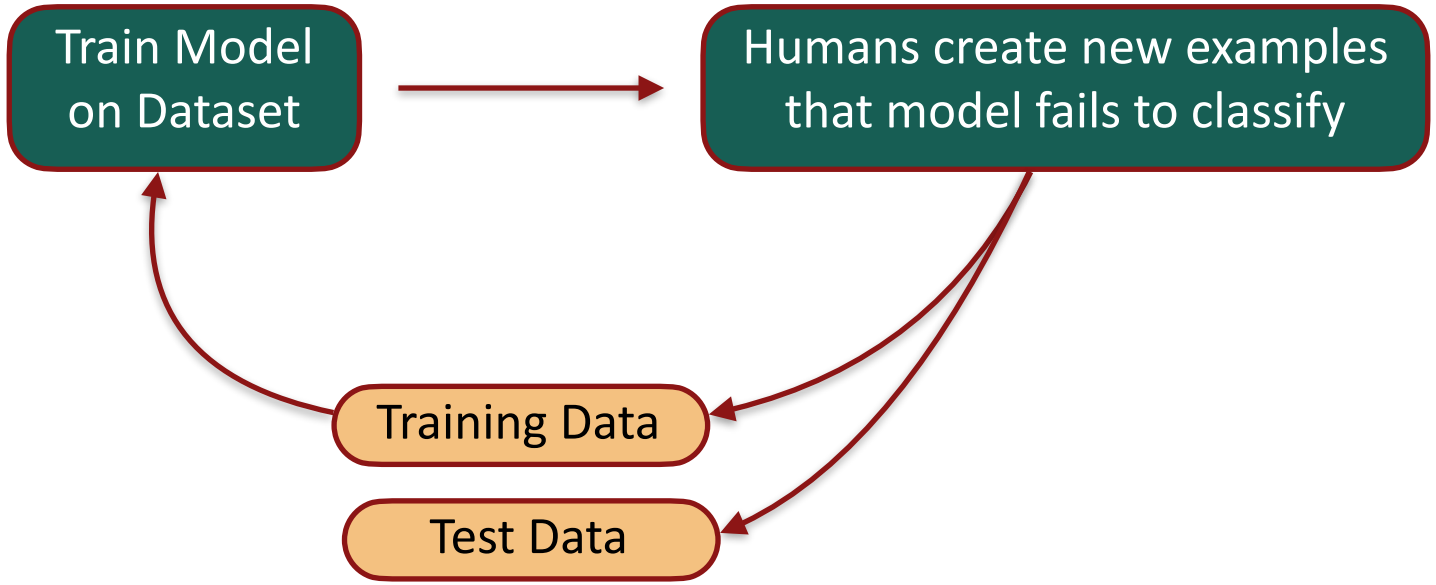
- Instead of testing models on static benchmarks, evaluate on an ever changing *dynamic* benchmark.
- Recent Examples:
  - Adversarial NLI by Nie et al. 2020
  - DynaSent by Potts et al. 2020
  - other related examples: “Build It, Break It” Workshop at EMNLP 17

Improving how we evaluate models in NLP: Dynamic Benchmarks



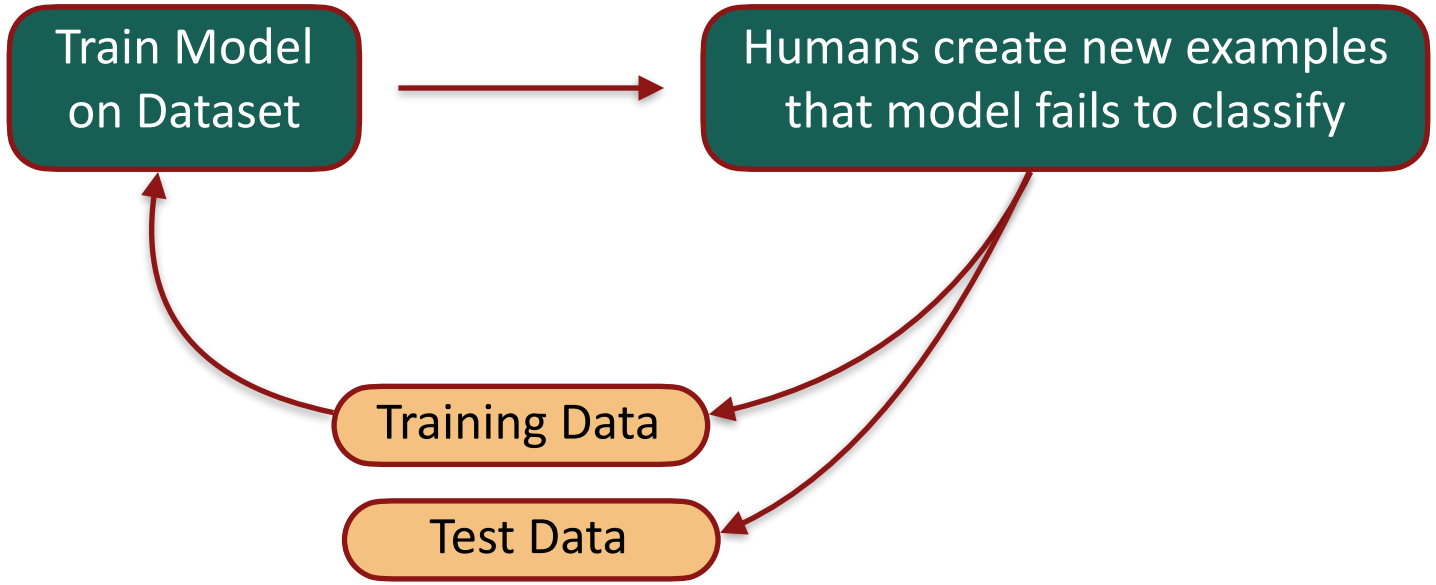
Overview of dynamic benchmarks

Improving how we evaluate models in NLP: Dynamic Benchmarks



1. Start with a pre-trained model and fine-tune it on the original train / test datasets
2. Humans attempt to create new examples that fool the model but not other humans
3. These examples are then added into the train / test sets and the model is retrained on the augmented dataset

Improving how we evaluate models in NLP: Dynamic Benchmarks

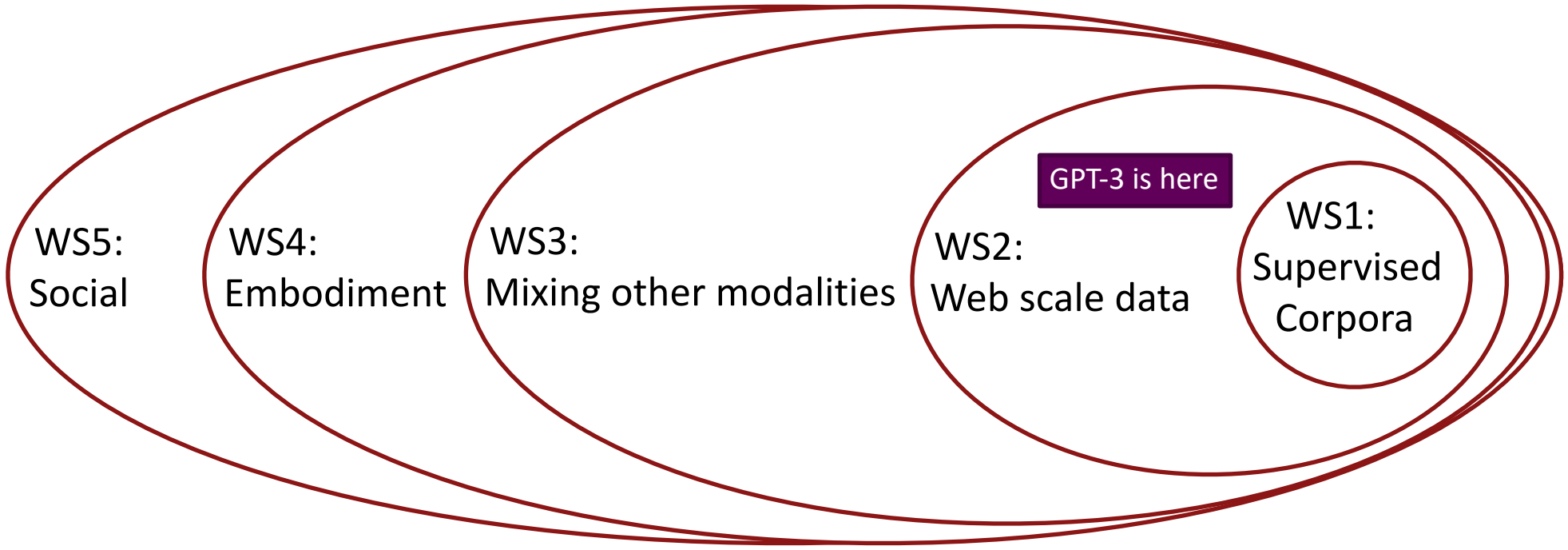


- Main Challenges: Ensuring that humans are able to come up with hard examples and we are not limited by creativity.
- Current approaches use examples from other datasets for the same task as prompts

Grounding Language to other modalities

Grounding Language to other modalities

- Many have articulated the need for using modalities other than text
- Bender and Koller [2020]: Impossible to acquire “meaning” (communicative intent of the speaker) from form (text / speech signal) alone
- Bisk et al [2020]: Training on only web-scale data limits the world scope of models.



Grounding Language to other modalities

1. Open questions in this space:
    1. Given that we might need to move beyond just text, what is the best way to do this at scale?
    2. Babies cannot learn language from watching TV alone [Snow et al 1976] but how far can models (especially when combined with scale)?
    3. If interactions with the environment is necessary, how do we collect data and design systems that interact minimally or in a cost effective way?
    4. Could pre-training on text still be useful by making any of the above more sample-efficient?

Interested in learning more about language grounding? Take CS224U (offered in the spring!)

Get Involved with NLP ∩ Deep Learning research!

Get Involved with NLP ∩ Deep Learning research:
General principles for making progress in neural NLP research

- Read broadly:
  1. Not just neural NLP papers from 2015 onwards but also pre-2010 NLP.
  2. Statistical machine learning (take CS229M) to understand generalization
  3. Learn more about language! (take CS224u, linguist 130a)
  4. Learn about Child language acquisition

Get Involved with NLP ∩ Deep Learning research:
General principles for making progress in neural NLP research

- Master your software tools:
  1. Scripting tools: awk, grep, bash
  2. version control: git
  3. data wrangling: pandas
  4. Jupyter notebooks for quick visualizations
  5. experiment management tools like Weights and Biases (https://wandb.ai/site)

- If your approach doesn't seem to be working: Don't panic!
- Put assert statements everywhere to check if computations are correct.
- **Use breakpoints `(import pdb; pdb.set_trace();)` extensively**
- Check if the loss computation is correct: For a k-way classification problem, initial loss should be ln(k)
- Start by creating a small training dataset (10 examples) and see if your model can overfit to this.
- check for saturating activations, dead ReLUs
- Check if gradient values are too small (perhaps use residual connections / LSTMs instead of RNNs/ better initialization) or too large (use gradient clipping)
- Overall, be methodical. If your approach doesn't work, come up with hypothesis for why this might be the case, and design oracle experiments to debug it!

Quick Final Project Tips: Primer on pdb

- Like gdb, but for python
- To create a breakpoint, simple add the line import pdb; pdb.set_trace() before the line you want to inspect.

Get Involved with NLP ∩ Deep Learning research!
Quick Final Project Tips: Primer on pdb

```
from transformers import AutoTokenizer, DistilBertForQuestionAnswering
import torch

# training corpus
context = "One morning I shot an elephant in my pajamas. How he got in my pajamas, I don't know."
question = "What did I shoot?"


def main():
    tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
    model = DistilBertForQuestionAnswering.from_pretrained("distilbert-base-uncased")
    tokenized_input = tokenizer(context, question)
    model.eval()
    output = model(**tokenized_input)


if __name__ == "__main__":
    main()
```

Get Involved with NLP ∩ Deep Learning research!
Quick Final Project Tips: Primer on pdb

```
from transformers import AutoTokenizer, DistilBertForQuestionAnswering
```

```
▶ python tinytransformer.py
Traceback (most recent call last):
  File "tinytransformer.py", line 25, in <module>
    main()
  File "tinytransformer.py", line 21, in main
    output = model(**tokenized_input)
  File "/Users/shikhar/miniconda3/lib/python3.6/site-packages/torch/nn/modules/module.py", line 727, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/Users/shikhar/miniconda3/lib/python3.6/site-packages/transformers/models/distilbert/modeling_distilbert.py
", line 709, in forward
    return_dict=return_dict,
  File "/Users/shikhar/miniconda3/lib/python3.6/site-packages/torch/nn/modules/module.py", line 727, in _call_impl
    result = self.forward(*input, **kwargs)
  File "/Users/shikhar/miniconda3/lib/python3.6/site-packages/transformers/models/distilbert/modeling_distilbert.py
", line 465, in forward
    input_shape = input_ids.size()
AttributeError: 'list' object has no attribute 'size'
```

Get Involved with NLP ∩ Deep Learning research!
Quick Final Project Tips: Primer on pdb

```
from transformers import AutoTokenizer, DistilBertForQuestionAnswering
import torch

# training corpus
context = "One morning I shot an elephant in my pajamas. How he got in my pajamas, I don't know."
question = "What did I shoot?"


def main():
    tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
    model = DistilBertForQuestionAnswering.from_pretrained("distilbert-base-uncased")
    tokenized_input = tokenizer(context, question)
    model.eval()
    import pdb; pdb.set_trace();
    output = model(**tokenized_input)


if __name__ == "__main__":
    main()
```

Get Involved with NLP ∩ Deep Learning research!
Quick Final Project Tips: Primer on pdb

```
▶ python tinytransformer.py
> /Users/shikhar/Desktop/tinytransformer.py(21)main()
-> output = model(**tokenized_input)
(Pdb) tokenized_input.keys()
dict_keys(['input_ids', 'attention_mask'])
(Pdb) type(tokenized_input['input_ids'])
<class 'list'>
(Pdb) l
 16          tokenized_input = tokenizer(context, question)
 17          model.eval()
 18          import pdb
 19
 20          pdb.set_trace()
 21  ->      output = model(**tokenized_input)
 22
 23
 24      if __name__ == "__main__":
 25          main()
[EOF]
(Pdb)
```

Get Involved with NLP ∩ Deep Learning research!
Quick Final Project Tips: Primer on pdb

```
from transformers import AutoTokenizer, DistilBertForQuestionAnswering
import torch

# training corpus
context = "One morning I shot an elephant in my pajamas. How he got in my pajamas, I don't know."
question = "What did I shoot?"


def main():
    tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
    model = DistilBertForQuestionAnswering.from_pretrained("distilbert-base-uncased")
    tokenized_input = tokenizer(context, question)
    model.eval()
    tokenized_input_tensorized = {
        key: torch.tensor(val).unsqueeze(-1) for key, val in tokenized_input.items()
    }
    output = model(**tokenized_input_tensorized)


if __name__ == "__main__":
    main()
```

Get Involved with NLP ∩ Deep Learning research!

- CLIPS program:
  - For Stanford undergrads / MS / PhDs interested in doing research with the NLP group, we highly encourage you to apply to CLIPS:

Concluding Thoughts

Serious progress in the last decade thanks to data + hardware + neural networks.

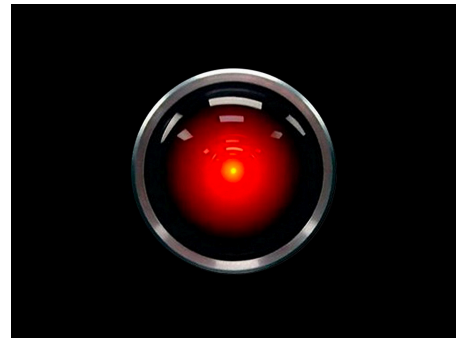
We now have amazing technologies such as GPT-3 that can do truly exciting things.

In the short term:

- Scaling helps, so perhaps even larger models?
- Scaling requires very non-trivial engineering efforts so a lot of interesting systems work to be done here!

In the long term:

- Making progress towards systematicity, fast adaptation, generalization
- Improved evaluation so we can trust benchmarks
- Figuring out how to move beyond text in a tractable way



Concluding Thoughts

Good Luck with Final Projects!!