Sub-Word Models, Data and Model Analysis, Data collection

Deep Natural Language Processing, 2022

Paweł Budzianowski

Group project

Link to the group teams - https://docs.google.com/spreadsheets/d/1TpwgPU9Z5WjBXln2QBX31g5dcDQozpD32ul_SHY0uZM/edit?usp=sharing

Let’s finalize the group formation **today**!

Plan

1. Character modelling
2. BPE and SentencePiece
3. Byte-level modelling
4. Model analysis as model evaluation
5. Model probing
6. Data-centric AI
7. Creating dialogue dataset

Sub-words model

Morphology

Morphology - morphemes as smallest **semantic** unit:

skak-ank-a

Morphology

Morphology - morphemes as smallest **semantic** unit:

skak-ank-a

Early work on building morpheme-based vectors dates back to 2013 [Luong et al., 2013]

Morphology

Character *n*-grams as alternative:

- Wickelphones [Rumelhart & McClelland, 1986]
- brew - #br, bre, rew, ew#

Variations in morphology

- No word segmentation
- Words mainly segmented
  - But…
    - Kraftfahrzeug-Haftpflichtversicherung
    - life insurance company employee

Multilingual approach

One model to rule them all?

We need to handle large, open vocabulary

- rich morphology
- databases
- transliteration
- informal spelling

Character-level models

Two ways to model:

1) Word embeddings can be composed from character embeddings - composing words out of characters

2) Process everything as a sequence!

Very Deep Convolutional Networks for Text Classification [Conneau et al. 2017]

Model increases with the depth: using up to 29 convolutional layers

Simple pooling and convolutional layers

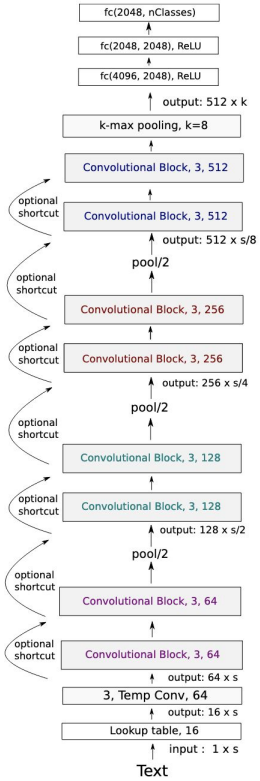


Figure 1: VDCNN architecture.

Purely character NMT models

Initial models were unsatisfactory [Vilar et al., 2007, Neubig et al., 2013]

Decoder only were helpful [Chung et al., 2016]

More promising results [Ling et al., 2015, Luong and Manning, 2016]

Sub-word models

Two family of models originated:

Sub-word models

Two family of models originated:

1) Same architecture as for word-level model - **BPE** [Sennrich et al., 2016]
2) Hybrid architectures - main model has words, UNKs for characters [Costa-Jussa and Fonollosa, 2016]

Byte Pair Encoding [BPE]

Originating as a compression algorithm:

most frequent **byte** pair -> a new **byte**

Replace bytes with **character *n*-grams**

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 l o w

2 l o w e r

6 n e w e s t

3 w i d e s t

Vocabulary:
l o w e r n w s t i d

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 l o w

2 l o w e r

6 n e w e s t

3 w i d e s t

Vocabulary:
l o w e r n w s t i d

Add a pair (e,s)

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 l o w

2 l o w e r

6 n e w **es** t

3 w i d **es** t

Vocabulary:
l o w e r n w s t i d **es**

Add a pair (e,s)

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 l o w

2 l o w e r

6 n e w es t

3 w i d es t

Vocabulary:
l o w e r n w s t i d es **est**

Add a pair (es,t)

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 l o w

2 l o w e r

6 n e w **est**

3 w i d **est**

Vocabulary:
l o w e r n w s t i d es **est**

Add a pair (es,t)

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 l o w

2 l o w e r

6 n e w est

3 w i d est

Vocabulary:
l o w e r n w s t i d es est **lo**

Add a pair (l,o)

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 **lo** w

2 **lo** w e r

6 n e w est

3 w i d est

Vocabulary:
l o w e r n w s t i d es est lo

Add a pair (l,o)

BPE algorithm

Goes with a bottom up clustering

Start with a unigram vocabulary of all characters in data

Most frequent *n*-gram pairs -> a new *n*-gram

Dictionary

5 lo w

2 lo w e r

6 n e w est

3 w i d est

Vocabulary:
l o w e r n w s t i d es est lo

Algorithm 1 Learn BPE operations

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

BPE algorithm

- Given a budget, stop when you reach it
- Do **deterministic** longest piece segmentation of words
- The segmentation happens only within words identified by some prior tokenizer

Automatically decides vocab for system!

- But we move away from words we operate with

WordPiece/SentencePiece model [Kudo and Richardson, 2018]

Rather than char *n*-gram count, use a greedy approximation to maximizing language model log likelihood to choose the pieces

Add *n*-gram that maximally reduces perplexity of the language model

WordPiece/SentencePiece model [Kudo and Richardson, 2018]

Wordpiece model tokenizes inside words - assumes we have a tokenizer

Sentencepiece model works from **raw text:**

- Whitespace is retained as a special token (_)
- You can reverse things at end by joining pieces and recoding them to spaces

WordPiece/SentencePiece model [Kudo and Richardson, 2018]

Wordpiece model tokenizes inside words - assumes we have a tokenizer.

Sentencepiece model works from **raw text:**

- Whitespace is retained as a special token (_)
- You can reverse things at end by joining pieces and recoding them to spaces

BERT uses a variant of the **WordPiece** model:

- common words are there: at, yes
- longers are builts from pieces: hypatia = h ##yp ##ati ##a

New studies come into play

Token-free encoding

Pros:

- They can process text in any language out of the box
- More robust to noise
- Minimize technical debt by removing complex and error-prone text pre-processing pipelines

Token-free encoding

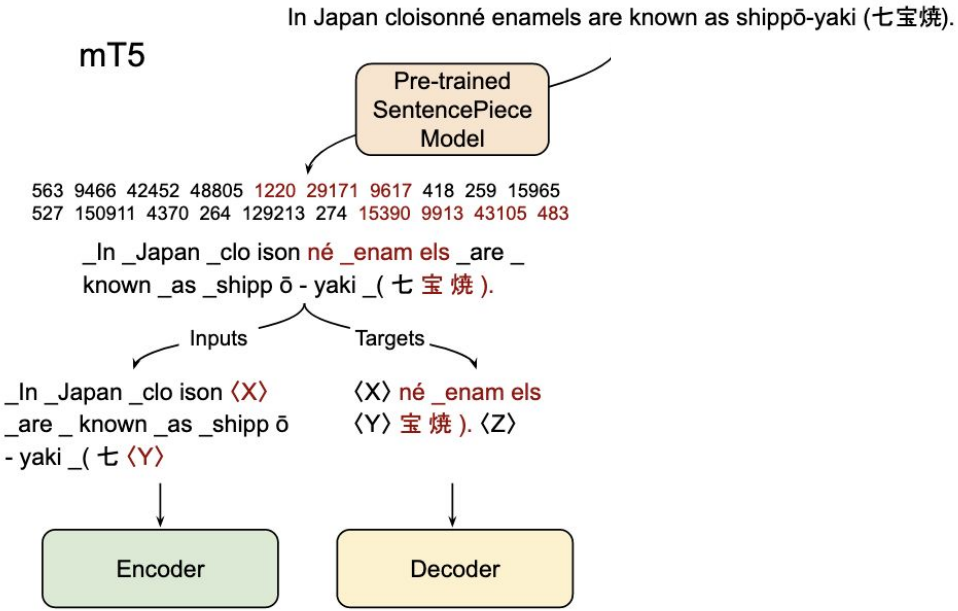
Pros:

- They can process text in any language out of the box
- More robust to noise
- Minimize technical debt by removing complex and error-prone text pre-processing pipelines

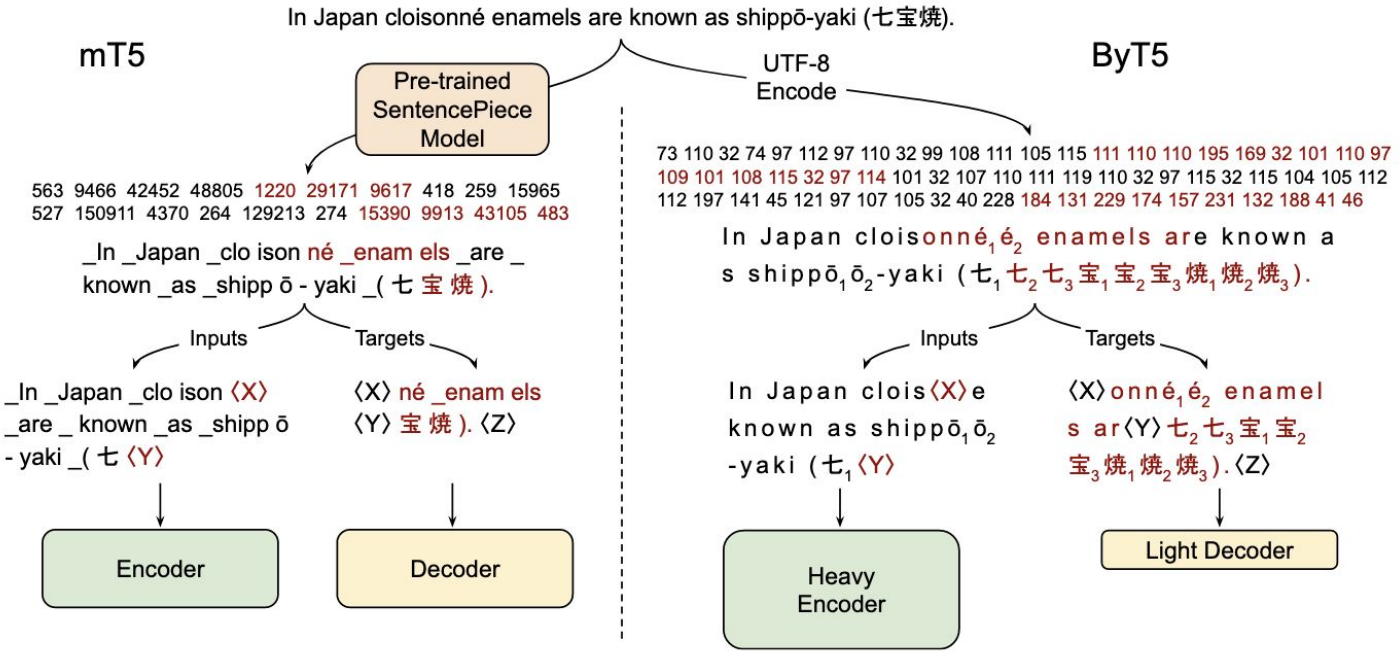
Cons:

- Byte or character sequences are longer than token sequences

ByT5 [Xue et al., 2021]



ByT5 [Xue et al., 2021]



mT5 vs ByT5

	GLUE		SuperGLUE	
Model	mT5	ByT5	mT5	ByT5
Small	75.6	**80.5**	60.2	**67.8**
Base	83.0	**85.3**	72.5	**74.0**
Large	**87.6**	87.0	**81.9**	80.4
XL	**88.7**	87.9	**84.7**	83.2
XXL	**90.7**	90.1	**89.2**	88.6

Ablation studies

Model	XNLI (Accuracy)	TyDiQA-GoldP (F1)	GEM-XSum (BLEU)
ByT5-Large (1.23B)	79.7	87.7	11.5
mT5-Large (1.23B)	81.1	85.3	10.1
(a) ByT5-36/12-668M	78.3	87.8	12.3
(b) ByT5-24/24-718M	75.4	83.0	7.1
(c) ByT5-12/36-768M	73.5	83.1	8.3
(d) mT5-36/12-1.18B	81.5	87.1	10.8
(e) ByT5-Large-Span3	79.4	87.4	10.2
(f) ByT5-Large-Span40	78.9	88.3	12.6
(g) CharT5-36/12-1.23B	79.0	87.6	11.2

Table 8: Ablation model results across three tasks.

Inference speed

	Grapheme-to-Phoneme		Dakshina	
	mT5	ByT5	mT5	ByT5
Small	1223	1190 (1.0×)	9483	6482 (1.5×)
Base	726	932 (0.8×)	7270	4272 (1.7×)
Large	387	478 (0.8×)	4243	2282 (1.9×)
XL	280	310 (0.9×)	2922	1263 (2.3×)
XXL	150	146 (1.0×)	1482	581 (2.6×)

	XNLI		GEM-XSum	
	mT5	ByT5	mT5	ByT5
Small	8632	1339 (6.4×)	750	202 (3.7×)
Base	5157	687 (7.5×)	450	114 (3.9×)
Large	1598	168 (9.5×)	315	51 (6.2×)
XL	730	81 (9.0×)	162	25 (6.4×)
XXL	261	33 (8.0×)	61	10 (6.3×)

Model Analysis

Model analysis and probing

What are our models doing?

We tend to summarize the model by the accuracy metric or some proxy to it

It works fine but doesn’t tell us anything

Different level of abstractions

- Neural model as a interlocutor
- Neural model as a probability distribution and decision function
- Neural model as a sequence of vector representation
- Parameter weights, attention mechanisms

Model evaluation as model analysis

When looking at the behavior of a model, we are not yet concerned with mechanisms the mode is using. We want to ask how does model behave in situations of interest.

Model evaluation as model analysis

When looking at the behavior of a model, we are not yet concerned with mechanisms the mode is using. We want to ask how does model behave in situations of interest.

You have trained your model on some samples

How does the model behave on samples from the same distribution:

a) Aka in domain or iid
b) This is your test set accuracy F1/BLEU

Model evaluation as model analysis in NLI

Imagine the task of the natural language inference (NLI):

A man inspects the uniform of a figure in some East Asian country.	**contradiction** C C C C C	The man is sleeping
An older and younger man smiling.	**neutral** N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	**contradiction** C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	**entailment** E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	**neutral** N N E C N	A happy woman in a fairy costume holds an umbrella.

Model evaluation as model analysis in NLI

What if our model is using simple heuristics to get good accuracy?

A **diagnostic test set** is carefully constructed to test for a specific skill or capacity of your neural model.

For example, HANS (Heuristic Analysis for NLI Systems) [McCoy et al., 2019] tests syntactic heuristics in NLI:

1) Lexical overlap
2) Subsequence
3) Constituent

Model evaluation as model analysis in NLI

What if our model is using simple heuristics to get good accuracy?

A **diagnostic test set** is carefully constructed to test for a specific skill or capacity of your neural model.

For example, HANS (Heuristic Analysis for NLI Systems) tests syntactic heuristics in NLI:

1) Lexical overlap

   The doctor was paid by the actor - The doctor paid the actor

Model evaluation as model analysis in NLI

What if our model is using simple heuristics to get good accuracy?

A **diagnostic test set** is carefully constructed to test for a specific skill or capacity of your neural model.

For example, HANS (Heuristic Analysis for NLI Systems) tests syntactic heuristics in NLI:

2) Subsequence

   The judges heard the actors resigned - The judges heard the actors

Model evaluation as model analysis in NLI

What if our model is using simple heuristics to get good accuracy?

A **diagnostic test set** is carefully constructed to test for a specific skill or capacity of your neural model.

For example, HANS (Heuristic Analysis for NLI Systems) tests syntactic heuristics in NLI:

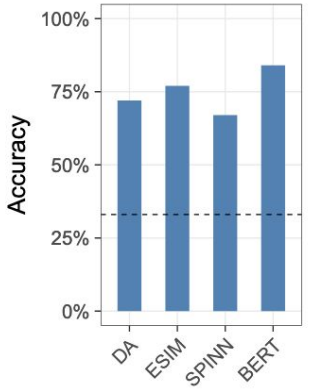
3) Constituent

   If the artist slept, the actor ran - The artist slept

HANS model analysis in natural language inference

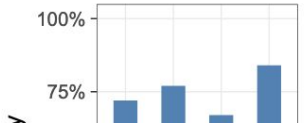
[McCoy et al., 2019] took 4 strong MNLI models with the following accuracies in **the original test data (in-domain)**

Evaluating on HANS, where syntactic heuristics work, accuracy is high

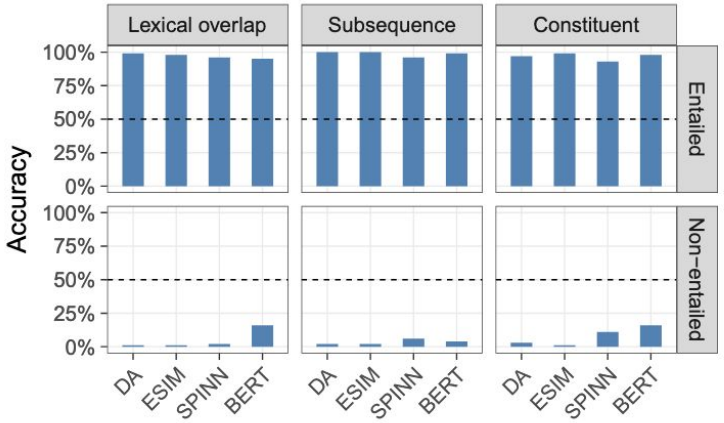


HANS model analysis in natural language inference

[McCoy et al., 2019] took 4 strong MNLI models with the following accuracies in **the original test data (in-domain)**



Evaluating on HANS, where syntactic heuristics work, accuracy i



But where syntactic heuristics fail, accuracy is very low

Language model as linguistic test subjects

How do we understand language behavior in humans?

One method: minimal pairs. What sounds okay to a speaker, but doesn’t with a small change?

The chef who made the pizzas is here

The chef who made the pizzas are here

Language model as linguistic test subjects

How do we understand language behavior in humans?

Language model as linguistic test subjects

How do we understand language behavior in humans?

One method: minimal pairs. What sounds okay to a speaker, but doesn’t with a small change?

The chef who made the pizzas is here

The chef who made the pizzas are here

Idea: English present-tense verbs agree in number with their subjects

Language models as linguistic test subjects

What's the language model analogue of acceptability?

Assign higher probability to the acceptable sentence in the minimal pair.

Just like in HANS, we can develop a test set with carefully chosen properties.

Specifically, can language models handle attractors in subject-verb agreement?

0 attractor: The chef is here

1 Attractor: The chef who made the pizzas is here.

2 Attractors: THe chef who made the pizzas and prepped the ingredients is here.

…

Language models as linguistic test subjects [Kuncoro et al., 2018]

An LSTM language model trained on a small set of Wikipedia text.

They evaluate it only on sentences with specific numbers of agreement attractors.

Language models as linguistic test subjects [Kuncoro et al., 2018]

An LSTM language model trained on a small set of Wikipedia text.

They evaluate it only on sentences with specific numbers of agreement attractors.

	n=0	**n=1**	**n=2**	**n=3**	**n=4**
Random	50.0	50.0	50.0	50.0	50.0
Majority	32.0	32.0	32.0	32.0	32.0
LSTM, H=50†	6.8	32.6	≈50	≈65	≈70
Our LSTM, H=50	2.4	8.0	15.7	26.1	34.65
Our LSTM, H=150	1.5	4.5	9.0	14.3	17.6
Our LSTM, H=250	1.4	3.3	5.9	**9.7**	13.9
Our LSTM, H=350	**1.3**	**3.0**	**5.7**	**9.7**	**13.8**
1B Word LSTM (repl)	2.8	8.0	14.0	21.8	20.0
Char LSTM	**1.2**	5.5	11.8	20.4	27.8

Language models as linguistic test subjects [Kuncoro et al., 2018]

An LSTM language model trained on a small set of Wikipedia text.

They evaluate it only on sentences with specific numbers of agreement attractors.

The **lead** is also rather long: 5 paragraphs **are** pretty lengthy.

Careful test sets as unit test suites:

Small careful test sets sounds like unit test suites, but for neural networks

Minimum functionality tests: small test sets that target a specific behavior

Careful test sets as unit test suites:

Small careful test sets sounds like unit test suites, but for neural networks

Minimum functionality tests: small test sets that target a specific behavior

Careful test sets as unit test suites:

Small careful test sets sounds like unit test suites, but for neural networks

Minimum functionality tests: small test sets that target a specific behavior

[Riberio et al., 2020] showed ML engineers working on a sentiment analysis product an interface with categories of linguist capabilites and types of tests.

The engineers found a bunch of bugs (categories of high error) through this method!

Fitting the dataset vs learning the task

1. Across a wide range of tasks, high model accuracy on the in-domain test set does not imply the model will also do well on other, reasonable out of domain examples.

Fitting the dataset vs learning the task

1. Across a wide range of tasks, high model accuracy on the in-domain test set does not imply the model will also do well on other, reasonable out of domain examples.

2. One way to think about this: models seem to be learning the **dataset** not the **task** like how humans can perform natural language inference.

Input influence: does my model really use long-distance context?

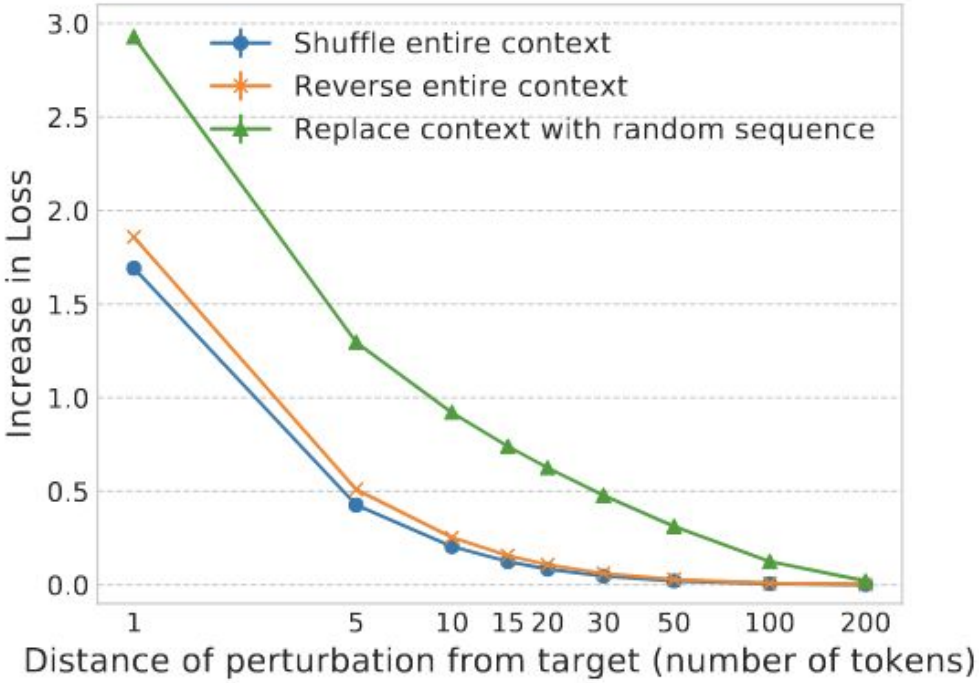
We motivated LSTM language models through their theoretical ability to use long-distance context to make predictions. But how long really is the LSTM?

Input influence: does my model really use long-distance context?

We motivated LSTM language models through their theoretical ability to use long-distance context to make predictions. But how long really is the LSTM?

[Khandelwal et al., 2019] idea: shuffle or remove all contexts farther than *k* words away for multiple values of *k* and see at which *k* the mode's predictions start to get worse!

Input influence: does my model really use long-distance context?



Prediction explanations: what in the input led to this output?

Prediction explanations: what in the input led to this output?

For a single example, what parts of the input led to the observed prediction?

Saliency maps: a score for each input word indicating its importance to the model's prediction

Method	Saliency Map
Conformity	an intelligent fiction about learning through cultural clash.
Confidence	an intelligent fiction about learning through cultural clash.
Gradient	an intelligent fiction about learning through cultural clash.
Conformity	<Schweiger> is talented and terribly charismatic.
Confidence	<Schweiger> is talented and terribly charismatic.
Gradient	<Schweiger> is talented and terribly charismatic.
Conformity	Diane Lane shines in unfaithful.
Confidence	Diane Lane shines in unfaithful.
Gradient	Diane Lane shines in unfaithful.

Color Legend Positive Impact Negative Impact

Prediction explanations: simple saliency maps

- How do we make a saliency map? Many ways to encode the intuition of importance

Prediction explanations: simple saliency maps

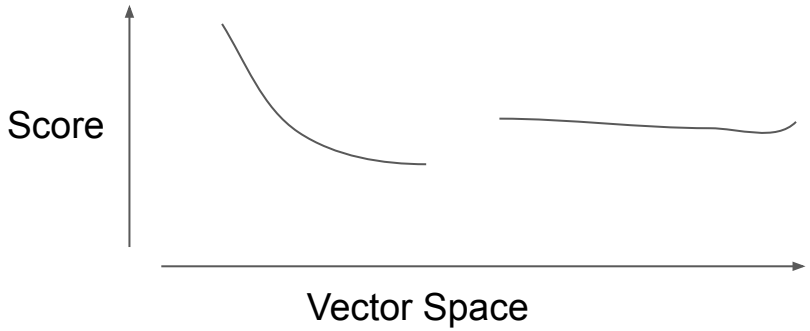
- How do we make a saliency map? Many ways to encode the intuition of importance
- Simple gradient method:
  - For words $x_1, \ldots, x_n$ and the model's score for a given class (output label) $l_c(x_1, \ldots, x_n)$ take the norm of the gradient of the score w.r.t. each word:

$$\text{salience}(x_i) = ||\nabla_{x_i} l_c(x_1, \ldots, x_n)||$$

- Idea: high gradient norm means changing that word (locally) would affect the score a lot

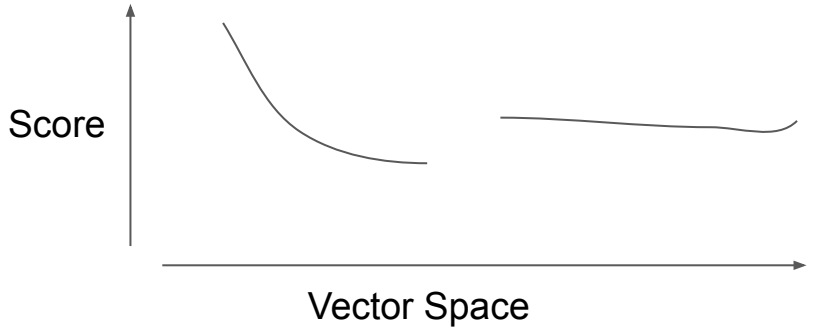
Prediction explanations: simple saliency maps

- Well-behaved model and space

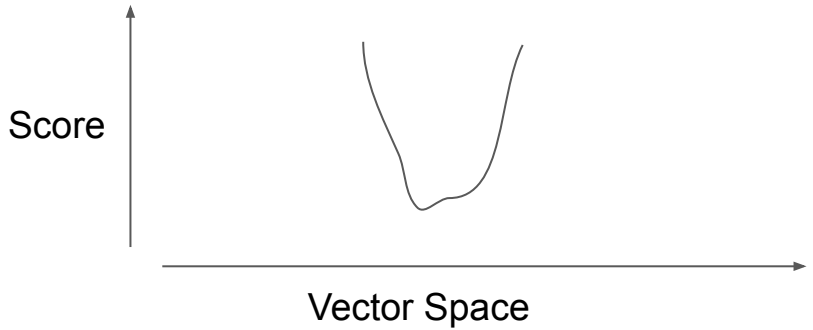


Prediction explanations: simple saliency maps

- Well-behaved model and space



- Reality is not as nice as wanted



A method for explanation by input reduction

Explanation by input reduction [Feng et al., 2018]

What is the smallest part of the input I could keep and still get the same answer?

An example from SQuAD:

SQuAD	
Context	In 1899, John Jacob Astor IV invested $100,000 for Tesla to further develop and produce a new lighting system. Instead, Tesla used the money to fund his Colorado Springs experiments.
Original	What did Tesla spend Astor's money on ?
Reduced	did
Confidence	$0.78 \rightarrow 0.91$

A method for explanation by input reduction

Idea: run an input saliency method. Iteratively remove the most unimportant words.

SQuAD

Context: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.

Question:
(0.90, 0.89) Where did the Broncos practice for the Super Bowl ?
(0.92, 0.88) Where did the practice for the Super Bowl ?
(0.91, 0.88) Where did practice for the Super Bowl ?
(0.92, 0.89) Where did practice the Super Bowl ?
(0.94, 0.90) Where did practice the Super ?
(0.93, 0.90) Where did practice Super ?
(0.40, 0.50) did practice Super ?

Analyzing models by breaking them

Analyzing models by breaking them

Idea: Can we break models by making seemingly innocuous changes to the input?

Yes: Look for adversarial examples

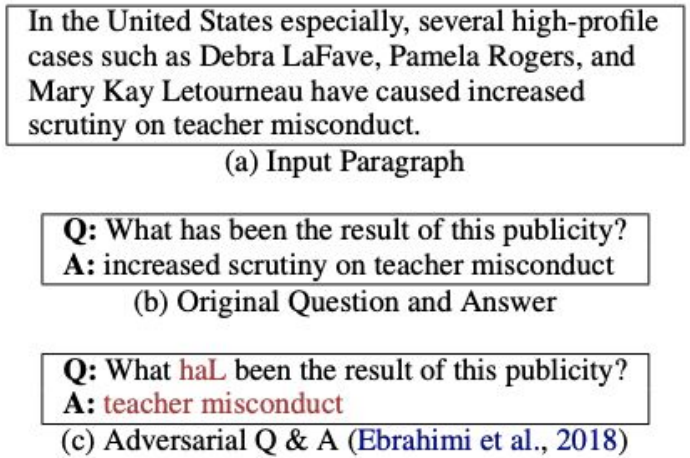
(just like in computer vision)

Analyzing models by breaking them

Idea: Can we break models by making seemingly innocuous changes to the input?

Yes: Look for adversarial examples

(just like in computer vision)

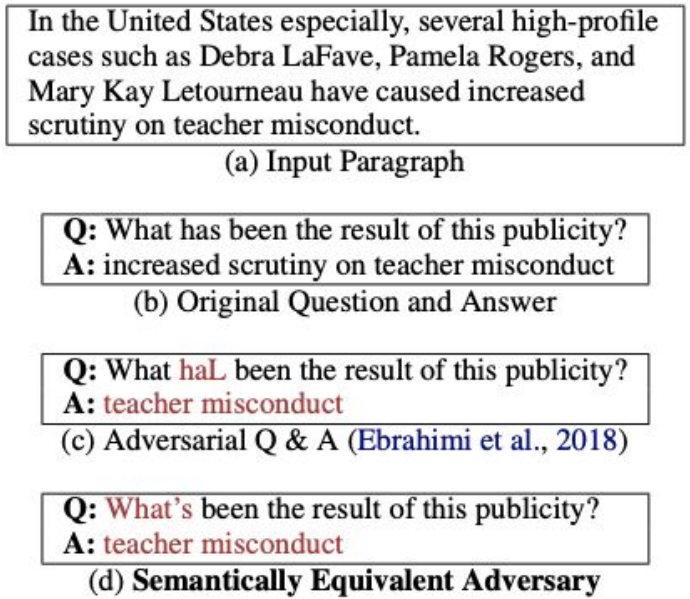


Analyzing models by breaking them

Idea: Can we break models by making seemingly innocuous changes to the input?

Yes: Look for adversarial examples

(just like in computer vision)



Are models robust to noise in their input.

Noise of various kinds is an inevitable part of the inputs to NLP systems. How do models trained on (relatively) clean text perform when typo-like noise is added?

Are models robust to noise in their input.

Noise of various kinds is an inevitable part of the inputs to NLP systems. How do models trained on (relatively) clean text perform when typo-like noise is added?

[Belinkov and Bisk, 2018] performed a study on popular MT models:

			Synthetic				
		Vanilla	Swap	Mid	Rand	Key	Nat
French	charCNN	42.54	10.52	9.71	1.71	8.26	17.42
German	charCNN	34.79	9.25	8.37	1.02	6.40	14.02
	char2char	29.97	5.68	5.46	0.28	2.96	12.68
	Nematus	34.22	3.39	5.16	0.29	0.61	10.68
Czech	charCNN	25.99	6.56	6.67	1.50	7.13	10.20
	char2char	25.71	3.90	4.24	0.25	2.88	11.42
	Nematus	29.65	2.94	4.09	0.66	1.41	11.88

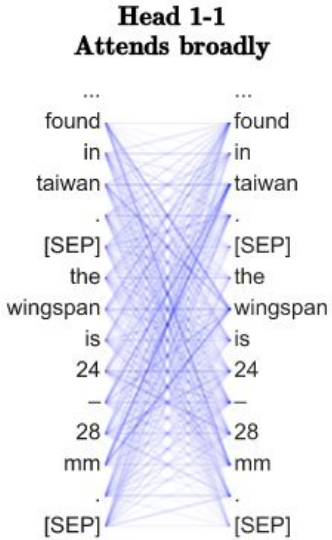
Analysis of “interpretable” architecture components [Clark et al., 2018]

Some modeling components are available to inspection.

Analysis of “interpretable” architecture components [Clark et al., 2018]

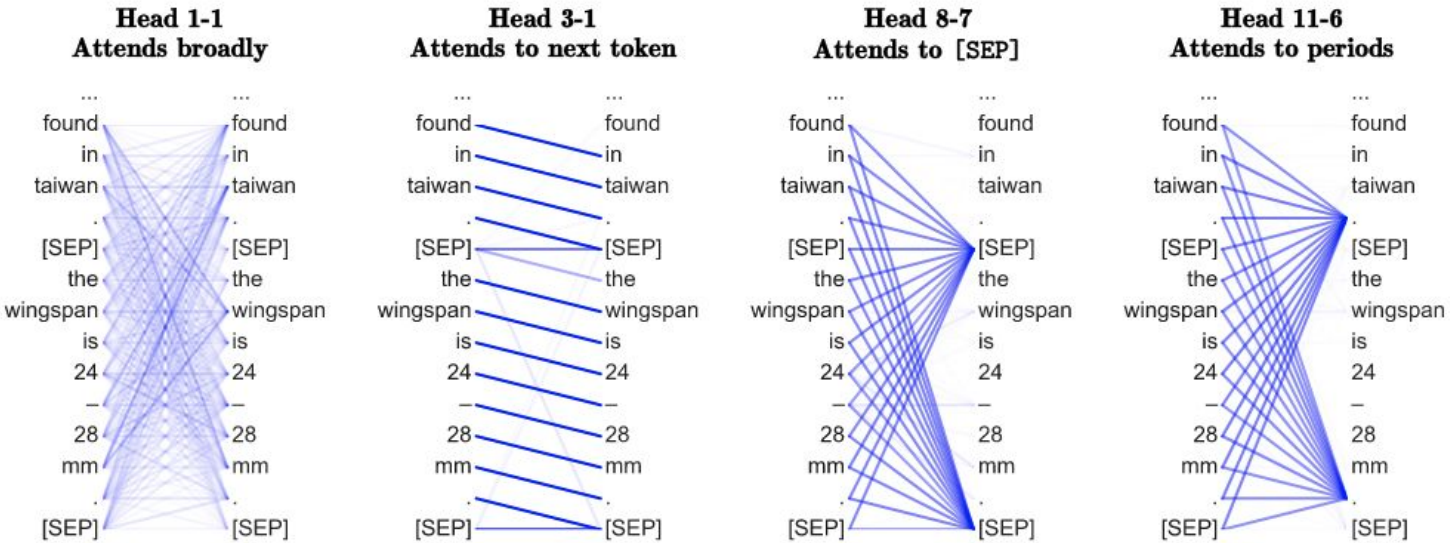
Some modeling components are available to inspection.

For example, can we try to characterize each attention head of Transformers?



Analysis of “interpretable” architecture components

Different heads to different parts of the input

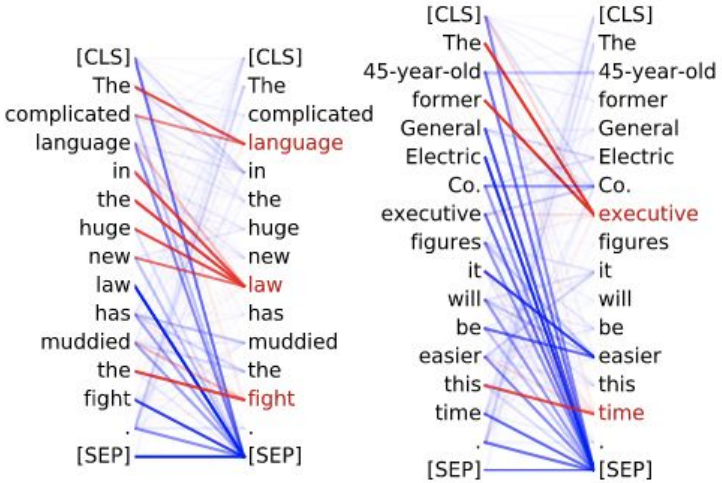


Analysis of “interpretable” architecture components

However, some heads shows actual language properties!

Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the **det** relation



Analysis of “interpretable” architecture components

Subject-verb agreement. What’s the mechanism by which LSTMs solve the task?

A word-level LSTM language model.

Analysis of “interpretable” architecture components

Subject-verb agreement. What’s the mechanism by which LSTMs solve the task?

A word-level LSTM language model.

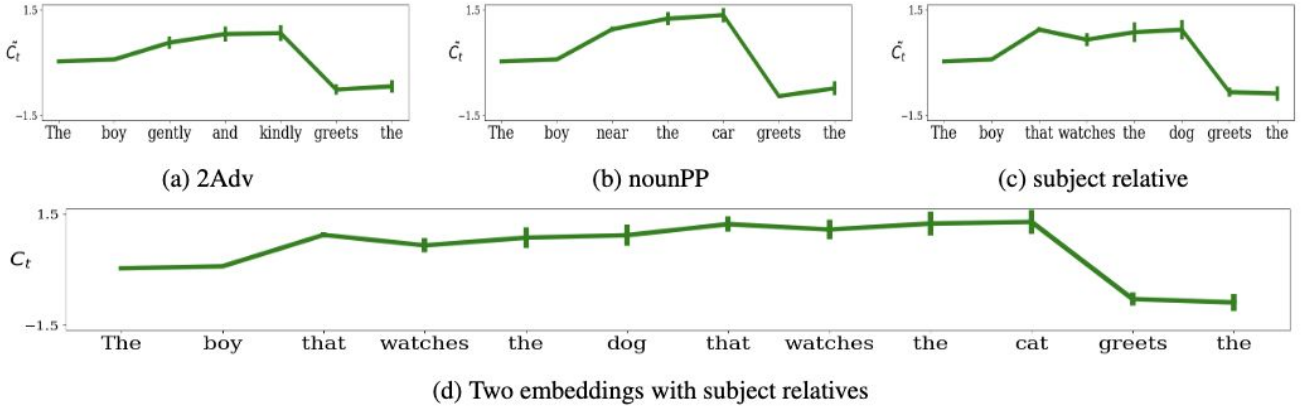


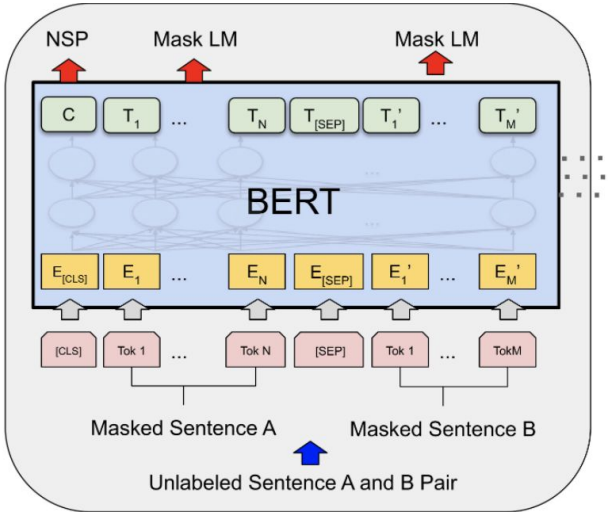
Figure 3: Cell activity of syntax unit 1150 while processing various syntactic structures. Values averaged across all stimuli in an NA-task, with error bars representing standard deviations. Relative clause NA-task stimuli were specifically generated for this visualization.

Probing: supervised analysis of neural networks

Probing: supervised analysis of neural networks

Assumption: general purpose language representations

Question: what do their representations encode about language?



Probing: supervised analysis of neural networks

Probing recipe:

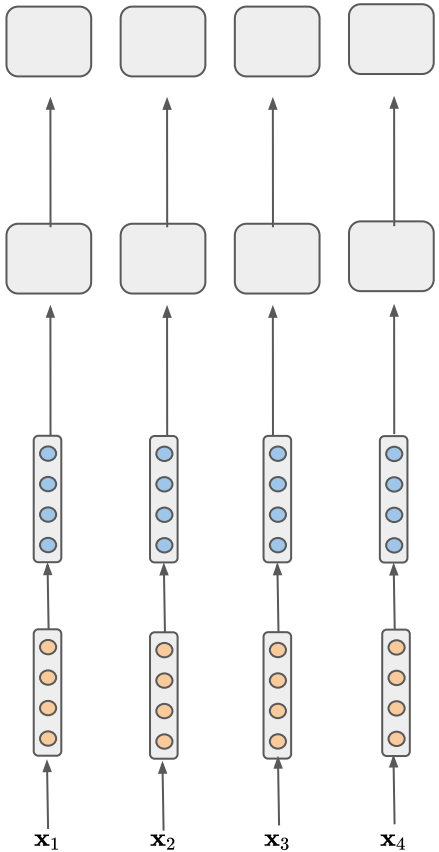
1. Freeze the whole model
2. Decide on a probe family (mostly linear)
3. What linguistic property we look at
4. Use supervision to train and evaluate

Performance: 0.94

Syntax task

Probe family (linear)

2 frozen layer of your LM

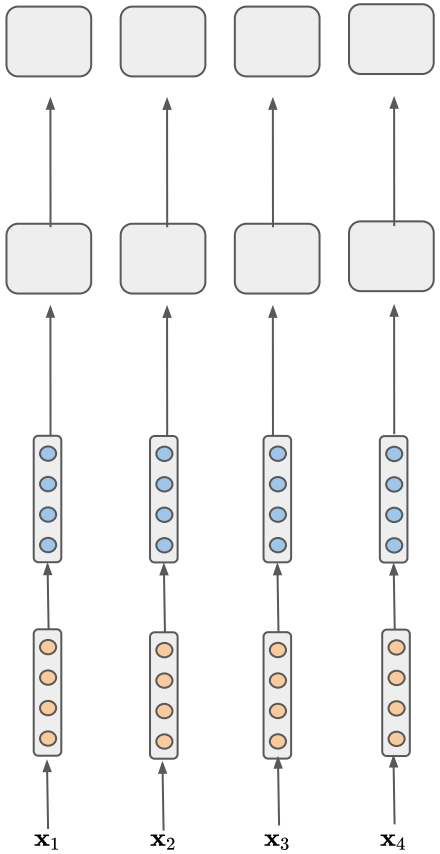


Performance: 0.94

Syntax task

Probe family (linear)

2 frozen layer of your LM

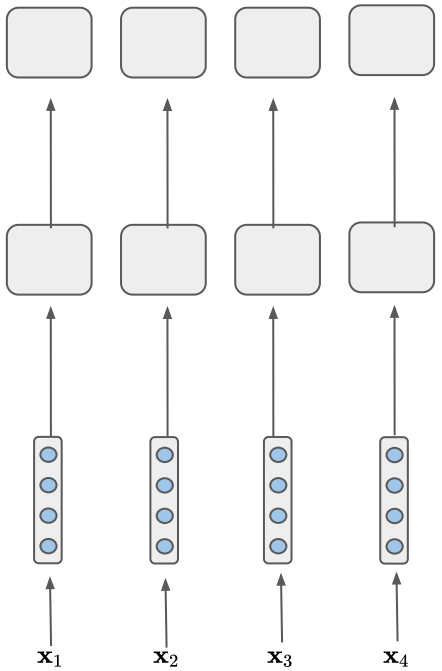


Performance: 0.98

Syntax task

Probe family (linear)

1 frozen layer of your LM

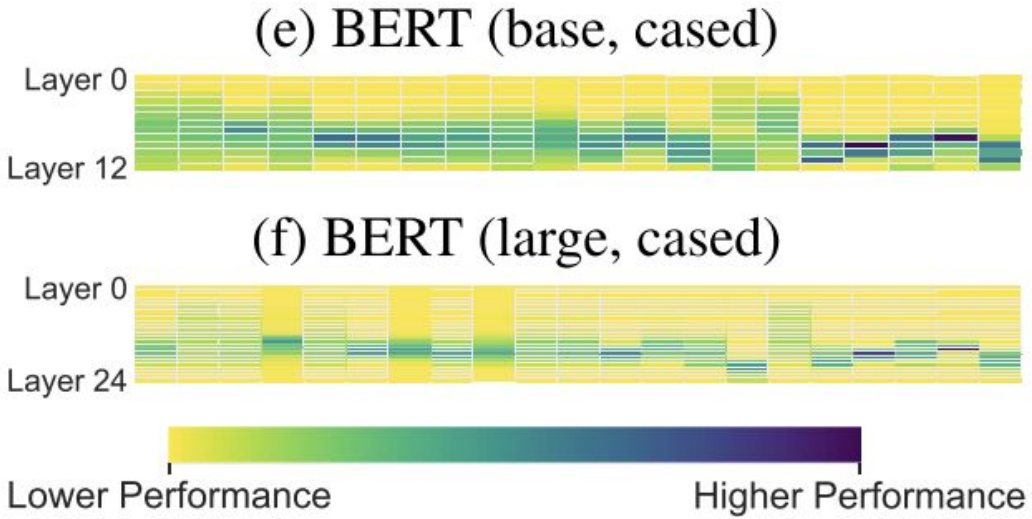


Probing: supervised analysis of neural networks

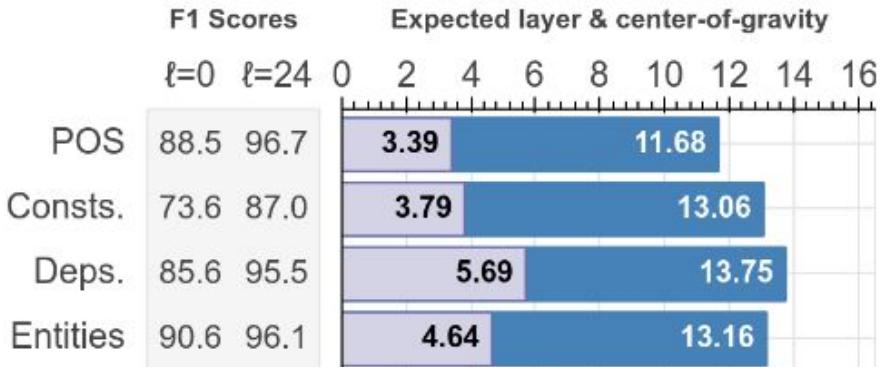
BERT (and other pretrained LMs) make some linguistic properties predictable to very high accuracy with a simple linear probe.

Layerwise trends of probing accuracy

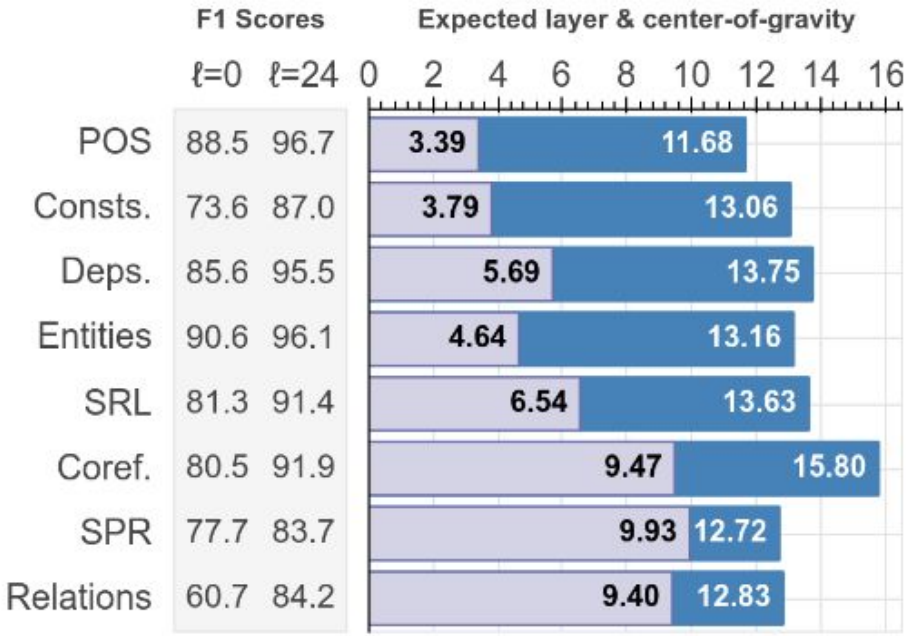
BERT (and other pretrained LMs) make some linguistic properties predictable to very high accuracy with a simple linear probe.



Layerwise trends of probing accuracy



Layerwise trends of probing accuracy



TODO add? Probing: trees simply recoverable from BERT representations

Dependency parse trees describe underlying syntactic structure in sentences.

[Hewitt and Manning, 2019] show that BERT models make dependency parse **tree structure** easily accessible.

Tree path distance: the number of edges in the path between the words

Squared euclidean distance of BERT vectors after transformation by the probe matrix B.

Probing and correlation studies

- Probing shows that properties are accessible to your probe family, **not** that they are used by the neural model you are studying
- Correlation studies (like attention maps) likewise

Probing and correlation studies

- Probing shows that properties are accessible to your probe family, **not** that they are used by the neural model you are studying
- Correlation studies (like attention maps) likewise
- For example:
  - [Hewitt and Liang, 2019] show that under certain conditions probes can achieve high accuracy on random labels
  - [Ravichander et al., 2021] show that probes can achieve high accuracy on a property even when the model is trained to know the property isn’t useful

Recasting model tweaks and ablations as analysis

Consider the usual neural network improvement process

[Michel et al., 2019] train transformers on MT and NLI

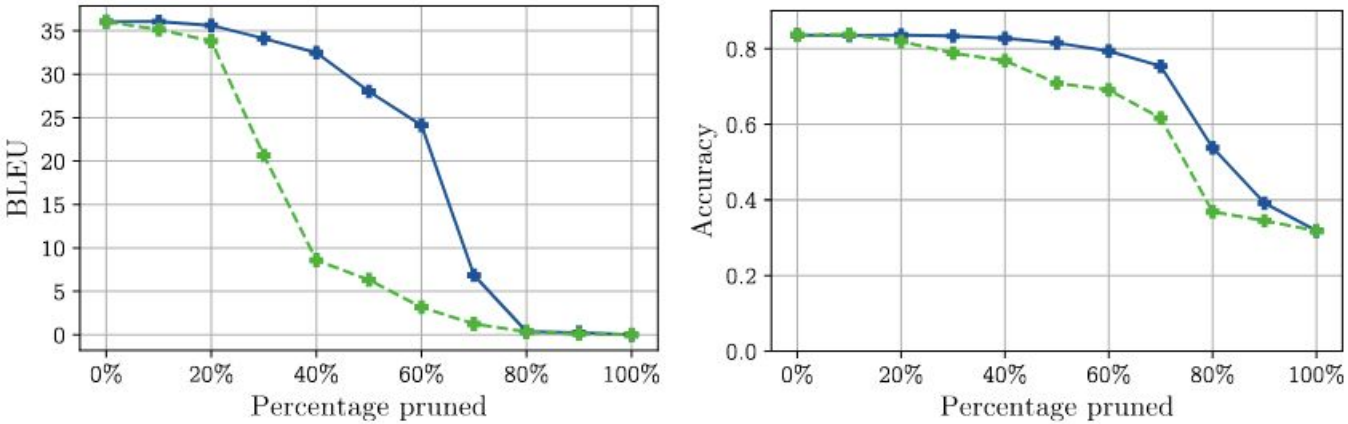
After training, they find many attention heads can be **removed** with **no drop** in accuracy

Recasting model tweaks and ablations as analysis

Consider the usual neural network improvement process.

[Michel et al., 2019] train transformers on MT and NLI.

After training, they find many attention heads can be **removed** with **no drop** in accuracy.



What is the right layer order for a transformer?

[Press et al., 2019] analyzed if there is a better ordering of self attention and feed-forward layers?

(a) Interleaved Transformer

(b) Sandwich Transformer

What is the right layer order for a transformer?

[Press et al., 2019] analyzed if there is a better ordering of self attention and feed-forward layers?

(a) Interleaved Transformer

(b) Sandwich Transformer

Model	**Test**
Baseline (Baevski and Auli, 2019)	18.70
Transformer XL (Dai et al., 2019)	18.30
kNN-LM (Khandelwal et al., 2019)	15.79
Baseline (5 Runs)	18.63 ± 0.26
Sandwich_6^{16}	17.96

Recap

Neural models are complex, and difficult to characterize. A single accuracy metric doesn’t cut it.

We struggle to find intuitive descriptions of model behaviors, but we have a many tools at many levels of abstraction to give insight.

Engage critically when someone claims a neural NLP model is interpretable - in what ways.

Bring this analysis and explanation way of thinking with you to your model building efforts even if analysis isn’t your main goal.

Sentence probing

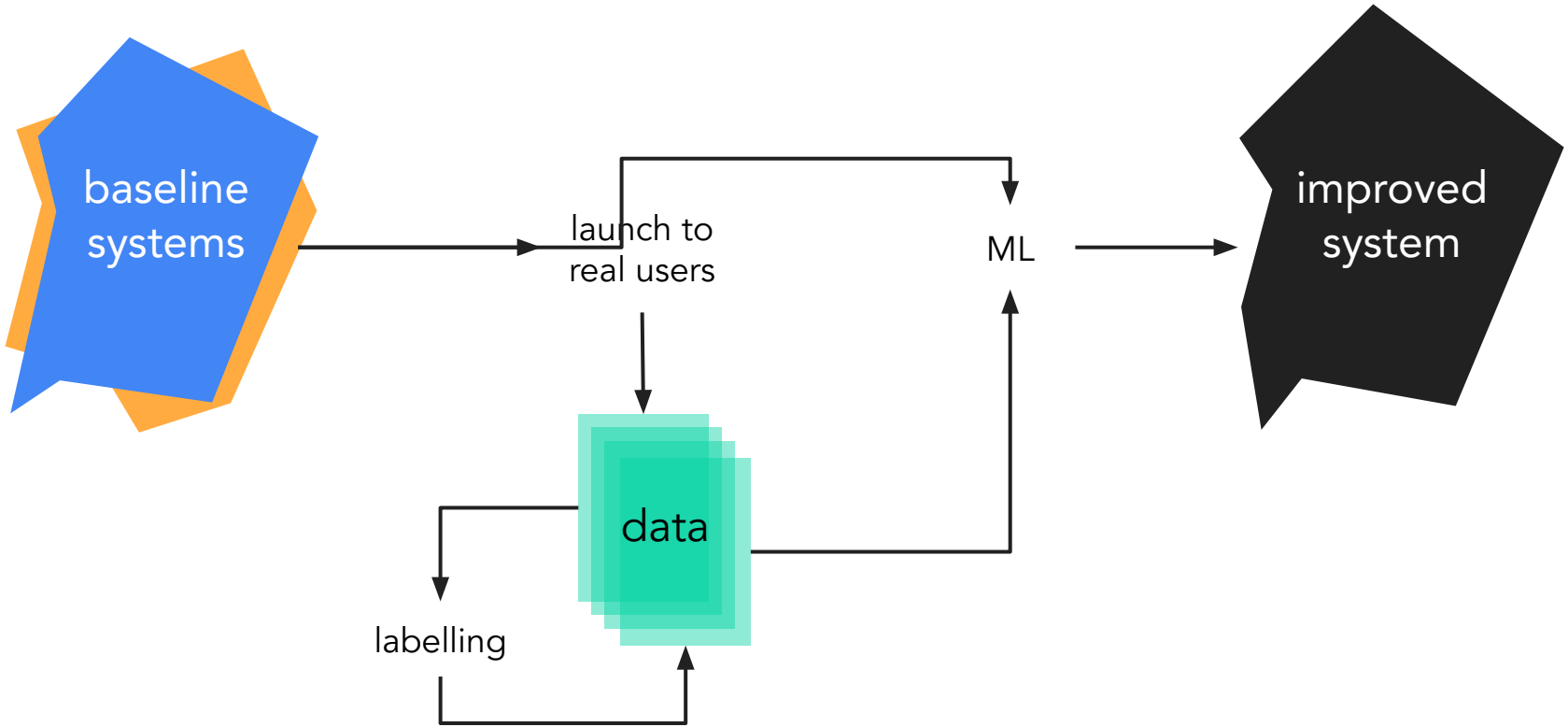
https://arxiv.org/pdf/1608.04207.pdf
sentence probing

edge probing

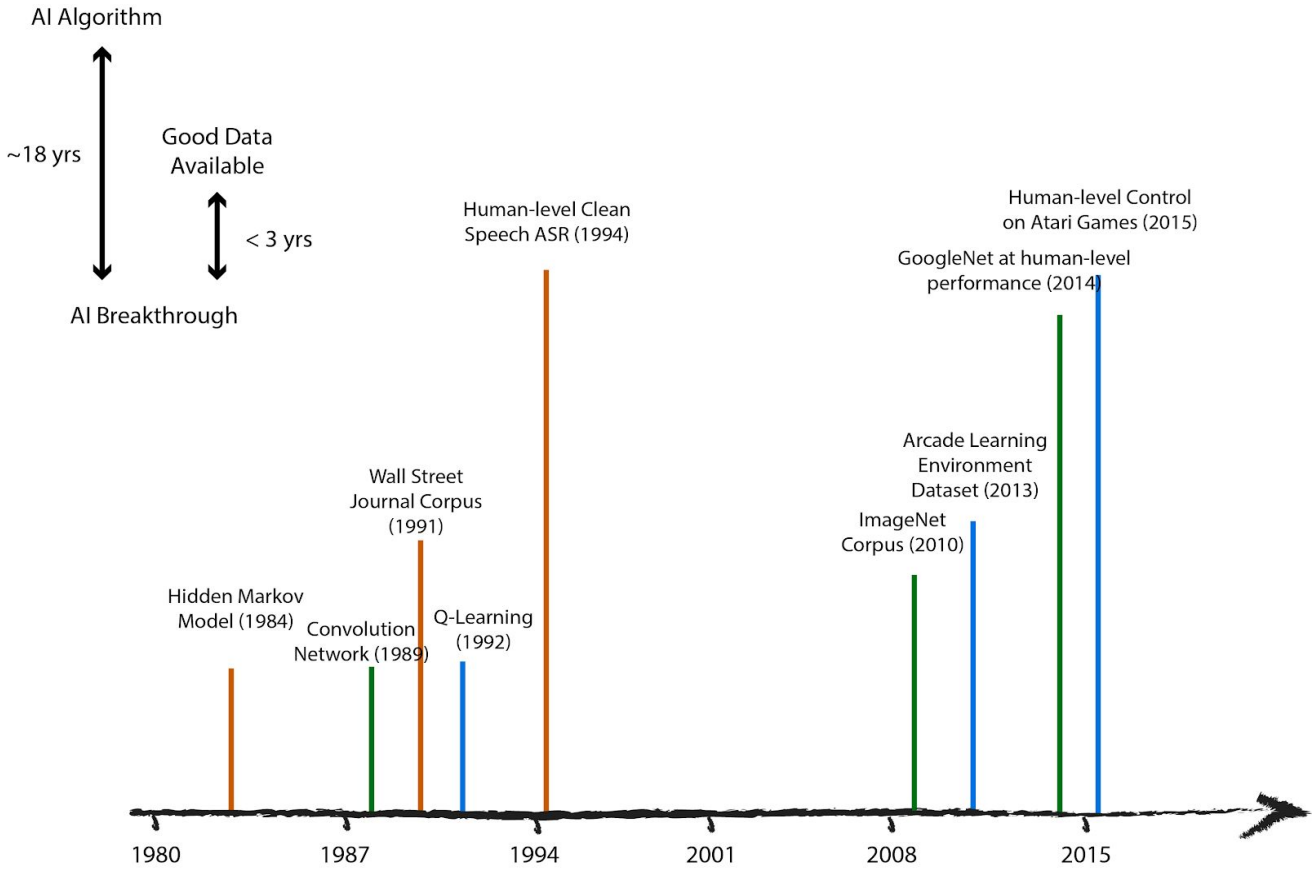
https://azpoliak.github.io/publications/edge-probe--iclr.pdf

Data collection

Data as necessary evil



Data First, Models (and Serenity) Later



Source:
http://www.spacemachine.net/views/2016/3/datasets-over-algorithms

Data success stories

- SQuAD
- Common Crawl
- GLUE
- Reddit

Stanford Question Answering Dataset (SQuAD)

- 100k annotated (passage, question, answer) triples
- Large-scale supervised datasets are also a key ingredient for training effective neural models for reading comprehension
- Passages are selected from English Wikipedia, typically 100-150 words
- Questions are crowd-sourced
- Each answer is a short segment of text (or span) in the passage.
    - **This is a limitation** - not all answers are given in that way
- It still remains the most popular reading comprehension dataset: it is almost solved today and the SOTA exceeds the estimated human performance.

Common Crawl

- A trillion words dataset available publicly
- GPT, T5, mT5, ….
- Data quality issues
- GPT2 case:
  - Scraped all outbound links from Reddit, a social media platform, which received at least 3 karma.
  - Heuristic indicator for whether other users found the link interesting, educational, or just funny
- GPT3 case:
  - filtered a version of CC based on similarity to a range of high-quality reference corpora
  - deduplication at the document level, within and across datasets, to prevent redundancy
  - addition of known high-quality reference corpora to the training mix to augment CC and increase its diversity

GLUE benchmark [Wang et al., 2018]

Corpus	|Train|	|Test|	Task	Metrics	Domain
			Single-Sentence Tasks		
CoLA	8.5k	**1k**	acceptability	Matthews corr.	misc.
SST-2	67k	1.8k	sentiment	acc.	movie reviews
			Similarity and Paraphrase Tasks		
MRPC	3.7k	1.7k	paraphrase	acc./F1	news
STS-B	7k	1.4k	sentence similarity	Pearson/Spearman corr.	misc.
QQP	364k	**391k**	paraphrase	acc./F1	social QA questions
			Inference Tasks		
MNLI	393k	**20k**	NLI	matched acc./mismatched acc.	misc.
QNLI	105k	5.4k	QA/NLI	acc.	Wikipedia
RTE	2.5k	3k	NLI	acc.	news, Wikipedia
WNLI	634	**146**	coreference/NLI	acc.	fiction books

Real Conversational Data Collection

Real Conversational Data Collection

Where can we find real data?

- Wikipedia, books, parliamentary sessions…
- Movie subtitles

Conversations on the internet

Conversational data on the internet

Internet is full of conversations:

- Social media (Facebook, Twitter, Instagram...)
- Product reviews (Amazon, Yelp, Tripadvisor...)
- QA sites (Quora, StackExchange, Yahoo Answers...)
- General discussion forums (Reddit)

Which of this data is publicly available?

Ubuntu dialogue corpus

Lowe et al., 2015, 2017

Chats from Ubuntu support IRC

~1M dialogues

Improved for DSTC7 in 2019 (Kummerfeld et al., 2019)

Open subtitles

Lison and Tiedemann., 2016

Movie subtitles on 62 languages

3.35 billion sentences (in all languages)

Large, diverse, multilingual

Divided in sentence fragments, video or history dependent

Reddit

- Largest corpus by far (3.7 billion comments)
- Spanning many topics (subreddits)
- Very conversational
- Long interactions
- Continuously growing

- Lots of hyperlinks, memes, images, non grammatical language, internet slang...

Dataset source: Jason Baumgartner, pushshift.io, reddit post

Dataset size comparison

	~ Turns	Annotations
DSTC 2&3	10^4	response, ASR, SLU
MultiWOZ	10^5	response, NLU
DSTC7 Reddit	10^6	response, entities
DSTC7 Ubuntu	10^6	response
AmazonQA	10^6	product, response
OpenSubtitles	10^8	'response'
Reddit	**10^9**	**response**

Literature

1. Sennrich et al., 2015 - https://arxiv.org/abs/1508.07909
2. Kudo and Richardson, 2018 - https://aclanthology.org/D18-2012.pdf
3. Xue et al., 2021 - https://arxiv.org/pdf/2105.13626.pdf
4. McCoy et al., 2019 - https://arxiv.org/pdf/1902.01007.pdf
5. Hewitt and Manning, 2019 - https://nlp.stanford.edu/pubs/hewitt2019structural.pdf
6. Rogers et al., 2020 - https://aclanthology.org/2020.tacl-1.54.pdf
7. Budzianowski et al., 2018 - https://arxiv.org/abs/1810.00278
8. Henderson et al., 2019 - https://arxiv.org/abs/1904.06472