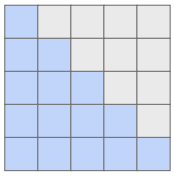
BERT and BERTology

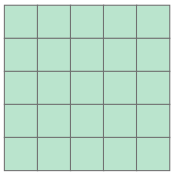
DS 4440 · LECTURE 14

1. Last time: the mask

For autoregressive generation token i must not attend to any $j > i$, so we set $a_{ij} = -\infty$ above the diagonal.



causal: i sees $j \leq i$



bidirectional: i sees all

Masking the *scores*, not the weights, so the softmax renormalizes over what is left:

$\text{sm}[5\ 3\ 4] = [0.665\ 0.090\ 0.245]$

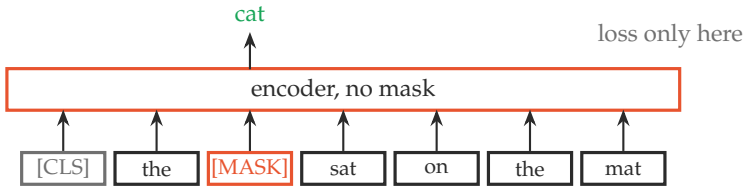
$\text{sm}[5\ 3\ -\infty] = [0.881\ 0.119\ 0]$

The mask is necessary for generation — without it "predict the next token" is trivial. But for *representation* it is a cost: why should a token see only its left context? ELMo's answer was two one-directional models glued together at the end.

> Drop the mask and every token sees every other at every layer. Then next-token prediction no longer works, and we need a different objective.

2. BERT [Devlin et al., 2019]

Same blocks as last lecture. Two changes: no causal mask, and a reconstruction objective.



Masking one sentence

the cat sat on the soft warm mat. Select 15% of positions — say 2 and 6 — and corrupt each independently:

pos	token	selected	model sees	target
2	cat	yes	`[MASK]`	cat
6	soft	yes	purple (random)	soft
others	—	no	unchanged	—

of the 15%	becomes
80%	`[MASK]`
10%	a random token
10%	unchanged

Reading out

From the final-layer state at each selected position:

$$\hat{y}_i = \underbrace{W_U}_{\langle V \times d_h \rangle} g(h_i^{(L)}) + b_U, \qquad \mathcal{L} = -\sum_{i \in \mathcal{M}} \log \text{softmax}(\hat{y}_i)_{x_i}$$

with $\mathcal{M}$ the selected positions and W_U usually tied to the input embeddings. Compute $h^{(L)}$ everywhere; score it on $\mathcal{M}$.

Why not 100% `[MASK]`**?**

A causal LM scores *every* position and nothing marks which — there is no shortcut available. MLM scores a few, and `[MASK]` says exactly where. So a model trained on 100% `[MASK]` can split its behavior:

	at `[MASK]`	**elsewhere**
100% `[MASK]`	gather context, guess	encode the token I was handed
80/10/10	gather context, guess	*also* gather context — might be scored, might be corrupt

At fine-tuning time there are no `[MASK]` tokens, so the first row's representations are useless. The two 10% cases each remove a cue: **unchanged** means a scored position can look ordinary; **random** means the token in front of me might be wrong, so the task becomes denoising rather than copying.

The rest

`[CLS]`	prepended to every input; its final state is a whole-sequence summary
`[SEP]`, segments	so a sentence pair is one sequence, with the parts marked
next sentence prediction	second objective; later found not to help [Liu et al., 2019]

task	**read off**	**example**
sequence classification	`[CLS]` state + linear head	sentiment, entailment
token classification	each state + shared head	NER, POS
span prediction	start and end heads	extractive QA

The transfer recipe from Lecture 10, with a transformer in place of a convnet.

3. BERTology

A large literature grew up asking what BERT had learned. Worth studying for its methods as much as its findings — it is where interpretability starts.



Before any result: **what claim does this method support?** Most disagreement here is about that, not about the measurements.

The first three support no causal claim. Ablation supports *necessity*, which is one-sided — it does not say a component is enough, or what it computes. Sufficiency and localization need intervention.

4. Behavioral probes

Write pairs that differ in one place, and check which one the model scores higher:

✓ I put the ice cream in the freezer.
× I put the ice cream in the oven.

✓ The capital of France is Paris.
× The capital of France is Berlin.

Black box throughout — we only ever look at outputs. That makes this the cleanest evidence we have and also the weakest: a model can be right for reasons we would not endorse. The classic syntactic version tests subject-verb agreement across an intervening noun [Linzen et al., 2016].

Compared to what?

The same design is used to measure bias and faithfulness: change one factor, measure how much the output moves. But an edit never changes only the factor we meant to change — it also perturbs the surface form, and models are sensitive to that on its own. So a nonzero effect is not yet evidence about the factor.
The fix is the same shape as the control task below: measure what a *meaning-preserving* edit does, and compare [Yang et al., 2026].

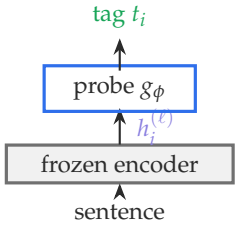
MedQA, prediction flip rate		
change patient gender	14.9%	the targeted intervention
paraphrase the input	14.1%	meaning preserved
difference	not statistically distinguishable	

Taken alone, 14.9% looks like strong evidence of sensitivity to gender. Next to the paraphrase baseline it is not evidence of anything in particular. Re-running a published analysis this way, only 5 of 120 reported effects survived — while a genuine directional effect in a separate task still came through, so the baseline is not simply washing everything out.

> A pattern worth noticing early: every method here needs a control. For probes it is a random relabelling; for counterfactual prompting it is a paraphrase. Without one, we are reading a number with nothing to compare it to.

5. Probing classifiers

Freeze the encoder, fit a small classifier to its states [Alain & Bengio, 2017]:



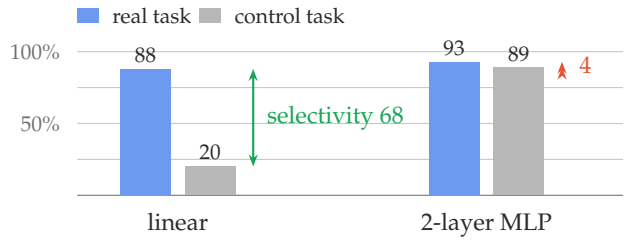
$\phi^\star = \arg\min_\phi \sum_i \ell\big(g_\phi(h_i^{(\ell)}), t_i\big)$,
θ fixed. Linear case: $g_\phi(h) = \mathrm{softmax}(hW+b)$,
so a high score says t is *linearly decodable* at ℓ.

A probe can learn the task by itself

Build a **control task** [Hewitt & Liang, 2019]: give every word in the vocabulary a random tag — the same tag every time that word appears — and refit the same probe on those labels instead. The

labels have nothing to do with grammar, so a probe can only predict them by memorizing which word it is looking at. Note this is perfectly learnable, and it transfers to held-out sentences, because those contain the same vocabulary. Whatever it scores there, A_{ctrl}, is what it can do without using any structure in the representation — so report the gap:

$$\text{selectivity} \;=\; A_{\text{real}} - A_{\text{ctrl}}.$$

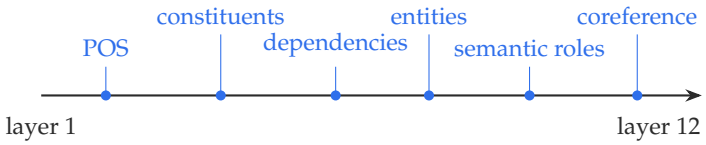


(Illustrative numbers; this is the shape of the finding.) The MLP wins on accuracy and says almost nothing — it can memorize word identity, so it does well whatever the labels mean. The weaker probe is the more informative one. Probing a *randomly initialized* encoder makes the same point: the result is not chance, because random features are still features.

Decodable is not used

A property can be linearly recoverable while the rest of the network ignores it. Probing measures what *we* can extract with a classifier trained for the purpose.

The pipeline result [Tenney et al., 2019]



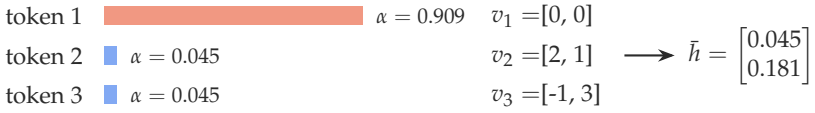
Roughly a classical NLP pipeline. A claim about where things become *decodable*, not about a sequence of computations the model performs.

6. Attention analysis

Attention weights are easy to plot, so people plotted them [Clark et al., 2019]: some heads attend to the next token, the previous token, the verb a noun depends on. Many attend overwhelmingly to `[SEP]`.

Why a big weight can mean nothing

Three tokens, $\alpha = \text{softmax}[3\ \ 0\ \ 0] = [0.909\ \ 0.045\ \ 0.045]$, with token 1 a delimiter whose value is near zero:



Ninety-one percent of the weight contributed nothing. And $\alpha' = [0\ \ 0.5\ \ 0.5]$ — a completely different distribution — gives $[0.5\ \ 2.0] = \bar{h}/0.091$: same direction, different scale.

Attention weight is not attribution. α_{ij} says how much of v_j went into $\bar{h}_i$; nothing about what v_j contained, or what the next layer did with it [Jain & Wallace, 2019]. The counter [Wiegreffe & Pinter, 2019]: it depends what we want from an explanation.

7. Breaking things on purpose

Ablation: which heads are necessary

Put a mask variable on every head of every layer. Writing the attention sublayer of layer ℓ as

$$\mathrm{MHAtt}^{(\ell)}(x) \;=\; \sum_{a=1}^{H} \xi_{\ell,a}\,\mathrm{Att}_{\ell,a}(x), \qquad \xi_{\ell,a} = 1 \text{ normally},$$

ablating head a of layer ℓ means setting $\xi_{\ell,a} = 0$. There are $L \times H$ of these — 144 for BERT-base — so trying subsets is hopeless and even one forward pass each is expensive. A first-order estimate is free. Since $\xi_{\ell,a}$ multiplies the head output, the derivative is (incoming) $\times$ (local), as in Lecture 5:

$$\frac{\partial \mathcal{L}}{\partial \xi_{\ell,a}} = \underbrace{\mathrm{Att}_{\ell,a}(x)}_{\text{local}} \cdot \underbrace{\frac{\partial \mathcal{L}}{\partial\, \mathrm{Att}_{\ell,a}(x)}}_{\text{incoming}}, \qquad I_{\ell,a} = \mathbb{E}_x \left| \mathrm{Att}_{\ell,a}(x) \cdot \frac{\partial \mathcal{L}}{\partial\, \mathrm{Att}_{\ell,a}(x)} \right|$$

which is the score of [Michel et al., 2019]: one backward pass ranks all $L \times H$ heads at once, across layers rather than within each. At a single position $\mathrm{Att}_{\ell,a}(x) \in \mathbb{R}^{d_k}$ and its gradient has the same shape, so the product is an inner product and $I_{\ell,a}$ is one number per head. $\xi : 1 \to 0$ is not a small perturbation, so $I_{\ell,a}$ ranks rather than predicts — and redundancy means one-at-a-time removal understates how much the set matters.

The empirical finding: most heads can go with small loss, and the survivors are not spread evenly across layers. Plotting accuracy against heads removed, the curve stays flat and then drops, rather than falling in a line [Michel et al., 2019; Voita et al., 2019].

And then?

Ablation tells us a component was *necessary*. It does not tell us it was sufficient, or what it computed. Getting at those means intervening on *representations* rather than deleting components — removing a property from a layer and testing whether behavior that should depend on it degrades [Elazar et al., 2021], or copying activations between runs and measuring the effect downstream. That is where the field went after BERTology, and where we pick up when we return to interpretability.

8. What to take from this

- BERT is last lecture's architecture with the mask removed and reconstruction in place of next-token prediction.
- The BERTology findings are mostly *decodability* results — real measurements of something that is not "what the model computes."
- Locate a method's claim before accepting its headline.