Amnesic Probing and INLP

Group 1: Qingxia Guo, Saiya Karamali, Lindsay Skinner and Gladys Wang
May 11th 2022

Overview

1. Amnesic probing part 1
   1.1. Background
   1.2. Method
2. INLP
   2.1. Overview of INLP
   2.2. Two examples applying INLP: algebraic and visual
   2.3. Some applications of INLP
3. Amnesic probing part 2
   3.1. More method
   3.2. Experiments
4. Our experiment

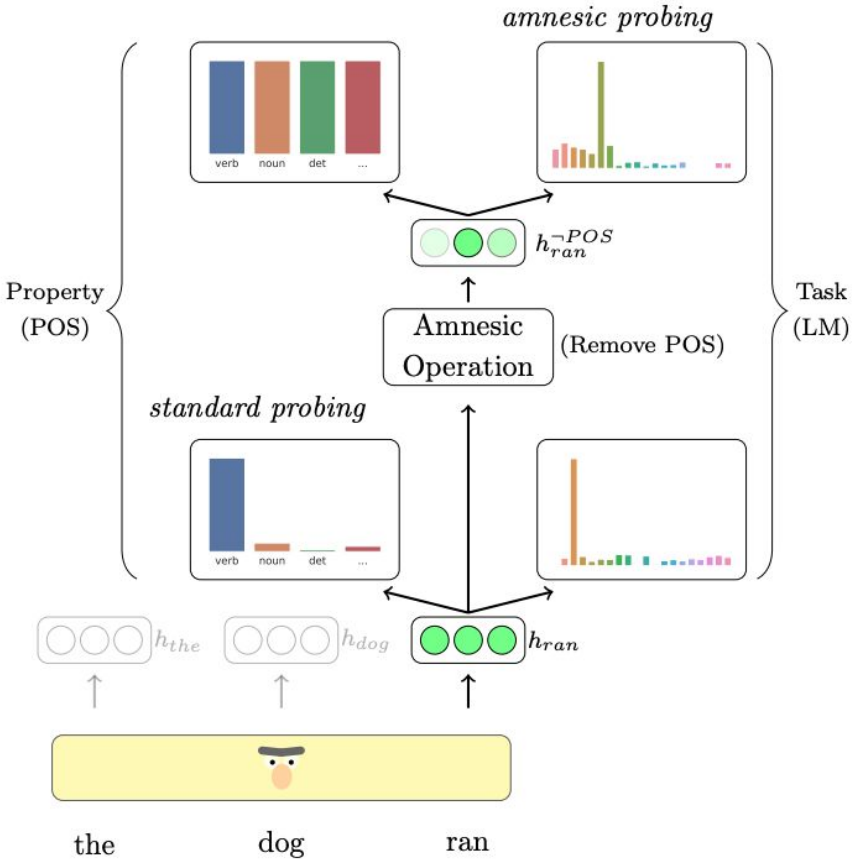
Amnesic Probing

Behavioral Explanation with Amnesic Counterfactuals

Amnesic Probing: Background

- Information can be extracted from the representation ≠ information is used for a specific task
- Core idea: If some information is used for a task, then removing such information from the representation should have negative impact
- Objective: evaluate the effectiveness of the proposed amnestic probing method

Amnesic Probing: Method



But wdym by removing?
🤔

“Null it out”

An overview of Iterative Nullspace Projection

Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection

Shauli Ravfogel[1,2] **Yanai Elazar**[1,2] **Hila Gonen**[1] **Michael Twiton**[3] **Yoav Goldberg**[1,2]
[1]Computer Science Department, Bar Ilan University
[2]Allen Institute for Artificial Intelligence
[3]Independent researcher
{shauli.ravfogel, yanaiela, hilagnn, mtwito101, yoav.goldberg}@gmail.com

Abstract

The ability to control for the kinds of information encoded in neural representation has a variety of use cases, especially in light of the challenge of interpreting these models. We present Iterative Null-space Projection (INLP), a novel method for removing information from neural representations. Our method is based on repeated training of linear classifiers that predict a certain property we aim to remove, followed by projection of the representations on their null-space. By doing so, the classifiers become oblivious to that target property, making it hard to linearly separate the data according to it. While applicable for multiple uses, we evaluate our method on bias and fairness use-cases, and show that our method is able to mitigate bias in word embeddings, as well as to increase fairness in a setting of multi-class classification.

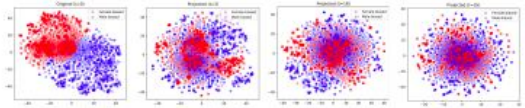


Figure 1: t-SNE projection of GloVe vectors of the most gender-biased words after t=0, 3, 18, and 35 iterations of INLP. Words are colored according to being male-biased or female-biased.

demographics of the author of the text (Blodgett et al., 2016; Elazar and Goldberg, 2018).

What can we do in situations where we *do not want* our representations to encode certain kinds of information? For example, we may want a word representation that does not take *tense* into account, or that does not encode *part-of-speech* distinctions. We may want a classifier that judges the *formality* of the text, but which is also oblivious to the *topic* the text was taken from. Finally, and also our em-

Related methods

- Adversarial methods
  - Use an adversary network that tries to extract protected information from an encoder
  - Difficult to train and can be computationally expensive
- Nullspace cleaning
  - Removes the null-space of the pre-trained classifier in order to remove information that is not used for the main task
  - Not exhaustive and not designed to remove protected attributes
- Projection onto user-defined subspaces
  - E.g. Protect gender information by removing the projection onto the user-defined gender subspace
    - e.g. span{(he - she), (king - queen), (Mister - Miss), etc.} + PCA
  - “...these methods only cover up the bias… in fact, the information is deeply ingrained in the representations.” (Ravfogel, et. al., p.2)

INLP: Purpose

3 Objective and Definitions

Our main goal is to "guard" sensitive information, so that it will not be encoded in a representation. Given a set of vectors $x_i \in \mathbb{R}^d$, and corresponding discrete attributes Z, $z_i \in \{1, ..., k\}$ (e.g. race or gender), we aim to learn a transformation $g : \mathbb{R}^d \to \mathbb{R}^d$, such that z_i cannot be predicted from $g(x_i)$. In this work we are concerned with "linear guarding": we seek a guard g such that no linear classifier $w(\cdot)$ can predict z_i from $g(x_i)$ with an accuracy greater than that of a decision rule that considers only the proportion of labels in Z. We also wish for $g(x_i)$ to stay informative: when the vectors x are used for some end task, we want $g(x)$ to have as minimal influence as possible on the end task performance, provided that z remains guarded. We use the following definitions:

The INLP Algorithm

1. Let X be the set of vectors we wish to guard, C be the set of protected attributes we wish to guard against, and P(x)=x be the identity projection.
2. Train a linear classifier, W, which, for each $x \in X$, uses P(x) in order to predict the affiliated category $c \in C$, with some accuracy.
3. If that accuracy is greater than a proportion-based decision rule…
   a. Define $P_i(x)$ be the function that projects P(x) onto the null space of W
   b. Let $P(x) = P_i(P(x))$
   c. Return to step 2
4. P(x) is the desired guarding function

An Algebraic Example: Set up

$$X_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad X_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad X_3 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

$$X_1 \rightarrow C_1$$

$$X_2 \rightarrow C_1$$

$$X_3 \rightarrow C_2$$

An Algebraic Example: Determine the first linear classifier

$$W_1 = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad b_1 = \begin{pmatrix} -1 \end{pmatrix}$$

$$W_1 X_1 + b_1 = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \end{pmatrix} = \begin{pmatrix} 1 \end{pmatrix}$$

$$W_1 X_2 + b_1 = \begin{pmatrix} 1 \end{pmatrix}$$

$$W_1 X_3 + b_1 = \begin{pmatrix} -1 \end{pmatrix}$$

An Algebraic Example: Project onto Null(W_1)

Null(W_1) = {all vectors in $\mathbf{R}^5$ with a 0 in the first entry}

$$X_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = P_1(X_1)$$

$$X_2 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = P_1(X_2)$$

$$X_3 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = P_1(X_3)$$

An Algebraic Example: Determine the next linear classifier

$$W_2 = \begin{pmatrix} 0 & 0 & 0 & -2 & 0 \end{pmatrix} \qquad b_2 = \begin{pmatrix} 1 \end{pmatrix}$$

$$W_2\, P_1(X_1) + b_2 = \begin{pmatrix} 0 & 0 & 0 & -2 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \end{pmatrix} = \begin{pmatrix} 1 \end{pmatrix}$$

$$W_2\, P_1(X_2) + b_2 = \begin{pmatrix} 1 \end{pmatrix}$$

$$W_2\, P_1(X_3) + b_2 = \begin{pmatrix} -1 \end{pmatrix}$$

An Algebraic Example: Project onto Null(W_2)

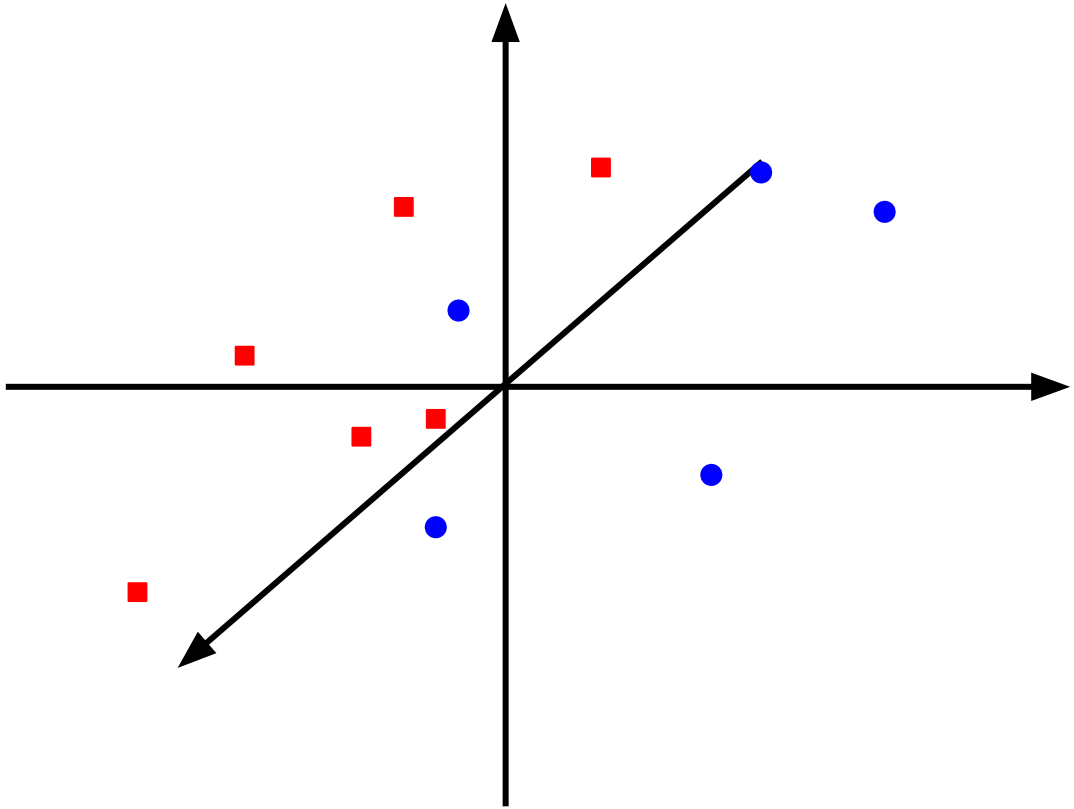
Null(W_2) = {all vectors in $\mathbf{R}^5$ with a 0 in the fourth entry}

$$P_1(X_1) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = P_2(P_1(X_1)) = P(X_1)$$

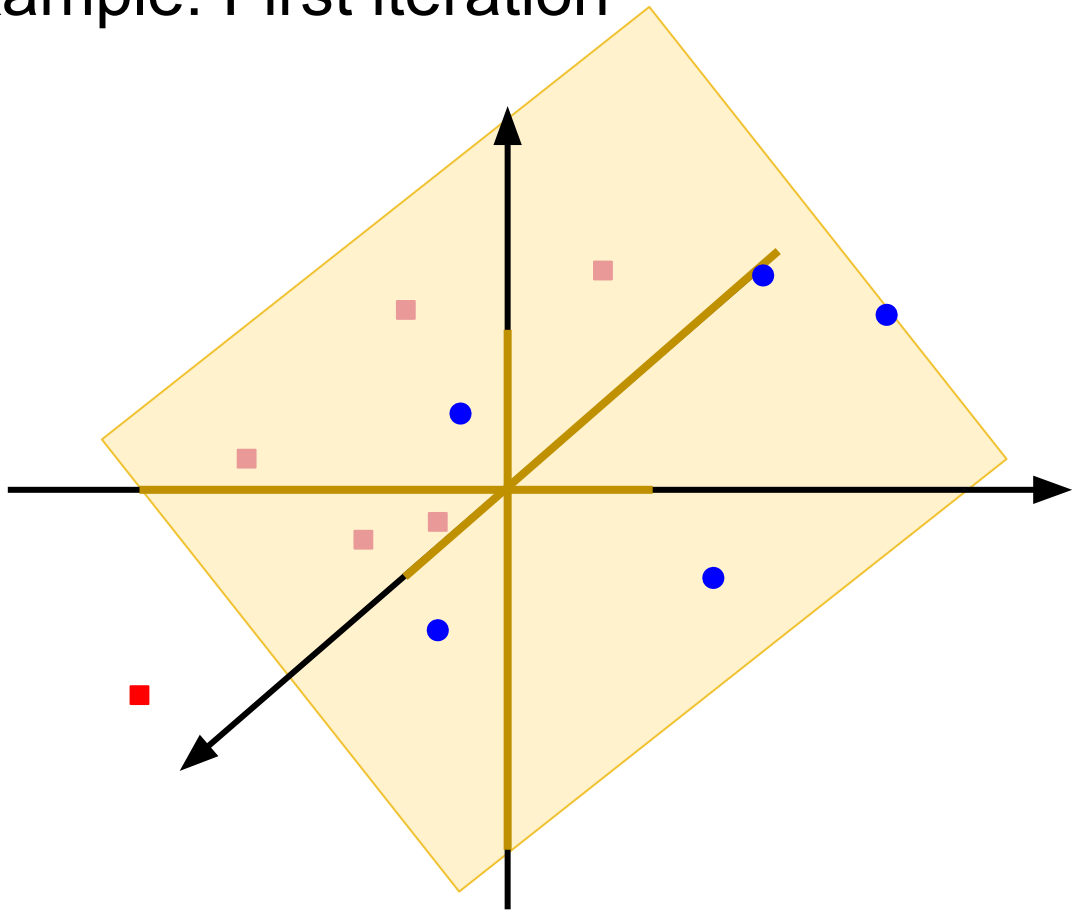
$$P_1(X_2) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = P_2(P_1(X_2)) = P(X_2)$$

$$P_1(X_3) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = P_2(P_1(X_3)) = P(X_3)$$

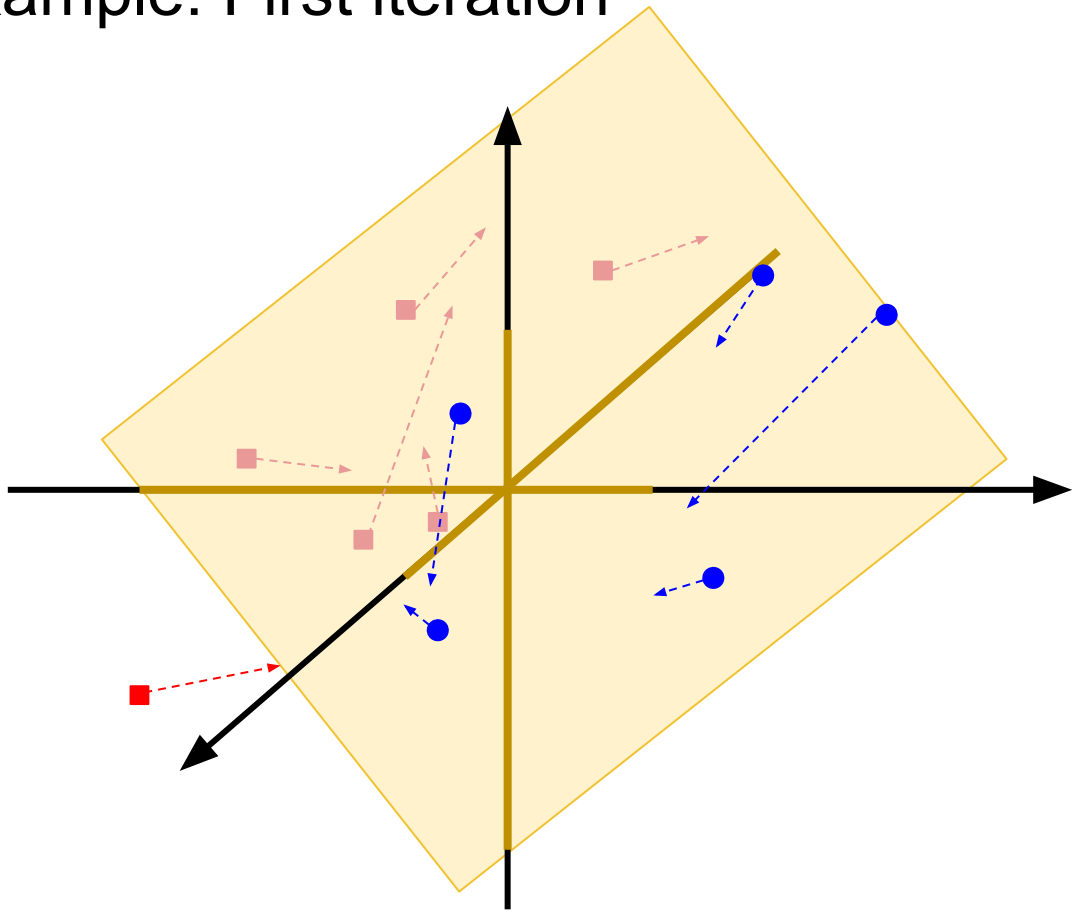
A Visual Example: First iteration



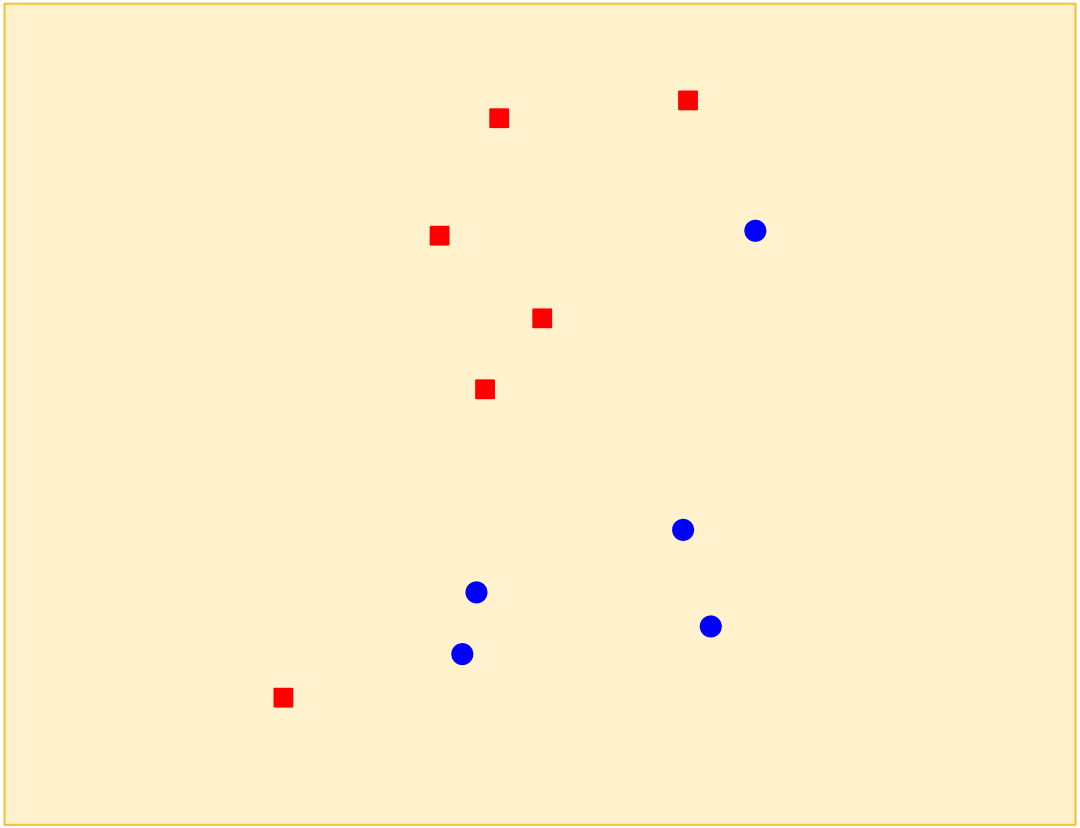
A Visual Example: First iteration



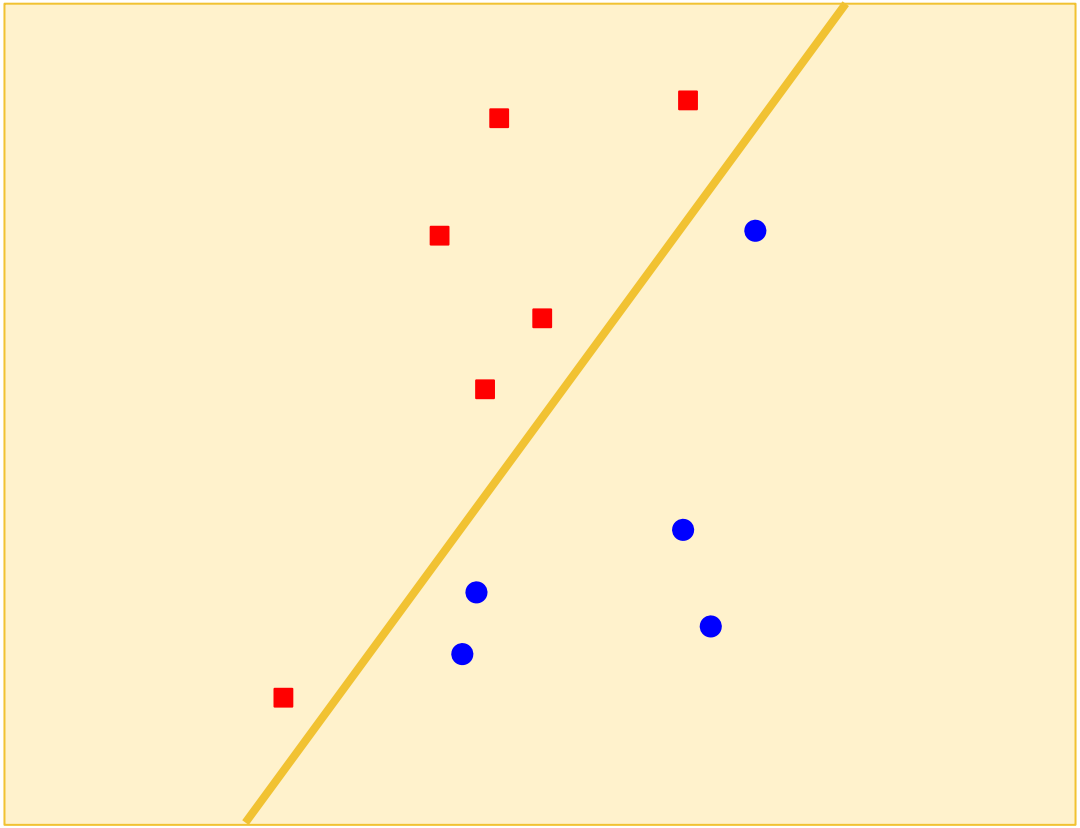
A Visual Example: First iteration



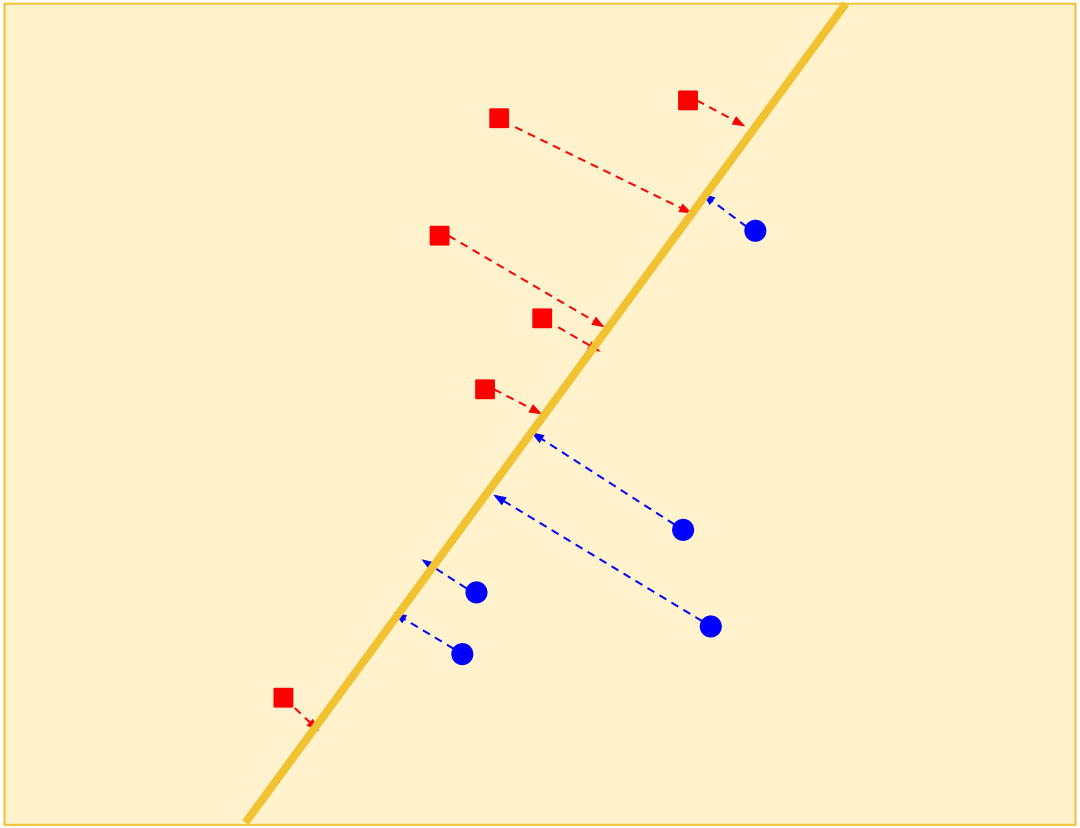
A Visual Example: Second iteration



A Visual Example: Second iteration

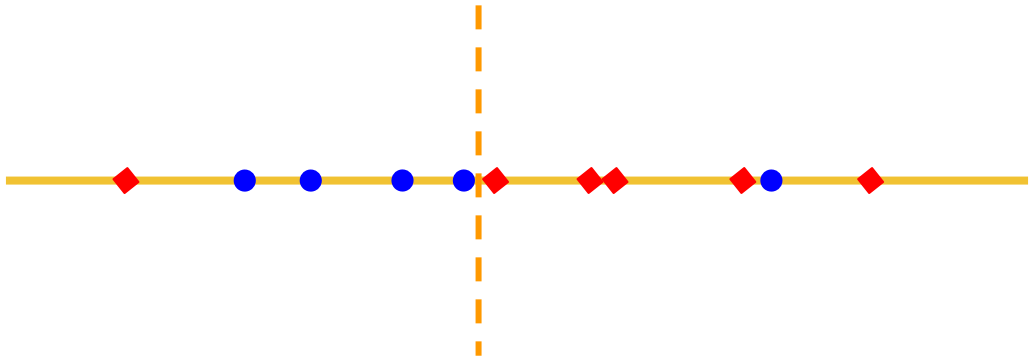


A Visual Example: Second iteration

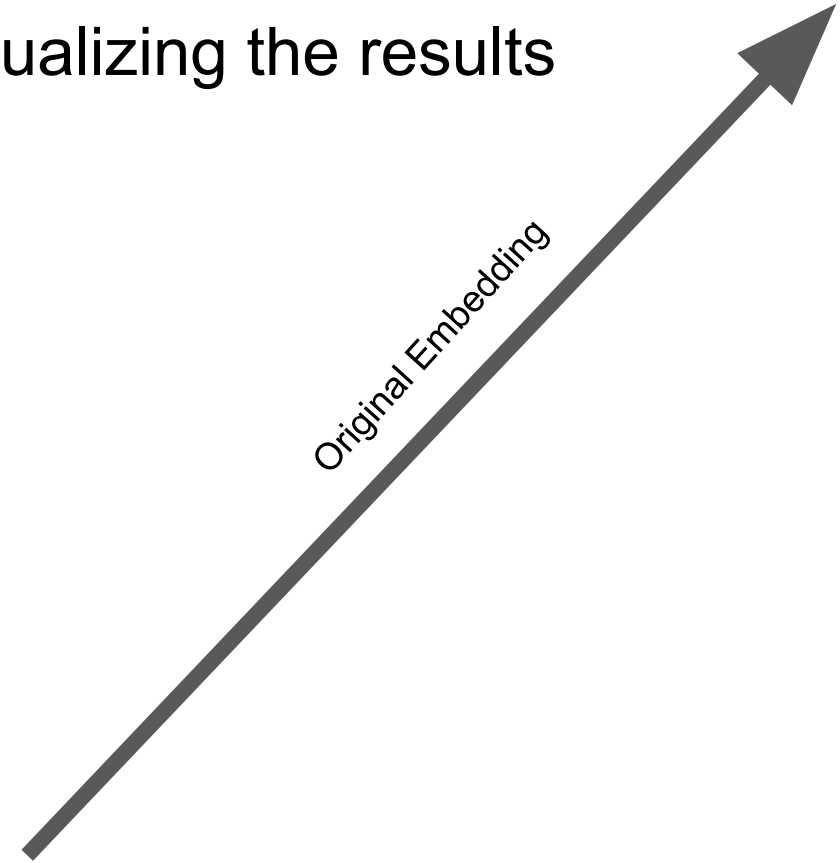


A Visual Example: Third iteration

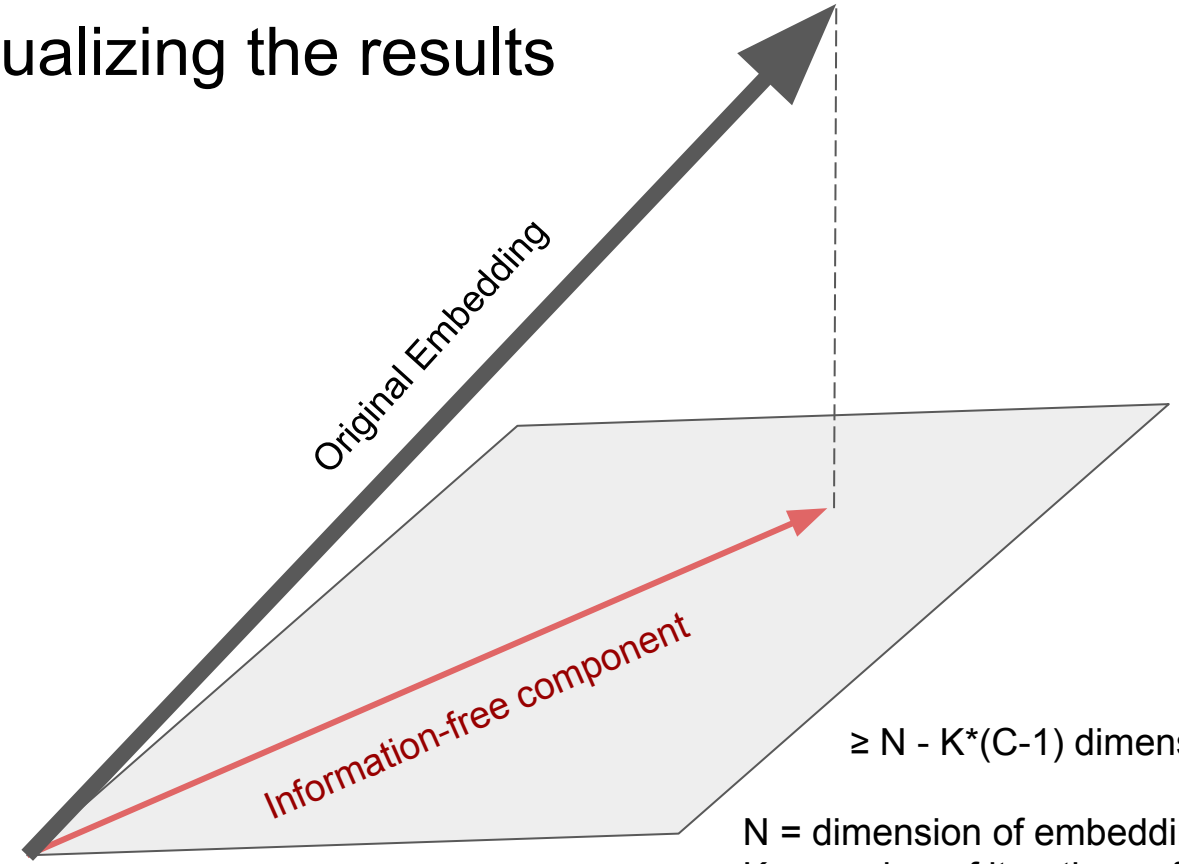
A Visual Example: Third iteration



INLP: Visualizing the results



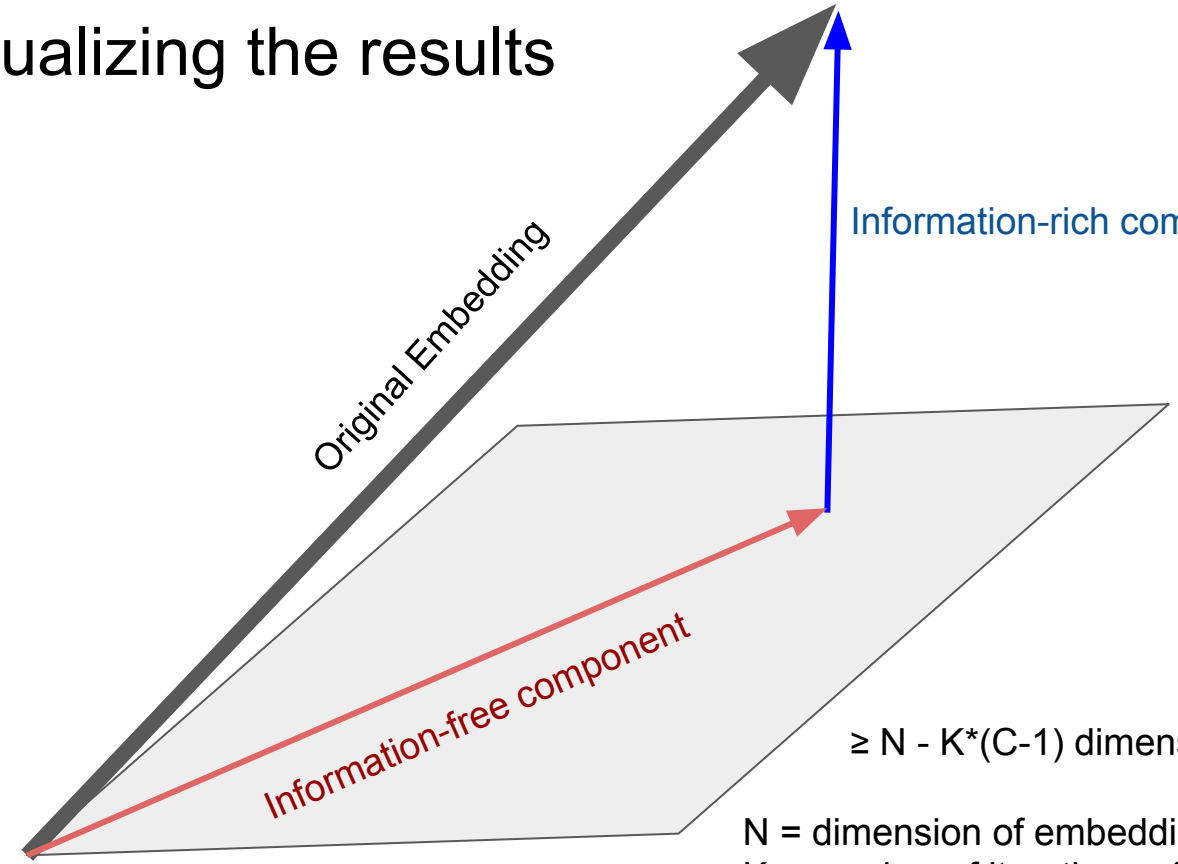
INLP: Visualizing the results



≥ N - K*(C-1) dimensional subspace

N = dimension of embedding space
K = number of iterations of INLP
C = number of classes for the protected attribute

INLP: Visualizing the results



≥ N - K*(C-1) dimensional subspace

N = dimension of embedding space
K = number of iterations of INLP
C = number of classes for the protected attribute

Some applications of INLP

- Guarding gender information (explored in the “Null It Out” paper)
- Guarding protected attributes for hiring or loan approvals
  (e.g. gender, race, age)
- Determining language-specific and language-agnostic components in mBERT
  - Potential applications to translation tasks
- Determining if BERT uses POS information in the language modeling task…

Amnesic Probing: Method, Resumed

- Control
  - Control over information
    - Create baseline ‘guarding’ function that removes the same number of directions as INLP does, but randomly
  - Control over selectivity
    - Fine tune the subsequent layers with gold information of the property that is removed.
    - Restoration of original performance is evidence that the property we removed can account for the damage to the model’s performance

Amnesic Probing: Experiment Setup

- Model
  - BERT
- Properties
  - Coarse and fine-grained part-of-speech tagging (*c-pos* and *f-pos*)
  - Syntactic dependency labels (*dep*)
  - Named-entity labels (*ner*)
  - Beginning and end of a phrase (*phrase start* and *phrase end*)
- Measures
  - LM accuracy
  - Kullback-Leibler Divergence for distributions before and after amnesic intervention
    - Measures how a probability distribution is different from another
    - Larger = greater change.

Amnesic Probing: Experiment 1

- Naive probing vs amnesic probing

		dep	*f-pos*	*c-pos*	*ner*	*phrase start*	*phrase end*
Properties	N. dir	738	585	264	133	36	22
	N. classes	41	45	12	19	2	2
	Majority	11.44	13.22	31.76	86.09	59.25	58.51
Probing	Vanilla	76.00	89.50	92.34	93.53	85.12	83.09
LM-Acc	Vanilla	94.12	94.12	94.12	94.00	94.00	94.00
	Rand	12.31	56.47	89.65	92.56	93.75	93.86
	Selectivity	73.78	92.68	97.26	96.06	96.96	96.93
	Amnesic	7.05	12.31	61.92	83.14	94.21	94.32
LM-D_{KL}	Rand	8.11	4.61	0.36	0.08	0.01	0.01
	Amnesic	8.53	7.63	3.21	1.24	0.01	0.01

Amnesic Probing: Experiment 2

- Naive probing vs amnesic probing, but with masked representations

		dep	*f-pos*	*c-pos*	*ner*	*phrase start*	*phrase end*
Properties	N. dir	820	675	240	95	35	52
	N. classes	41	45	12	19	2	2
	Majority	11.44	13.22	31.76	86.09	59.25	58.51
Probing	Vanilla	71.19	78.32	84.40	90.68	85.53	83.21
LM-Acc	Vanilla	56.98	56.98	56.98	57.71	57.71	57.71
	Rand	4.67	24.69	54.55	56.88	57.46	57.27
	Selectivity	20.46	59.51	66.49	60.35	60.97	60.80
	Amnesic	4.67	6.01	33.28	48.39	56.89	56.19
LM-D_{KL}	Rand	7.77	6.10	0.45	0.10	0.02	0.04
	Amnesic	7.77	7.26	3.36	1.39	0.06	0.13

Amnesic Probing: Experiment 3

- How does removing *c-pos* affect the model's accuracy in predicting words from each category?

c-pos	Vanilla	Rand	Amnesic	Δ
verb	46.72	44.85	34.99	11.73
noun	42.91	38.94	34.26	8.65
adposition	73.80	72.21	37.86	35.93
determiner	82.29	83.53	16.64	65.66
numeral	40.32	40.19	33.41	6.91
punctuation	80.71	81.02	47.03	33.68
particle	96.40	95.71	18.74	77.66
conjunction	78.01	72.94	4.28	73.73
adverb	39.84	34.11	23.71	16.14
pronoun	70.29	61.93	33.23	37.06
adjective	46.41	42.63	34.56	11.85
other	70.59	76.47	52.94	17.65

Table 3: Masked, *c-pos* removal, fine-grained LM analysis. Removing *c-pos* information and testing the accuracy performance of words, accumulating by their label. Δ is the difference in performance between the Vanilla and Amnesic scores.

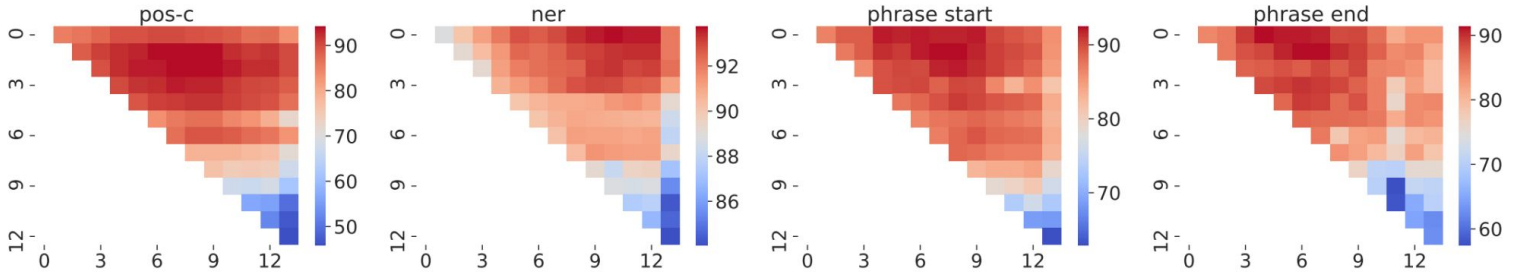
c-pos	Vanilla	Amnesic	Δ
verb	56.98	55.60	1.38
noun	56.98	55.79	1.19
adposition	56.98	53.40	3.58
determiner	56.98	51.04	5.94
numeral	56.98	55.88	1.10
punctuation	56.98	53.12	3.86
particle	56.98	55.26	1.72
conjunction	56.98	54.29	2.69
adverb	56.98	55.64	1.34
pronoun	56.98	54.97	2.02
adjective	56.98	55.95	1.03

Table 4: Word prediction accuracy after fine-grained tag distinction removal, *masked* version. Rand control performance are all between 56.05 and 56.49 accuracy (with a maximum difference from Vanilla of 0.92 points).

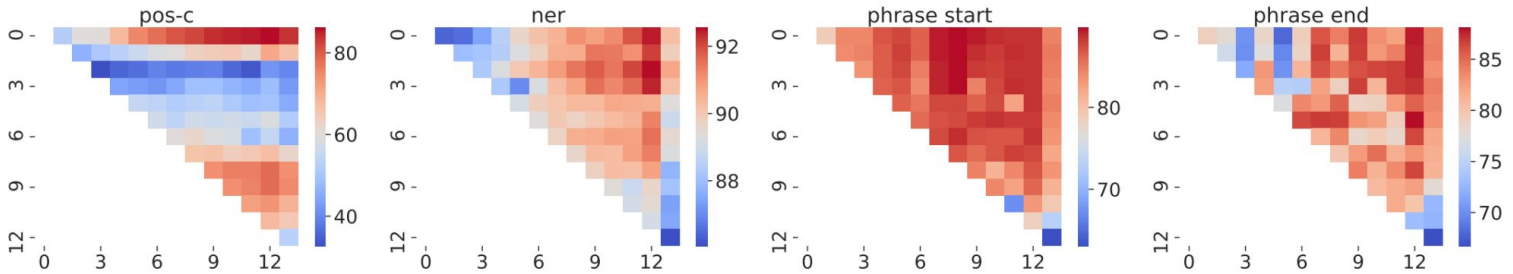
Amnesic Probing: Experiment 4

- Removal of properties in earlier layer, rather than BERT's transformer blocks
- Information relating to the properties should still be recoverable by subsequent layers in non-linear ways
- Two sub experiments
  - Removing information related to a property in an early layer, then measure the probing accuracy for the given property in subsequent layers
  - Removing information related to a property in an early layer, then measure the LM accuracy at the final layer

Amnesic Probing: Experiment 4.1

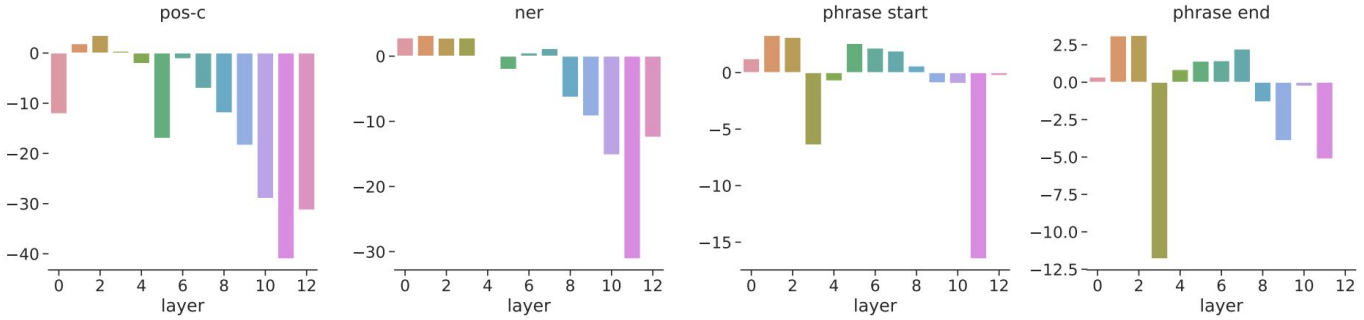


(a) Non-Masked version

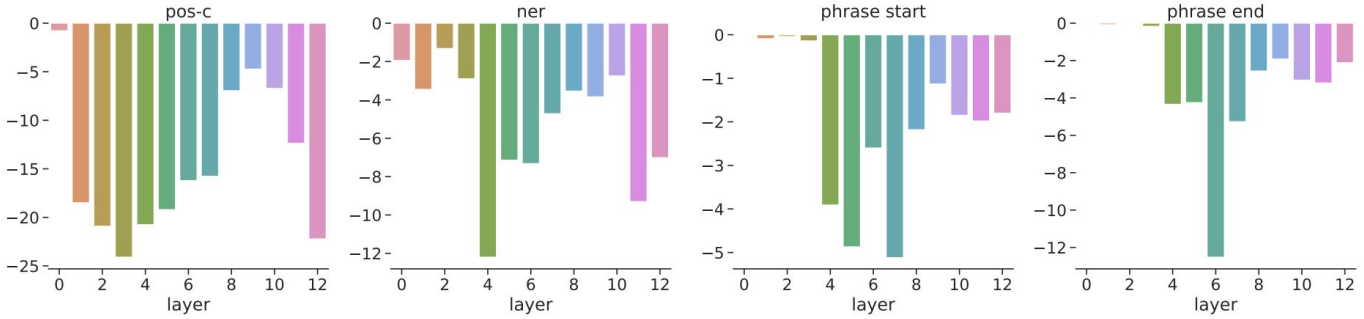


(b) Masked version

Amnesic Probing: Experiment 4.2



(a) Non-Masked version



(b) Masked version

Amnesic Probing: Discussion & Conclusion

- A method to quantify the influence of certain properties on a model's performance on a specific task
  - Does not quantify the *relative importance* of different properties
  - Differences between two versions of the model (masked vs unmasked) in this paper not cross comparable

Our Experiment

- Use INLP to decompose BERT’s representation into semantic & non-semantic, and syntactic & non-syntactic components
- By measuring a linear classifier’s performance on POS tagging and semantic role labeling for each of the above components, we can draw conclusions about the importance of syntactic information in semantic tasks and vice versa

References

- Yanai Elazar, Shauli Ravfogel, Along Jacovi, and Yoav Goldberg. 2020. Amnesic Probing: Behavioral explanation with amnesic counterfactuals.
- Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection.
- Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. 2020. It’s not Greek to mBERT: Inducing Word-Level Translation from Multilingual BERT.