Probing for Neurons that Encode and Exploit Negation in BERT

Alexis J. Ross
Harvard University

Sean J. Sullivan
Harvard University

Tynan J. Seltzer
Harvard University

Abstract

BERT has been shown to obtain state-of-the-art results on eleven natural language tasks, including a 5.6% absolute improvement in accuracy on the MultiNLI task to 86.7% (Devlin et al., 2018; Williams et al., 2017). However, the question of what information learned in BERT pre-training is responsible for this increased performance remains an open question. In this work, we propose a method of finding specific neurons in BERT that encode information about linguistic features such as negation explicitly used by the model for NLI classification. Drawing from methods in causal inference, we evaluate both the direct and indirect effects of the linguistic feature negation on NLI predictions. Our results suggest that there are no specific neurons that encode and exploit information about negation in the top three layers of the model.

1 Introduction

Work on creating models capable of general natural language understanding has focused on learning sentence embeddings through some pre-training task and evaluating these embeddings by feeding them as inputs to downstream natural language processing tasks (Conneau et al., 2017; Bowman et al., 2019). GLUE, a suite of eleven NLP tasks testing understanding of different linguistic phenomena, has become a standard set of such downstream tasks used to evaluate sentence embeddings (Devlin et al., 2018).

Recently, the sentence representation model BERT, Bidirectional Encoder Representations from Transformers, achieved state-of-the-art results on GLUE, with a 7.6% absolute improvement in overall GLUE score (Devlin et al., 2018). These results also included a 5.6% absolute improvement in accuracy on MNLI, a multi-genre dataset for the natural language inference (NLI) task. In NLI, an example consists of a premise sentence and a hypothesis sentence, with the task of determining whether the relationship between the sentences is one of entailment, contradiction, or neutrality. MNLI consists of 433,000 of these examples, with crowdsourced labellings.

BERT's state-of-the-art performance on GLUE, along with other recent results, (White et al., 2017; Devlin et al., 2018; Bowman et al., 2019; Radford et al., 2019) suggest that models pre-trained to model language in a general sense via next-word prediction are effective at learning general language properties because they are able to transfer what they learn to a plethora of heterogeneous tasks with little retraining. What is unknown, however, is the method by which these generalizable models are able to adapt to these different tasks. The optimistic theory claims that because models like BERT learn underlying properties of language that are task agnostic, generalization is possible in the same way humans complete these differing tasks. However, benchmarks like GLUE fail to elucidate exactly what is learned during BERT pretraining. Thus, researchers have developed more fine-grained techniques for exploring what properties of language models like BERT exploit in order to do well on such tasks.

2 Related Work

More fine-grained probing analysis of what is learned by sentence embedding methods for NLI has typically made use of challenge NLI datasets and auxiliary classifiers (Belinkov and Glass, 2018). Challenge datasets are sets of NLI pairs where the hypothesis and premise differ minimally, designed to test for a specific linguistic phenomenon (Poliak et al., 2018; Dasgupta et al., 2018; Kim et al., 2019). For example, researchers have used examples like the following to test

for a subsequence heuristic (predicting entailment when the hypothesis appears as a subsequence in the premise):

p : Alice believes Mary is lying.

h : Alice believes Mary.

In such an example, a model relying on the subsequence heuristic would incorrectly predict *entailment* (McCoy and Linzen, 2018).

Auxiliary classifiers have also been trained to predict a linguistic property of interest given some pre-trained embedding as input (Adi et al., 2016; Conneau et al., 2018; Giulianelli et al., 2018). A successful classifer in this task demonstrates that the embeddings indeed encode that property.

However, while both of these approaches can be used to probe what linguistic properties are *encoded* in sentence embeddings, a downside of these methods is that they cannot show whether those properties are actually *exploited* by the classifiers in making predictions for downstream tasks such as NLI. For instance, Vanmassenhove (2017) trained a diagnostic classifier on neural machine translation sentence embeddings and found that though tense could be accurately predicted around 90% of the time, output translations only had correct tense 79% of the time, suggesting that encoded information is not always used downstream. Thus, these methods are not able to answer the question: What information in these sentence embeddings that is learned during pre-training is *responsible* for improved performance on downstream tasks? Simply determining whether linguistic information is encoded in representations cannot answer this question because such encoded information may not ultimately used in downstream tasks.

In this paper, we provide one approach to answering this causal question by combining measures of whether information encoded with measures of their causal effects. We add to the line of work probing sentence embedding methods with a method analyzing the contribution of individual dimensions of BERT embeddings to NLI. Our approach is adapted from that proposed by Bau et al. (2019) for neural machine translation models and identifies causal effects of linguistic features on predictions for NLI. In particular, we measure the direct and indirect causal effects of negation mediated through individual neurons. Though we apply the method to layers of BERT, the method is model-agnostic and can be applied to dimensions in any model before a classification layer.

3 Problem Set-Up

3.1 Linguistic Phenomenon: Negation

The linguistic phenomenon we probe in this paper is negation. More specifically, we investigate the effect of negation in the premise of an NLI pair. For each original NLI pair (p, h), we create another pair (p', h), where p' is the negated version of p. All MNLI pairs in our constructed test set are originally taken from the MNLI *dev matched* dataset (Williams et al., 2017). In Section 4.1, we describe how we create each negated p'.

We probe how BERT encodes and exploits the property of negation in particular because it has been shown that the ability to use negation is critical for doing well on MNLI, with models learning strong biases to predict *contradiction* when there is a negation present in the NLI pair (Dasgupta et al., 2018).

3.2 Causal Effects

We investigate both the direct and indirect causal effects (Pearl, 2013) of linguistic properties on output NLI predictions as mediated by individual neurons in BERT. We probe for causal effects by manipulating the values of embeddings' dimensions before they are given as input into a linear NLI classification layer. Our formal approach is described below and expanded upon in Section 4.3.

Let (p, h) and (p', h) denote two NLI premise/hypothesis pairs, where p and p' differ only with respect to the linguistic property being probed. In this paper, this linguistic property is negation in the premise. p' then represents the negated version of p. Call x and x' the NLI pairs (p, h) and (p', h) respectively. $M_i(x)$ represents the activation of the ith dimension of a learned BERT sentence embedding when x is given as input into BERT. Similarly, $M_i(x')$ represents the activation of the ith dimension when x' is given as input into BERT. Let $Y(x*, m))$ represent the probability of *entailment* predicted by the NLI classifier when NLI pair version $x*$ is given as input and m value, of the form $M_i(x)$ or $M_i(x')$, is used. We use this probability as our predicted variable rather than the discrete predicted label value so that the analysis can be done in continuous space.

3.3 Direct Effects

We measure the *direct* effect of negation on the probability of entailment predicted for a given NLI pair (p, h) as follows:

$$DE(p', p) = E[Y(p', M_i(p))] - E[Y(p, M(p))]$$

3.4 Indirect Effects

We measure the *indirect* effect of negation on the probability of entailment predicted for a given NLI pair (p, h) as follows:

$$IE(p', p) = E[Y(p, M_i(p'))] - E[Y(p, M(p))]$$

3.5 Correlational Measurements

In addition to measuring direct and indirect effects, we wanted to measure whether a neuron encoded for negation uniquely. That is, we wanted to see whether the magnitude of activation for a candidate neuron for an NLI input was correlated uniquely with the presence of negation in the input and not with its prediction for the input. To show this, we examined the distribution of activations of each neuron modified across 4 groups:

1. Training samples that are negated and output not entailed (- *not entailed*)
2. Training samples that are positive and output not entailed (+ *not entailed*)
3. Training samples that are negated and output entailment (- *entailed*)
4. Training samples that are positive and output entailment (+ *entailed*)

Finding a differential activation for a neuron across the two negation groups with minimal difference in activation across the two *entailment* or *not entailment* groups would show a neuron to correlate with negation but not with other undesired factors. This method would allow us to say that a particular neuron encodes the presence of *negation*, as opposed to a bias towards a *not entailment* prediction, which might otherwise be conflated because negation is in general correlated with a prediction of *not entailed*.

3.6 Hypothesis

The specific question we wanted to investigate was the following: To what extent is the effect of negation mediated by an individual neuron?

To answer this question, we looked at a combination of the correlational measure of the extent to which the neuron encoded negation, as well as the causal effects of the neuron on the model's prediction.

More specifically, we first used the correlational measurements to assess whether the activation value of a particular neuron encoded information about whether there was negation present in the premise. We then looked to see whether that particular neuron had a strong causal (direct or indirect) effect on the output of the model. This combination of results would allow for the conclusion that the particular neuron was a "not-neuron", encoding information about negation used by the model in its calculations. Without the correlational measure verifying that the neuron was uniquely encoding *negation*, any neuron with a strong causal effect would be a "contradiction-neuron." And any neuron encoding negation but not causally affecting the model would just be a negation-encoding neuron.

4 Probing Methodology

4.1 Data

We create p' by negating the main verb of p. We identify the main verb of p using the original parse trees that come with the MNLI examples. More specifically, we take the MNLI *dev matched* dataset and use a heuristic to exclude already negated examples (by filtering out examples with the words *never, not,* and *n't*). With the remaining examples in this filtered dev set, we negate the first verb of the sentence using the verb conjugator from `pattern.en`[1]. For instance, the following pair of NLI inputs represents an example of (p, h) and (p', h):

> p : Dozens of additional militants arrived on later flights.
>
> p' : Dozens of additional militants did **not** arrive on later flights.
>
> h : The blockade stopped militants from arriving by aircraft.

Our resulting dataset consisted of $6,341$ unique NLI pairs, each with one positive version and one negated version.

[1] https://www.clips.uantwerpen.be/pages/pattern-en

4.2 BERT Architecture

BERT consists of an embedding layer, followed by 12 transformer blocks as in (Vaswani et al., 2017), then a fully connected dense layer with tanh activation, and finally a simple linear layer for classification.

4.3 Probing Neurons for Effects

In order to measure the direct and indirect effects of individual neurons in BERT, we modify the value of a single neuron and observe the change in output probabilities between the two possible classes, *entailment* and *not entailment*.

To measure the effects of neurons in a particular layer, we implement a modified forward method for BERT, which records the activations of the specified layer of neurons for a single MNLI example. Secondly, we implement a second forward method that passes a modified layer through the remaining layers of the model. Thus, to change the value of an individual neuron at some layer of BERT, we simply run the example we wish to use through the model, take the neuron activations at the desired layer, then modify the activation value of a single neuron, and use our second forward method to then push these neuron activations through the rest of the model.

4.3.1 Direct Effects

In order to measure the direct effects of a single neuron at a specific layer as defined in Section 3.3, we feed an MNLI pair and the corresponding negated pair created from Section 4.1 through the model. We then take the neuron activations at the desired layer from the the negated example, and change the value of the target at that layer to be the activation value of the same neuron at the same layer of the positive MNLI example. We push this modified neuron activation set through the rest of the model, and record the difference between this output entailment probability and that of the original positive example.

4.3.2 Indirect Effects

Measuring the indirect effects as defined in Section 3.4 is similar to measuring the direct effects. Instead of swapping a single neuron activation from the positive MNLI example into the negated one at a specific neuron and layer, we swap a single neuron activation from the negated MNLI example into the positive example, push the modified activation set through the rest of the model, and measure the difference between the output entailment probability of the original positive example, and this modified example.

5 Experiments

5.1 Models

We probe a fine-tuned BERT model, where the weights of both the classification layer weights and pre-trained layers are fine-tuned on MNLI. The model is fine-tuned for 3 epochs and uses the original hyperparameters described by Devlin et al. (2018). Additionally, we train a copy of the model on each of two versions of MNLI, which correspond to two methods of removing examples with the label *neutral* from the data.

Binary: The first method is to simply consider as the dataset only those examples labeled as either *entailment* or *contradiction*, shrinking our data set by approximately one third. We will refer to these *contradiction* examples as *not entailment* examples for consistency.

Full Binary: The second is to consider the *neutral* and *contradiction* examples as a single class, *not entailment*. This is consistent with previous work from (Wang et al., 2018), which used this method to collapse three class tasks into binary tasks for the GLUE tasks.

We refer to these two modified data sets as "Binary" (*neutrals* removed), and "Full Binary" (*neutrals* and *contradiction* forming a single class). We remove neutral examples to simplify our method for measuring effects, which involves comparing probabilities of the output classes.

5.2 Layers Probed

For each of the two models mentioned in section 5.1, we explore the neurons of three different layers of the BERT architecture.

Layer 1: The first layer we probe is the pooled output layer of the whole model, which represents the hidden state of the CLS token and is fed through the final linear classifier to generate the inference prediction. This layer of neurons has size equal to the hidden dimension of BERT, which is 768.

Layer 2: The second layer we probe is an extra pooled output layer that is transformed into our first probed layer by a dense layer with tanh activation. The tanh function provides a non-linearity that means this layer is not simply linearly connected to the final output, unlike the first probed

layer. Like the first probed layer, it also has dimensionality 768.

Layer 3: The third layer we probe is the layer before the final encoder block of Bert. Unlike the previous layers probed, this is not a 1×768-sized layer as the first two layers we probed, but rather a 70×768-sized layer, where 70 is the max sequence length. Each of the 768 dimensions is a set of 70 intra attention values over the sequence of words representing the premise hypothesis pair. To inject change in this layer, we change a full set of attention values at a time, with this layer again having 768 overall dimensions to probe.

6 Results

We found that negation produces a significant total effect on the predictions of both the Binary and Full Binary models. As shown in Table 1, however, this effect was largely only seen for positive examples that were originally labeled as *entailment*, confirming results that models learn a bias to predict *contradiction* when there is a negation present in the NLI pair (Williams et al., 2017).

Regarding our specific hypothesis about "not-neurons," however, the results of our model indicate that, within the final three layers of BERT, there are no individual neurons that encode and exploit negation. In fact, we also did not find neurons *merely* encoding negation or individual neurons *merely* having a strong causal effect on the model's outputs. Across our experiments with layers and models, this result is consistent.

6.1 Correlation Results

To find candidate "not-neurons" encoding negation and mediating its causal effect on models' outputs, we observe the correlation of neuron activation across the four groups from section 3.5. This was meant to identify neurons with large differences in activation between negated and positive sentences but small differences in activation between *not entailment* and *entailment*. For a given layer and model from section 5 we partitioned the sentence examples into the 4 groups based on the output prediction of the model for the positive and negated cases.

Within each of these groups, there is a distibution of activations. To investigate whether the activation for negated examples (- *entailed* and - *not entailed*) were far from that of positive examples (+ *entailed* and + *not entailed*) we calculated the Wasserstein distance between each pair of groups. The Wasserstein distance between distributions measures the density of one distribution that must be moved to turn it into the other. More formally, this is equivalent to the integral of the absolute difference of the CDFs of the two distributions:

$$W(p, q) = \int_{-\infty}^{\infty} |P(x) - Q(x)| dx$$

where P and Q are the CDFs of p and q respectively. We calculate this distance between the positive and negated distributions and between the *entailment* and *not entailment* distribution. We sum the distances between the positive and negated group distributions and subtract the distance between the *not entailment* and *entailment* group distributions. Looking at the dimensions that maximize this quantity will show neurons that have large differential activation across negated and positive examples but minimal differential activation across entailment and contradication. In order to compare these distances across dimensions, before calculating Wasserstein distance we normalize the activations for a particular neuron to the range (-1, 1).

The results of this experiment are shown in 2 for the binary model and 3 for the full binary model.

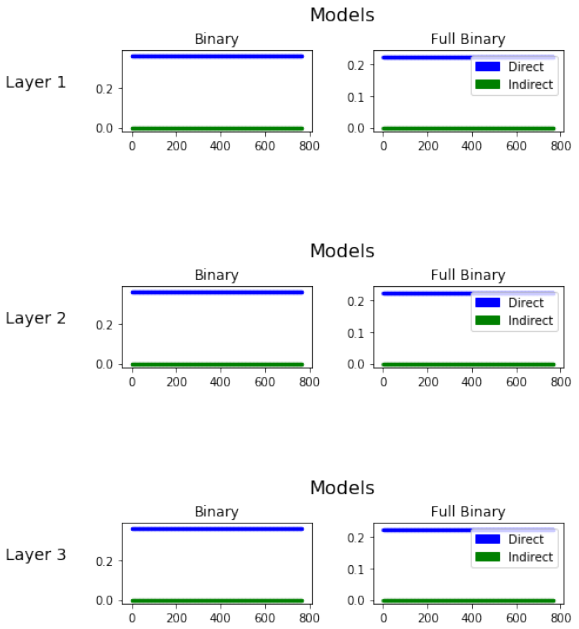
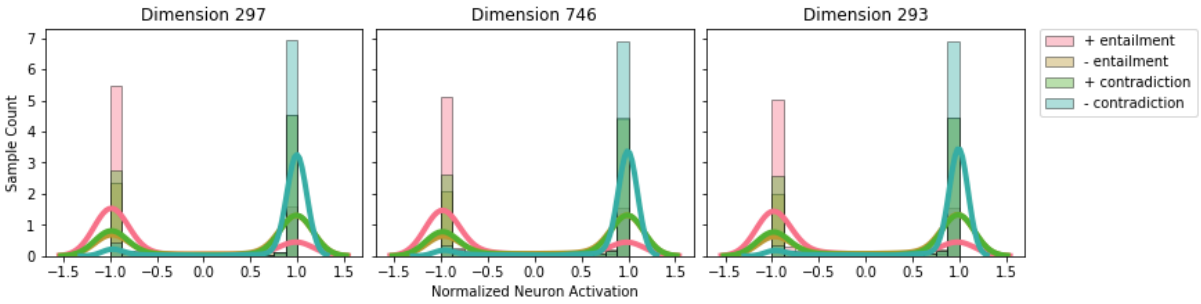


Figure 1: This figure displays the results by model and layer of each neuron's direct and indirect effects on the model's output probabilities of entailment.

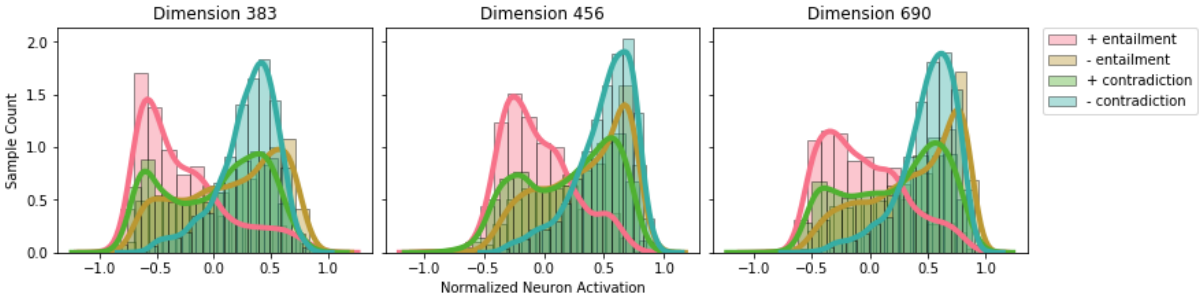
	Dev Matched Acc	Positive Preds	Switch from *Entailed*	Switch from *Not Entailed*
Full Binary:	.877	.318 **E** / .682 **C**	.655	.324
Binary:	.926	.529 **E** / .471 **C**	.578	.121

Table 1: This table gives general results for each of our two models, including accuracy on the original MNLI *dev matched* dataset, as well as prediction breakdowns for our positive/negated constructed datasets. This table highlights that negation caused the models to switch their predictions when their original positive predictions were *entailment*, but not when the initial predictions were *not entailment*

.



(a) Distribution of 3 hidden neuron activations from the final layer of the binary model



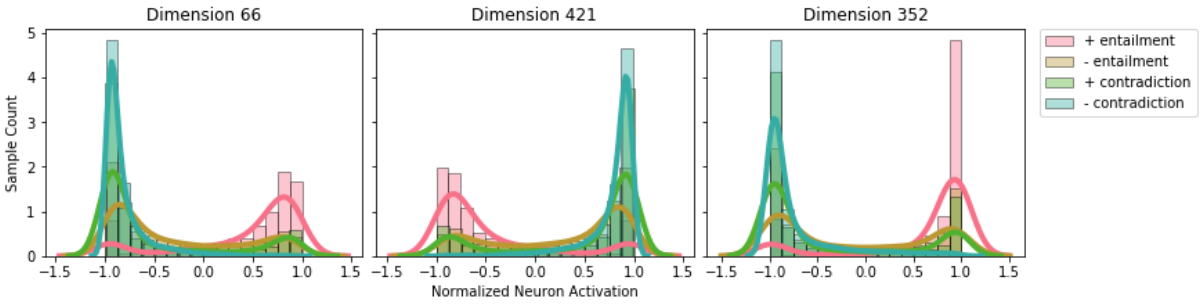
(b) Distribution of 3 hidden neuron activations from the second to last (pooling) layer of the binary model

Figure 2: Distributions of neuron activation (normalized to the range -1.0, 1.0) are shown for various hidden dimensions in the last and second to last layer of the binary model. Four separate groups are shown in each image, for each of the four groups detailed in section 3.5

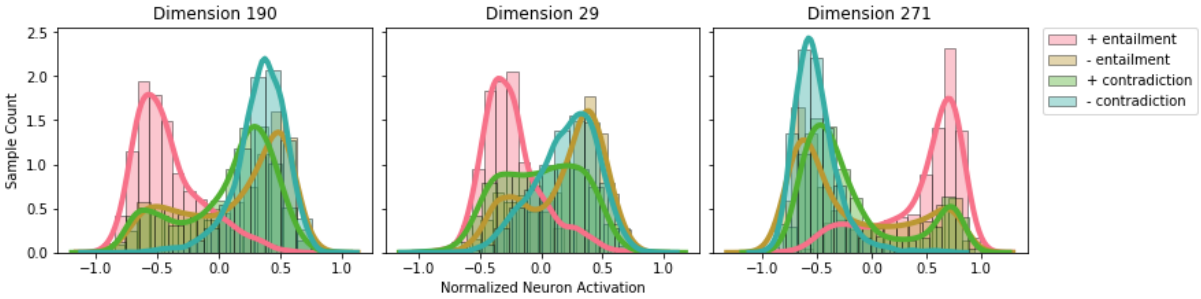
We show results for neurons in layer 1 and layer 2. For layer 3, which is the last self attention layer of BERT, our changes were not to a single activation but rather a sequence of activations, and so the separability of these could not be measured in this way.

Across both graphs, a noticeable trend is that, while the negated *entailment* and positive *entailment* distributions are furthest apart, as expected, the positive distribution does not overlap much with the other distributions in most of the graphs. In most graphs, however, *entailed* contradiction, negated entailment, and negated contradiction heavily overlap. In many graphs, in fact, positive contraditiction and negated entailment have nearly identical distributions. This suggests that there is some confounding between negation and contradiction as the examples in these two groups are those that show exactly one of these characteristics. Examples with both of these characteristics, examples with negated contradiction, overlap with these two groups heavily but often show more extreme values. This indicates that in each of these neurons, which are the most promising for separability for negation, we find not separability for negation, but separability for the criteria of negation or contradiction. The conjunction of these two variables is somewhat expected given previous work that suggests negation is a heuristic used to predict contradiction in MNLI models.

Between the graphs, we also see several differences. In the binary model, the overlap between the distributions is greater, with the negated entailment and positive contradiction distributions being



(a) Distribution of 3 hidden neuron activations from the final layer of the full binary model



(b) Distribution of 3 hidden neuron activations from the second to last (pooling) layer of the full binary model

Figure 3: Simalar to the previous figure, these distributions of neuron activation are shown for hidden dimensions in the last and second to last layer of the full binary model. The different distributions in each plot are the same groups as above from Section 3.5

nearly identical. In the Full Binary model, this distribution is more spread out. The increased spread of the distributions shows the conjunction of negation and contradiction is not as strong in the full binary model. This makes sense as the neutral examples that are included in the contradiction class for this model which might lessen the dependence on negation as a heuristic. Finally, between the layers 1 and 2 in both graphs, the distribution show stark differences. There is a much more clear divide in layer 1 between entailment and contradiction. This makes sense as we would expect that in later layers, the representations of neurons better represent the distinction between the two predicted classes. These similarities and differences are consistent across not only the three examples for each case that we show here, but the 20 dimensions for each model that maximize our distance metric based on Wasserstein difference, which are those dimensions most promising for differential activation in the presence of negation.

Across all layers and models, we found no neuron that had differential activation for negation but not for contradiction. This suggests that there is no specific neuron that encodes only for negation because of conflation with other factors.

6.2 Causal Effects

Given that there was no neuron in these last three layers encoding negation (as shown through our correlational measures), there also was no neuron individually responsible for mediating the causal effect of negation on output prediction, a result which would require *both* that the neuron encode negation and have a causal effect. However, we investigated the indirect and direct effects of the neurons in the three layers in order to see whether there was a "contradiction" neuron in any of these layers. That is, we wanted to see whether the effect of negation had been localized by the last three layers such that a single neuron had a large effect on whether the output label was *contradiction*.

First, the direct and indirect effects of introducing negation were largely distributed across each of the layers we examined. For each combination of layer and model mentioned in Section 5, we calculated the direct and indirect effects of the values of each neuron in the layer and sorted the effects by magnitude across the hidden dimensions. As shown in Figure 1, across all experiments these lines are flat, indicating that the direct and indirect effects on output prediction probabilities when injecting change between the positive and negated

premise are nearly identical for each of these neurons and insignificant.

This result did not change when we altered either the model or the layer we examined. For each of these experiments, with a specific layer and model, we observed an indirect effect that was near zero and direct effect that was almost exactly the value of the probabilities of entailment for the negated examples. The invariability across neurons suggests that any information in these layers that may be encoded about whether label to be predicted is *entailment* or *not entailment* must be distributed. There is not a single neuron that significantly encodes this information or we would see a much higher indirect effect for at least one of these neurons. This does not rule out, however, the possibility that this information is encoded by a group of neurons, as a group might have an indirect effect higher than the sum of the component neurons.

7 Conclusion

Our results suggest that information about both negation and a bias towards a *contradiction* prediction is distributed by the time it reaches the last three layers of BERT. Additionally, information about these two features is entangled at this level of the model; the results from our correlational experiments showed conflation between these two factors that confirms the findings of previous works indicating negation as a heuristic for predicting *contradiction*. Together, this result suggests that if information about negation is encoded and used in a localized manner, it is happening before the last three layers of the model.

7.1 Future Work

Our research opens up some possible avenues of future research. The first is a deeper probing of our models. Our experimental method requires changing 768 dimensions for each of 6341 example sentences and running them through n layers of the model, where n is how far from the last layer the layer in question is. It is an open question at what point in the model negation is encoded and when it starts to be exploited by the model as a bias towards a prediction of *contradiction*.

We also note that our methodology is agnostic with respect to the linguistic feature being probed for, and similar techniques could be applied to other linguistic phenomena. The methodology is also model-agnostic and can be applied to models besides BERT.

7.2 Our code

Our code is publically available at `https://github.com/alexisjihyeross/cs287_causality_project`.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, abs/1608.04207.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*.

Yonatan Belinkov and James Glass. 2018. Analysis methods in neural language processing: A survey. *CoRR*, abs/1812.08951.

Samuel R. Bowman, Ellie Pavlick, Edouard Grave, Benjamin Van Durme, Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, and Berlin Chen. 2019. Looking for ELMo's friends: Sentence-level pretraining beyond language modeling.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *CoRR*, abs/1805.01070.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating compositionality in sentence embeddings. *CoRR*, abs/1802.04302.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem H. Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. *CoRR*, abs/1808.08079.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, R. Thomas McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel Bowman, and Ellie Pavlick. 2019. Probing what different nlp tasks teach machines about function word comprehension. In *Joint Conference on Lexical and Computational Semantics (StarSem)*.

R. Thomas McCoy and Tal Linzen. 2018. Non-entailed subsequences as a challenge for natural language inference. *CoRR*, abs/1811.12112.

Judea Pearl. 2013. Direct and indirect effects. *CoRR*, abs/1301.2300.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Eva Vanmassenhove. 2017. Investigating aspect in nmt and smt. *Computational Linguistics in the Netherlands Journal*, 27.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.