Probing Negation in Language Models

Shashwat Singh[1*] **Shashwat Goel**[1*] **Saujas Vaduguru**[2] **Ponnurangam Kumaraguru**[1]
IIIT Hyderabad[1] Carnegie Mellon University[2]
{shashwat.s, shashwat.goel}@research.iiit.ac.in
svadugur@cs.cmu.edu pk.guru@iiit.ac.in

Abstract

Prior work has shown that pretrained language models often make incorrect predictions for negated inputs. The reason for this behaviour has remained unclear. It has been argued that since language models (LMs) don't change their predictions about factual propositions under negation, they might not detect negation. We show encoder LMs do detect negation as their representations across layers reliably distinguish negated inputs from non-negated inputs, and when negation leads to contradictions. However, probing experiments show that these LMs indeed don't use negation when evaluating whether a factual statement is true, even when fine-tuned with the objective of changing outputs on negated sentences (Hosseini et al., 2021). We hypothesize about why pretrained LMs are inconsistent under negation: when the statement could refer to multiple ground entities with conflicting properties, negation may not entail a change in output. This means negation minimal pairs in different training samples can have the same completion in pretraining corpora. We argue pretraining may not provide enough signal to learn the distribution of ground referents a token could have, confusing the LM on how to handle negation.

1 Introduction

Pretrained language models (PLMs) are extensively being deployed in the real world. While they are becoming increasingly better at giving plausible human-acceptable outputs, it has been shown that they fail at even basic reasoning tasks (Huang and Chang, 2022). In this paper, we focus on a language model's (LM) ability to handle negation, which is important to follow instructions (Jang et al., 2023) and be consistent with facts (Burns et al., 2022). Negation can also change the inferences that can be drawn from any set of clauses (Hossain et al., 2020b). For example, "Tommy is a dog. Tommy is a human." is a contradiction but "Tommy is not a dog. Tommy is a human." is not one. More generally, it can change the classification of any input; one easy example is sentiment analysis, where "not good" is clearly a negative rating.

Models have been shown to not change their predictions sufficiently for negated inputs compared to their positive counterparts across NLP tasks like NLI (Naik et al., 2018), sentiment analysis (Zhu et al., 2014; Barnes et al., 2019), paraphrase identification (Kovatchev et al., 2019), machine translation (Hossain et al., 2020a), and question answering (Ribeiro et al., 2020; Sen and Saffari, 2020). For example, RoBERTa (Liu et al., 2019) predicts "NATO" for both P1: "Germany is a member of `[MASK]`." and P2: "Germany is not a member of `[MASK]`", where a valid output for P1 like "NATO" would be an invalid output for P2 (Kassner and Schütze, 2020). Scaling model size has been shown to degrade their performance on negated inputs across tasks like commonsense reasoning, question answering, and sentence completion (Jang et al., 2023; McKenzie et al., 2022).

Similar to prior work (Kassner and Schütze, 2020; Hosseini et al., 2021), we focus on simple expressions of negation based on "not" rather than "without", "except", etc. We wish to isolate and study how the model deals with negation once recognized rather than mechanisms of detecting negation or resolving its scope. Prior work has mainly gone in three directions. (1) demonstrating that pretrained LMs perform poorly under negation (Kassner and Schütze, 2020; McKenzie et al., 2022; Jang et al., 2023), (2) creating datasets to study the processing of negated sentences (McKenzie et al., 2022; Ravichander et al., 2022), and (3) modifying pretrained models for improved negation consistency (Hosseini et al., 2021). We seek to articulate better how pretrained models build representations for negated expressions. We show the presence of negation is reliably encoded even in the final layer

[*]These authors contributed equally to this work

of an LM. Using probing methodologies similar to Burns et al. (2022), we show that encoder LM representations linearly separate true and false factual statements indicating they can make factuality judgements. However, negation does not affect the LM's factuality judgement even though it should be changed. We show that even explicit finetuning to change outputs on negated factual propositions as done in the BERTNOT model (Hosseini et al., 2021) does not help incorporate negation into factuality judgements. This indicates that the objective of simply changing outputs in the presence of "not" is an insufficient characterization of what LMs need to "understand" negation.

Why then do LMs detect negation but still fail to incorporate the changes it causes? We outline a proposal by observing that changing outputs is not always required under negation. For example, "America" can be a valid completion present in the training dataset for both "The boy was born in `[MASK]`" and "The boy was not born in `[MASK]`". Intuitively we know that the two samples could have two different referents (say John and Fernandez respectively) by the same token "boy". Thus, negation does not necessarily imply changing outputs in statements involving tokens with referent ambiguity. However, we expect a change when the referent of the token is unambiguous. We argue that lack of explicit referent ambiguity signal in the pretraining corpora could lead to LMs improperly learning when to change outputs under negation.

Overall our main contributions are the following:

1. Through carefully controlled probing studies, we show PLMs can detect negation and reason about contradictions due to negation.
2. However, LMs fail to incorporate negation in their understanding of whether a factual proposition is true or false.
3. We draw a distinction between statements where negation changes the output and statements where the same output is also valid due to ambiguity in the referents. We argue this distinction could explain why LM outputs are often incorrect for negated inputs.

2 Related Work

2.1 Prior hypotheses

There have been some attempted hypotheses on why models fail at negation despite their progress and the simplicity of the task for humans (Jang et al., 2023). Kassner and Schütze (2020) conclude that "PLMs poorly distinguish positive and negative sentences." Similarly, Jang et al. (2023) claim "LMs could not find any distinction between the original and the negated prompts, treating them as identical instructions." In our work, we draw the distinction between detecting negation and correctly performing the operation it entails, showing that the problem lies in the latter instead.

McKenzie et al. (2022) present a different hypothesis. They claim that "not" is too small a perturbation to the input to cause the drastic changes making the most acceptable output the least acceptable one. The model may be incapable of deviating sufficiently from the plausible outputs provided by the rest of the tokens (the not-negated proposition) in the presence of "not." However, Hosseini et al. (2021) show that BERT can be finetuned to change outputs when "not" occurs in the input while retaining performance on non-negated sentences. Thus, models are capable of learning to change outputs in the presence of "not.". In Section 4 we explore why pretraining may not induce this behaviour.

2.2 Interpretability

We use probing methods throughout this work. Alain and Bengio (2016) show that models are incentivized to make useful information linearly separable, so it is often enough to use linear classifiers to check representations for encoding properties of interest. However, Pimentel et al. (2020) suggest that more complex probes like Multi-Layer Perceptrons (MLPs) are better for tighter estimates of what information exists in the model's representations. Voita and Titov (2020) show that if the probe can learn from small training sets, the efficacy is more likely to come from the representations being probed rather than the probe itself. Furthermore, Hewitt and Liang (2019) discuss the importance of control tasks to show probes are selective to the concepts studied and are not using spurious heuristics. Our experimental design in Section 3 incorporates these insights.

3 Experiments

In this section, we investigate whether representations of encoder-based LMs contain information about the presence of negation and the truth of factual statements. We also check whether negation is incorporated in the models' factual judgement.

3.1 General setup

We study two popular pretrained Transformer-based (Vaswani et al., 2017) encoder models: RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020). We use the Negated LAMA dataset (Kassner and Schütze, 2020), which contains a human-generated negated version of each datapoint in the LAMA dataset (F. Petroni and Riedel, 2019). The LAMA dataset contains masked factual propositions with their correct completions. Finally, we also compare BERTNOT (Hosseini et al., 2021), a state of the art method that claims to improve LM's understanding of negation. It finetunes BERT (Devlin et al.) on an objective that penalizes the model for predicting the same token as in the non-negated premise for the negated masked version of the premise. We use BERTNOT to investigate whether finetuning an LM to change outputs in the presence of "negation" improves the LM's understanding of "negation" as claimed in Hosseini et al. (2021). Most of the experiments deal with sentence representations obtained by averaging across token representations; this pooling strategy has been shown to perform decently on SentEval in (Reimers and Gurevych, 2019).

3.2 Representations distinguish the presence of negation

Kassner and Schütze (2020) hypothesized that LMs cannot detect the presence of "not." If this is true, the hidden layer representations, especially in the final layers, should be indistinguishable for negated and non-negated sentences. We use a linear probe to distinguish encoder LM representations on the negated LAMA dataset. If the probe can reliably distinguish representations of negated sentences, it means that representations of negated inputs are linearly separable from their positive minimal pairs.

Probe setup We take the mean-pooled representations of each layer and train a linear classifier on the binary classification task of whether the input contained "not." The training set for this experiment was constructed using representations created from negated ("Germany is not a member of NATO") and non-negated samples ("Germany is a member of NATO") from the dataset in Kassner and Schütze (2020). The classifier is trained on a balanced training set of 500 samples and tested on a held-out set of 2000 test samples. We use a small training set for the probe to ensure that the probe's complexity is low and the information it learns is from the LM representations with high likelihood as recommended in Voita and Titov (2020).

Control tasks We design a control task to ensure the specificity of our probe. Particularly, we replace the "not" inputs with a fixed random 3-character string and make predictions based on the LM representations of such inputs using our learned linear classifier. We find consistent results irrespective of the 3 characters chosen. We also run a control experiment with the negation token replaced with an adverbial phrase, specifically "actually." The probes trained for the task are used to obtain predictions on this modified control data. If the probe is specific to "not" and not using spurious heuristics like length, it should obtain low (near random, i.e. 50%) accuracy for both these control tasks.

Results The linear probes trained for all layers detect negation with above 95%, with results for the final layer reported in Table 1.

Model	Accuracy	$\text{Control}_{\text{3-char}}$	$\text{Control}_{\text{actually}}$
RoBERTa	0.985	0.50	0.53
DeBERTa	0.984	0.50	0.653
BERTNOT	0.98	0.54	0.81

Table 1: Linear probe accuracies for distinguishing negated and non-negated sentences from final layer LM representations. The control tasks have near-chance accuracies, indicating the probes are specific to "not." $\text{Control}_{\text{3-char}}$, $\text{Control}_{\text{actually}}$ refer to the control experiments with a random 3-character string, "actually" used to replace "not." respectively.

Conclusion *Encoder LM's detect the presence of "not" up to the final layer.*

3.3 Representations distinguish contradictory statements

Next, we want to see if the LM representations linearly separate a pair of sequences that contradict due to negation or agree despite negation.

Setup We take every proposition A and its negated counterpart A' from Negated LAMA and create 4 samples by concatenation. For example:

- AA: *"Germany is a member of NATO. Germany is a member of NATO."*
- AA': *"Germany is a member of NATO. Germany is not a member of NATO."*
- $A'A$: *"Germany is not a member of NATO. Germany is a member of NATO."*
- $A'A'$: *"Germany is not a member of NATO. Germany is not a member of NATO."*

We then formulate a binary classification problem that puts the contradictory pairs (AA', $A'A$) in one class and pairs that agree (AA, $A'A'$) in the other. This task can also be seen as performing the exclusive-or (XOR, $\oplus$) operation over the presence of "not" in the two given sentences. Since linear probes can only approximate $\oplus$ with upto 75% accuracy, if a linear probe can learn to reliably (much more than 75%) distinguish these classes, the LM represents negation-based contradiction and agreement in a linearly separable manner.

We train a linear classifier for each layer of the encoder to do the task based on the encodings in that layer. Results for the final layer and the embedding layer have been reported in Table 2.

We also use the same two control tasks as earlier to check for the specificity of the probe to contradiction and agreement based on "not" and rule out length heuristics. Specifically, we run inference for samples with "not" replaced by a fixed random 3-character string. Note that in the control task, the AA samples that form 25% of the entire dataset remain the same as the original task, and there is no instance of "not." This makes the most selective probe have an accuracy of 62.5%, i.e. 25% plus chance accuracy (37.5%) on the other 75% samples.

Results The linear classifier on the final layer representations of the encoder LMs gets high accuracies (90%+) indicating LMs can reason about contradictions. The random character control task accuracy is significantly lower whereas the "actually" control task achieves below 50% accuracy for both models. This shows that the classifier is specifically distinguishing negation-based contradiction and agreement between the concatenated clauses. The task requires the LMs knowledge as the accuracy from the embeddings is near chance (50%).

Conclusion *Negation-based contradiction and agreement due to "not" are linearly separable in the final layer of encoder LMs indicating that pretrained LMs can reason about contradictions induced by the presence of "not."*

3.4 Representations may not encode negation consistent factuality

Prior work has shown that language models capture some factual knowledge (F. Petroni and Riedel, 2019). However, Kassner and Schütze (2020) show that masked language models do not change the token they predict to fill the mask in a negated context. We wish to investigate if negation is used to inform an LM's understanding of whether a proposition is true at all. Burns et al. (2022) show that a model's evaluation of the truth value of propositions is encoded as a feature in their representations that can be extracted. We replicate their "upper bound" (highest accuracy) setup to try and extract negation-sensitive latent representations of factuality as we do have access to gold-standard factual statements.

Setup The Negated LAMA dataset (Kassner and Schütze, 2020) has masked facts and their negated versions. Each sample also has a correct completion for the non-negated sentence which can be used to create non-negated true facts and negated false facts. To create a dataset for non-negated false facts and negated true facts, we run the masked non-negated proposition through RoBERTa and sample from the probability distribution over tokens that fill the mask until the completion is not the same as the *correct* completion given in Negated LAMA to create a dataset of facts and false facts. [1] Note that this dataset is never re-generated and the other models being tested are made use the same dataset.

For example:

- **True fact non-negated**: *"Germany is a member of NATO."*
- **False fact non-negated**: *"Germany is a member of OPEC."*
- **False fact negated**: *"Germany is not a member of NATO."*
- **True fact negated**: *"Germany is not a member of OPEC."*

Using the above dataset, we train the following probes to predict whether a fact is true or not:

- Trained *exclusively on non-negated data*.
- Trained *exclusively on negated data*.
- Trained on both *negated and non-negated data*.

Note that since both the presence of negation (say a binary variable N) and the judgement of factuality (say a binary variable F) for the positive statement is available in the representations, the task could be solved as $N \oplus F$. Since XOR is a non-linear operation, the linear probe cannot directly learn the task this way. However, linear approximations of XOR can be made upto 75% ac-

[1] An examination of 400 randomly sampled datapoints reveals that 1% of the sentences labelled false facts are just simply not factual statements, i.e. they don't have a set truth value. No fact labelled false was actually true.

Representation	Case	Precision			Recall			F1-Score			Accuracy		
		RB	DB	BN	RB	DB	BN	RB	DB	BN	RB	DB	BN
Final layer	NC	0.94	0.97	0.90	0.94	0.99	0.89	0.94	0.98	0.90	0.94	0.98	0.90
	C	0.94	0.99	0.90	0.94	0.97	0.91	0.94	0.98	0.90			
Embedding layer	NC	0.55	0.50	0.54	0.51	0.51	0.53	0.53	0.50	0.53	0.55	0.50	0.54
	C	0.54	0.50	0.54	0.59	0.49	0.54	0.56	0.50	0.54			
3-char control	NC	0.65	0.55	0.77	0.88	0.99	0.82	0.74	0.71	0.80	0.70	0.60	0.79
	C	0.81	0.96	0.81	0.53	0.21	0.76	0.64	0.35	0.78			
"actually" control	NC	0.48	0.47	0.54	0.58	0.52	0.48	0.53	0.49	0.51	0.48	0.46	0.54
	C	0.48	0.46	0.53	0.39	0.40	0.59	0.43	0.43	0.56			

Table 2: Encoder LM representations can distinguish contradictory concatenated statements (AA', $A'A$; labeled C) from those that agree (AA, $A'A'$; labeled NC) with high reliability as shown in Rows 1–2. The last 6 rows show results on control tasks with the probe achieving low accuracies which indicates the probe is not powerful enough to learn the task on its own and the contradiction judgement is extracted from the LM representations. RB, DB, and BN refer to RoBERTa, DeBERTa, and BERTNOT respectively.

curacy, so the probe can achieve non-trivial (>50%) accuracy by approximating the XOR even if it cannot hone in on a negation-sensitive factuality dimension.

Results Results of the different probes can be seen in Table 3.

Rows 1–2: We see that a Linear probe trained with just 1000 training samples to classify non-negated facts achieves high accuracy (~75%) on held-out non-negated facts. This indicates factuality judgements are indeed encoded in LM representations consistent with (Burns et al., 2022). However, this probe performs very poorly on negated propositions.

Rows 3–4: We observe a similar but inverted trend for the Linear probe trained only on negated data as it performs almost equally well on negated inputs and poorly on non-negated inputs. These results indicate that the factuality judgement the probes hone in on does not change the truth value when negated.

Rows 5–6: The probes trained on exclusively positive or negative data may just not be honing on negation-sensitive factuality dimensions. To check whether negation-sensitive factuality judgements exist in LM representations we consider the probe trained on both negated and non-negated propositions. We observe only 64% accuracy from the linear probe, which can be achieved by approximating an XOR (upto 75% accuracy) of separately encoded negation (almost 100% accuracy) and positive factuality (almost 80% accuracy) information as discussed earlier.

Rows 7–10: A more complex MLP cannot achieve much higher accuracy than our linear probes trained on just 1000 samples. This shows that factuality judgements are encoded in a linearly separable manner in LM representations.

Rows 11–12: As discussed earlier, a non-linear probe can learn the $N \oplus F$ using the separate negation and factuality judgement information in the representations. As expected, the MLP achieves high accuracies, significantly outperforming the linear probes in Rows 5, 6.

Note that despite finetuning for changing outputs on negated factual propositions, the probes on BERTNOT obtain similar results to vanilla pretrained RoBERTa and DeBERTa. This shows changing outputs in the presence of "not" is not a sufficient objective to learn that "not" changes the truth value of a proposition which is necessary to truly "understand" negation.

Conclusion *The results suggest that LM factuality judgements are not negation sensitive as they should be, even though both the presence of negation and factuality judgements are encoded in the representations separately.*

4 Characterizing Negation Consistency

Consider the sentence "The boy was born in `[MASK]`" and its negated counterpart "The boy was not born in `[MASK]`". If these two sentences are provided independently, "America" forms a plausible high probability completion to both. If they occur in different samples of the training data, they could act as evidence for the LM that it is okay to produce the same output upon negation. When then do we expect different outputs for a negated input?

#	Probe	# samples		RoBERTa		DeBERTa		BERTNOT	
		+	−	+	−	+	−	+	−
1.	Linear	1000	0	0.78	0.26	0.76	0.26	0.75	0.189
2.	Linear	7000	0	0.82	0.23	0.81	0.27	0.81	0.21
3.	Linear	0	1000	0.26	0.79	0.28	0.76	0.29	0.76
4.	Linear	0	7000	0.21	0.81	0.21	0.81	0.23	0.81
5.	Linear	1000	1000	0.64	0.64	0.61	0.62	0.61	0.58
6.	Linear	7000	7000	0.68	0.66	0.67	0.64	0.63	0.61
7.	MLP	1000	0	0.79	0.25	0.79	0.22	0.78	0.24
8.	MLP	7000	0	0.85	0.19	0.84	0.19	0.85	0.19
9.	MLP	0	1000	0.29	0.76	0.24	0.77	0.27	0.78
10.	MLP	0	7000	0.19	0.86	0.18	0.85	0.20	0.85
11.	MLP	1000	1000	0.75	0.78	0.76	0.77	0.75	0.76
12.	MLP	7000	7000	0.85	0.85	0.83	0.83	0.81	0.83

Table 3: Factual judgement extraction for probes trained on positive negative or both classes. + represents the number of training samples that were not negated & − represents negated training samples. We report accuracies on + and − held-out samples.

When referring to a single entity, we expect different outputs for negated inputs. Consider the case of sentences that state well-known facts (which we refer to as *factuality sentences*). Typically, factuality sentences refer to famous people, objects or events. In these cases, the ambiguity of reference can be resolved without additional context. The probability that the token "Einstein" refers to the Nobel Prize winner is very high, and other people named Einstein have a negligible effect on the output probability distribution. So it is reasonable to expect the output to change, or different completions to "Einstein likes to eat `[MASK]`" and "Einstein does not like to eat `[MASK]`".

On the other hand, consider a scenario like "The boy was born in America. The boy was not born in `[MASK]`". The context here narrows the set of possible referents of "The boy" to the set of boys that were born in America, say B_{America}. We expect "America" to be a very low probability completion for `[MASK]` as for each $b \in B_{\text{America}}$, we expect "b was not born in `[MASK]`" to have "America" as a very low probability completion. Thus it is again reasonable to expect different completions to `[MASK]` in the negative sentence. We refer to these cases as *in-context scenarios*.

We thus draw a distinction between:

- **Type 1** low ambiguity in referent. Situations like factual and in-context propositions where we can expect changes in outputs for negation.
- **Type 2** situations where the referent has high ambiguity. The same output under negation is possible in a subset of these propositions.

Note that while nouns like the subject are the main source of ambiguity and type determination, they do not necessarily determine whether negation consistency entails a change in output. This is because not all Type 2 propositions require a change when negated. The relation can induce structure such that subjects with high ambiguities also require changed outputs for negation consistency. For example, in "The diesel car does not consume `[MASK]`", "consume" ensures that all diesel cars have the same completion (i.e. "diesel") for the non-negated proposition, requiring a changed output under negation despite being a Type 2 situation. Next, we use a toy example to illustrate how the Type 1 and Type 2 distinction can arise in the case of negation for masked language modelling.

4.1 Masked language modelling illustration

Masked language modelling involves learning to fill in a masked token M given the context of surrounding tokens C, where M and C are random variables taking values of a single token and a sequence of tokens respectively. Further, we separate the negation information from the context using the random variable N that takes the value 1 when the input is negated, and 0 otherwise. We still use C to refer to the remaining context tokens. Let us consider a situation where we are modelling statements made about two worlds w_1 and w_2, where different facts are true. If we know which world is being referred to, we could correctly determine the completion m for a statement C that has a token masked out. However, the LM does not have access

to referents and learns a distribution that is unaware of the underlying world about which the statements are made. We can view this as marginalizing over different worlds (Xie et al., 2022).

$$P(M|C,N) = \sum_{w} P(M|C,N,w)P(w|C)$$

where $P(w|C) = P(w|C,N)$ is the probability that the statement is made about the world w given the context C, N assuming the world chosen is independent of the presence of negation and only determined by the form C. In particular, when C resolves to a single entity with high probability ($P(w|C)$ is high for one of the worlds w), we call the statement Type 1, and otherwise Type 2. Whether the output should be changed for Type 2 statements depends on the joint distribution $P(M|C,N)$. While the Type 1 and Type 2 distinction is arbitrary, it is helpful for intuition.

Table 4 shows a distribution $P(M|C,N)$ where C either resolves to world w_1 with high probability (Type 1), or both worlds w_1, w_2 are equally likely (Type 2). It shows how the top prediction may not change upon negation, even retaining the same probability. Let C (example: "The boy likes to eat `[MASK]`") have two possible referents, say c_1 and c_2 if two worlds w_1 and w_2 respectively. Let there be three values for `[MASK]`: m_1, m_2, m_3 (for example, "fruits", "vegetables" and "meat") which have a non-zero probability. The probabilities in Table 4 follow the probability sum constraint $\sum_{m \epsilon support(M)} P(M = m|C,N) = 1$. Further, they follow the constraint of the top prediction changing for a single referent, i.e. from m_3 to m_1 for c_1 and from m_1 to m_2 for c_2. Given just the ambiguous token C which could refer to both c_1, c_2 with equal probability, the top prediction can be the same, i.e. m_1 with equal probability.

We do not have access to the ground truth referents that a model was trained on, so we cannot compute how a model estimates $P(M|C,N)$ like in the illustration above. However, viewing negation consistency in terms of cases where the referent is more or less ambiguous (as in Type 2 or Type 1) may suggest a way to probe for what a model learns about negation consistency.

	$N = 0$			$N = 1$		
	m_1	m_2	m_3	m_1	m_2	m_3
w_1	0.2	0.3	**0.5**	**0.6**	0.1	0.3
w_2	**0.6**	0.2	0.2	0.2	**0.5**	0.3
Type 1 C	0.24	0.29	**0.47**	**0.56**	0.14	0.3
Type 2 C	**0.4**	0.25	0.35	**0.4**	0.3	0.3

Table 4: Example for how negation consistency may not require changing outputs in case of referent ambiguity. In Type 1 situations, it is clear which world the utterance is referring to (here w_1). In this case the probabilities for the two worlds are mixed with $P(w_1|C) = 0.9$ and $P(w_2|C) = 0.1$. In Type 2 situations, it is not clear which of the two worlds the utterance is referring to. The probabilities for the two worlds are mixed with $P(w_1|C) = P(w_2|C) = 0.5$. The top predictions (highest probability) for each situation are in bold. Upon negation ($N = 0 \rightarrow N = 1$), the top prediction changes for Type 1 situations (here $m_3 \rightarrow m_1$) but may not change for Type 2 situations (here m_1).

5 Discussion

5.1 Negation and Factuality

We have shown that LM's factuality judgements are not sensitive to negation (Section 3.4), which can be a major shortcoming in their understanding of "truth," or their own "truthfulness."

One may then wonder how these LMs can still change their predictions on some factual propositions when negated, with BERTNOT producing a change for most samples as it is explicitly fine-tuned to do so. We have shown that final layer LM representations encode both the presence of negation (Section 3.2) and factuality judgements (Section 3.4). The LM head can learn to use the separately available negation information to change the output in the presence of negation. Note that just changing the output is simple, a sufficient amount of random noise added to the representations in the presence of negation would also change the output. It is then not surprising that the model changes its output on *some* factual propositions when negated despite not incorporating negation in judging whether a factual proposition is true. Our results in Section 3.4 show that simply achieving changed output in the presence of negation is a weak objective for LMs to truly understand negation. We should also evaluate other constraints negation poses, such as changing the binary truth value of propositions.

5.2 Referent Ambiguity in Pretraining

As humans, if different people talk to us about a "box," we initially assume they are referring to different boxes and thus may not be confused if the boxes have contradicting properties. However, for an LM, if a "box" has contradictory descriptions

in different training samples ("The box is big, The box is not big"), it may not understand the boxes being referred to are different. This could confuse the model regarding the meaning of "not."

During training, independent samples may contain a Type 2 proposition and its negated version with the same completion. It is possible that pretraining does not have sufficient signal to distinguish Type 1 and Type 2 situations as learning the ambiguity in referents may require pragmatics and grounding. In this case, negation minimal pairs not requiring a change in completions can act as noise that hampers the model's ability to be negation consistent in Type 1 situations. This may explain the poor performance of LMs on negated inputs.

5.3 Limitations and Future Work

We perform our analysis on only Type 1 data (factual statements) due to a lack of corpora that distinguish Type 1 and Type 2 situations. Creating such datasets is an important direction for future work, which could also help extend our analysis of how LMs behave on Type 2 datasets. We have not yet studied whether LMs can model referent ambiguity. Causal interventions (Geiger et al., 2020) to show whether LMs use referent ambiguity to learn negation consistency is an exciting direction for future work. Future work could also explore more diverse negation datasets (Jiménez-Zafra et al., 2020) like CondaQA (Ravichander et al., 2022).

6 Conclusion

By probing their representations, we show that language models can reliably detect negation and reason about negation-based contradictions. However, we show that the presence of negation does not inform the encoding of the truth value of a statement in the LMs representations. Simply finetuning the LM to change its output in the presence of "not" (Hosseini et al., 2021) is insufficient for the LM to understand negation, specifically that "not" changes the truth value of statements. We propose learning to be consistent under negations may be hard as negation doesn't always necessitate a different output from the positive proposition when there could be multiple possible referents. We hypothesize that the lack of sufficient explicit signal in pretraining corpora about possible referents in the real world may confuse LMs about when negation entails changing outputs. Overall, we hope our characterization of negation consistency in the context of LMs guides research that improves how LMs reason about negated sentences.

7 Acknowledgement

Shashwat Singh was funded by the Centre for AI safety through their Research Stipend program.

We would like to thank Abhinav Menon, Ameya Prabhu, Anmol Goel, Arvindh A, Hitkul Jangid, Prashant Kodali, Shivansh Subramanian, and Vamshi Krishna for their valuable feedback.

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment analysis is not solved! assessing and probing sentiment classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

A. H. Miller P. Lewis A. Bakhtin Y. Wu F. Petroni, T. Rocktäschel and S. Riedel. 2019. Language models as knowledge bases? In *In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019*.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Md Mosharaf Hossain, Antonios Anastasopoulos, Eduardo Blanco, and Alexis Palmer. 2020a. It's not a non-issue: Negation as a source of error in machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020b. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2023. Can large language models truly understand prompts? a case study with negated prompts. In *Transfer Learning for Natural Language Processing Workshop*. PMLR.

Salud María Jiménez-Zafra, Roser Morante, María Teresa Martín-Valdivia, and L. Alfonso Ureña-López. 2020. Corpora annotated with negation: An overview. *Computational Linguistics*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Venelin Kovatchev, M. Antonia Marti, Maria Salamo, and Javier Beltran. 2019. A qualitative evaluation framework for paraphrase identification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ian McKenzie, Alexander Lyzhov, Alicia Parrish, Ameya Prabhu, Aaron Mueller, Najoung Kim, Sam Bowman, and Ethan Perez. 2022. The inverse scaling prize.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. *arXiv preprint arXiv:2003.12298*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. *ICLR*.

Xiaodan Zhu, Hongyu Guo, Saif Mohammad, and Svetlana Kiritchenko. 2014. An empirical study on the effect of negation words on sentiment. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Abhilasha Ravichander, Matt Gardner, and Ana Marasović. 2022. Condaqa: A contrastive reading comprehension dataset for reasoning about negation.