Not All Negation Cues Are Equal: Improving Negation Understanding by Leveraging All Cue Types

Anonymous ACL submission

Abstract

This paper investigates whether pre-training with affixal, single-word, and multi-word negation cues results in improved negation understanding. We work with encoder-only LMs as well as smaller LLMs. Experimental results show that pre-training with affixal negations is most beneficial across all negation types at inference time, and benefits persist regardless of cue frequency. Our pre-training is model- and task-agnostic; we show improved results across all models and five benchmarks for question answering, information retrieval and inference.

1 Introduction

Negation is a complex linguistic phenomenon that "relates an expression *e* to another expression with a meaning that is in some way opposed to the meaning of *e*" (Horn and Wansing, 2025). Negation goes beyond mere opposites and antonyms; nuance is often required to interpret the underlying affirmative meaning negations often imply (Huddleston and Pullum, 2002). For example, "John wasn't famous prior to his TV appearance" implies "John became famous because of his appearance on TV," which is key to answering "How did John become famous?"

Text with negations is challenging for state-of-the-art models, including LLMs. Ettinger (2020) showed that BERT disregards negation cues when filling placeholders (e.g., standard language modeling), and more recent work has shown that LLMs underperform in many tasks when negation understanding is required, including question answering (Ravichander et al., 2022) and information retrieval (Weller et al., 2024).

Prior model-agnostic proposals to improve negation understanding either (a) generate affirmative interpretations similar to the example above, thereby doubling the input size and requiring additional compute (Hossain and Blanco, 2022) or (b) further pre-train encoder-only LMs with a handful of selected negation cues (Hosseini et al., 2021). Inspired by the latter, in this paper we study the benefits of pre-training with all negation cues (affixal, single-word, multi-word) on both encoder-only LMs as well as (smaller) LLMs.

Samples from NegCue:

S_1^1: The room was extremely hot and crowded.
S_2^1: Everyone felt [uncomfortable]$^{\text{affixal cue}}$ inside.

S_1^2: The big gate was locked from the outside.
S_2^2: She did [not]$^{\text{single-word cue}}$ manage to open it.

S_1^3: The public garden is now private property.
S_2^3: It is [no longer]$^{\text{multi-word cue}}$ open to the public.

S_1^4: The weather is sunny and quite warm.
S_2^4: People enjoy their afternoon outside.$^{\text{no negation}}$

Next Sentence Polarity Prediction (NSPP)

Input:	S_1^1	S_1^2	S_1^3	S_1^4
Output:	Yes	Yes	Yes	No

Figure 1: Four examples from NegCue, the corpus we create for Next Sentence Polarity Prediction, our self-supervised task to improve negation understanding. Given a sentence (S_1^i), the task is to predict whether the next sentence (S_2^i) contains negation. While previous work is limited to a handful of cues (e.g., *not*, *never*), we work with all negation types (affixal, single- and multi-word) and over 200 negation cues.

Specifically, we pre-train for Next Sentence Polarity Prediction, the task of predicting whether a sentence from a document is followed by a sentence containing negation. To this end, we create NegCue, a corpus of over 1.6M instances with an equal number of affixal, single-word, and multi-word cues. Figure 1 illustrates the negation cue types, the NegCue corpus, and the NSPP task. In this paper, we work with BERT, RoBERTa, and smaller versions of Llama and Qwen, but neither NSPP nor NegCue are model-dependent.

The main contributions of this paper are:

1. NegCue, a large dataset for Next Sentence Polarity Prediction (1.6M instances and 214

unique cues). It considers all negation cues: affixal, single-word, and multi-word.

2. Experiments with encoder-only LMs (BERT and RoBERTa) and (smaller) LLMs (Llama and Qwen) show that NSPP pre-training with NegCue benefits across 5 benchmarks.
3. Analysis showing which negation cues are most beneficial to pre-train with and which are most beneficial during inference.

2 Related Work

Negation is ubiquitous in English: over 28% of sentences from books and 27% from oral and written conversations contain negation (Hossain et al., 2020b). The same is true in over 13% of sentences from biomedical papers (Szarvas et al., 2008). Beyond frequent single-word cues (e.g., *not*, *never*), negation can be expressed in several ways, including affixal cues (e.g., careless, unadjusted), infrequent single-word cues (e.g., forbid, deny), and multi-word cues (e.g., no longer, be deprived of).

Despite being frequent, there is plenty of evidence that negation is not quite acquired by models during standard pre-training (language modeling, next sentence prediction, etc.). For example, previous work has shown that BERT mostly disregards negation (Ettinger, 2020), NLI is more challenging when negation is present in the input with both encoder-only LMs (Hossain et al., 2020b) and LLMs (Geiger et al., 2020; She et al., 2023). Similarly, machine translation is more challenging if there is a negation in the source sentence (Hossain et al., 2020a), as are other tasks we work with in this paper: question answering (Ravichander et al., 2022; Zhang et al., 2023) and information retrieval (Weller et al., 2024).

Previous work on making models better at processing negation—regardless of the benchmarks used for evaluation—targets a handful of negation cues: *not*, *n't*, and *never*. They can be broadly divided into two categories. The first category includes approaches to generate underlying affirmative interpretations from negated statements and concatenating these interpretations to the original negated sentences (Hossain and Blanco, 2022; Rezaei and Blanco, 2024). For example, given "Extinct volcanoes have not erupted in recent history," they first generate the affirmative interpretation "Extinct volcanoes erupted a long time ago" and then feed to the model both sentences (as opposed to only the original statement with negation). Note that these approaches increase compute at inference time. The second category explores self-supervised tasks and pre-training with encoder-only LMs, including NSPP (Hosseini et al., 2021; Rezaei and Blanco, 2025). The latter requires additional compute to further pre-train a model for negation, but has no overhead at inference time.

In this paper, we build upon the second category. Specifically, we explore for the first time the benefits of working with all negation cue types (affixal, single-, and multi-word). Further, we demonstrate that pre-training for NSPP with all negation cue types benefits encoder-only LMs as well as (smaller) LLMs with 5 benchmarks for three core tasks: question answering, information retrieval, and natural language inference. The benchmarks we work with were built to assess the ability of models to understand negation.

3 Improving Negation Understanding

We improve negation understanding by further pre-training off-the-shelf models with Next Sentence Polarity Prediction (NSPP), a model-agnostic, self-supervised task. We first define the NSPP task, and then present NegCue, the corpus used to pre-train. NegCue contains instances of 3 negation types (1/3 each: affixal, single- and multi-word) and over 200 negation cues.

3.1 Next Sentence Polarity Prediction (NSPP)

We use the NSPP task as our pre-training task. Given a sentence from some document, the task consists of predicting whether the next sentence contains negation (i.e., the polarity of the next sentence). As illustrated in Figure 1, the model outputs a binary label indicating whether the next sentence has a negation cue.

3.2 NegCue: A Large Corpus for NSPP

Existing negation-related corpora and tasks often rely on a very limited set of high-frequency negation cues (Section 2). For example, Rezaei and Blanco (2025) use NSPP to make models more robust against negation, but they only consider three cues: *not*, *n't*, and *never*. Negation, however, is expressed in many complex and varied ways in language (Horn, 1989), appearing in affixal (e.g., *impossible*, *careless*), single-word (e.g., *without*, *few*), and multi-word (e.g., *with the exception of*, *no longer*) forms. To examine how these cue types influence negation understanding, we

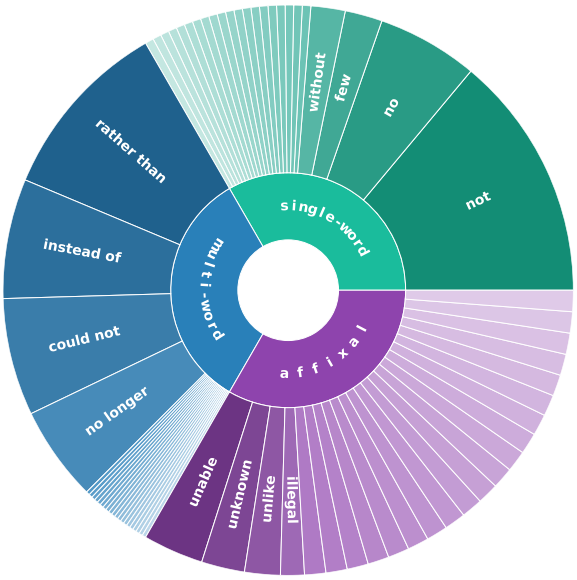

Figure 2: Frequency distribution of the negation cues in the NegCue training splits. Recall that each cue type (inner ring) accounts for 1/3 of NegCue. The four most frequent cues per type (outer ring) account for 88% and 71% of multi-word and single-word cues, but only 28% of affixal cues. The unlabeled segments illustrate the frequency of the remaining cues.

construct NegCue, a corpus including consecutive sentences (S_1, S_2) from genuine documents, where S_2 may or may not contain an affixal, single- or multi-word negation cue.

Selecting Text Sources and Negation Cues. The construction of NegCue relies on two key decisions: which texts to consider and the criteria to identify negation cues. Regarding text sources, we chose to work with a snapshot of the 2023 English Wikipedia (Wikimedia-Foundation, 2023), as it contains polished sentences. Further, previous work has estimated that ≈9% of sentences in Wikipedia contain negation (Hossain et al., 2020b).

While understanding how negation changes meaning in natural language poses substantial challenges, identifying negation cues in text is much easier. Indeed, traditional machine learning with lexical features obtains 91 F1 (Lapponi et al., 2012). Inspired by this insight and previous work relying on word matching (Morante and Daelemans, 2012), we identify negation cues via keyword matching. Specifically, we use the negation cues in CondaQA (Ravichander et al., 2022) and the affixal cues corpus proposed by van Son et al. (2016), following the definition of Joshi (2012). Combining these corpora, we obtain 214 unique negation cues.

Categorizing Negation Types. We must know the types of each negation cue in NegCue in order to analyze the impact of each type in improving negation understanding with NSPP. CondaQA contains affixal, single-, and multi-word negation cues, but it does not explicitly indicate cue types. Categorizing cues into types appears straightforward, but in practice it is non-trivial (Morante et al., 2011; van Son et al., 2016). For example, some single-word negations include a prefix that sometimes—but not always—indicates an affixal negation. Consider *absence*, a single-word negation cue. It starts with *a-*, a prefix that often signals affixal cues such as *atypical*. To make things more complicated, many English words begin with *a-* but are not negation cues (e.g., *attention*).

We devise a simple deterministic approach to categorize the negation cues in NegCue. First, cues are categorized as affixal if they are in the corpus created by van Son et al. (2016). Second, the remaining cues are categorized as single- or multi-word based on their token counts. Lastly, we adopt a strategy prioritizing longest matches. This allows us to identify *no longer* as a multi-word negation cue in "John no longer enjoys city life" as opposed to the single-word cue *no*.

Obtaining Sentence Pairs and Instances for NSPP. In order to obtain (S_1, S_2) sentence pairs, we iterate over each pair of consecutive sentences in each Wikipedia article. We select those such that S_2 (a) contains exactly one negation cue from our list and (b) is not a question. Then, we create a positive instance (S_1, yes_neg) for NSPP. In addition, we create a negative instance (S_1, no_neg) for NSPP by randomly selecting from the same article a sentence followed by a sentence without negation. If it is impossible to extract as many negative as positive instances from an article, we collect the necessary negative instances from other articles.

The final version of NegCue contains the same amount of instances derived from each cue type (affixal, single-, and multi-word). The frequency of each negation cue, however, follows the natural distribution in Wikipedia (i.e., more frequent cues in Wikipedia appear more often in NegCue).

Negation Cues in NegCue. Figure 2 and Table 1 analyze the negation cues in NegCue (more specifically, the cues present in S_2 in the positive instances). The three cue types have the same count (33%), but we observe substantially different cue frequencies per type. There are only 11 multi-word

	Affixal	Single	Multi
Avg. sent. length	22.3	22.2	22.9
# negation cues	167	36	11
most frequent	unlike	not	rather than
frequency (%)	10.3	41.8	31.1
least frequent	uncarved	forbid	be deprived of
frequency (#)	12	90	126

Table 1: Descriptive statistics of the NegCue training splits. There are more affixal cues, but most are infrequent. On the other hand, there are fewer single- and multi-word cues, but they are relatively frequent (*not*: 41.8%, *rather than*: 31.1%). Sentence length (# tokens) is roughly the same across cue types.

cues, and the top-4 most frequent account for 88%. On the other hand, there are 167 affixal cues, and the most frequent (*unlike*) accounts for 10.3%. Single-word cues are relatively infrequent except *not* (41.8%); similar to the affixal cues, there is a long-tail of infrequent single-word cues.

It is worth noting that sentence length is remarkably constant (22.2–22.9 tokens) regardless of which cue type is present (Table 1). Overall, NegCue contains in the train split over 1,673,226 instances (50/50 positive/negative), with 557,742 belonging to each negation cue type. The validation and test splits contain 92,952 and 92,964 instances (also balanced across labels and cue types).

4 Evaluation Benchmarks

We experiment with a battery of benchmarks purposely designed for negation understanding. The benchmarks cover question answering, information retrieval, and natural language inference.

CondaQA (Ravichander et al., 2022) is a large, contrastive question answering benchmark designed to check the ability to reason about negation in context. Each instance consists of a question paired with a passage containing a negated statement relevant to the question. Answers take one of several forms: *Yes*, *No*, *Don't know*, or a span from the passage. Each question is paired with the original passage (with negation) from Wikipedia and three modified versions: a (meaning-preserving) paraphrase, a version modifying the scope of the negation, and a version removing the negation. Note that the same question often has different answers depending on which passage is given in the input. In addition to *accuracy*, the authors of CondaQA propose as an evaluation metric *group consistency*: whether the answers from all (or a subset) of passages (original and modified) are correct. It contains 7.240 instances in the test split, and 5,832 and 1,110 in the train and validation splits.

NeQA (Zhang et al., 2023) is a multiple-choice commonsense question answering benchmark that involves negation. Each instance consists of a negated question paired with two answer choices: one correct and one incorrect. NeQA exhibits inverse scaling behavior (Wei et al., 2023; McKenzie et al., 2024), meaning that performance does not consistently improve as model size increases. NeQA is relatively small (train: 374, validation: 54, test: 100) but was built specifically to check negation understanding in a simple setting—and shows that LLMs underperform.

NevIR (Weller et al., 2024) is a benchmark that evaluates negation understanding in information retrieval, specifically, ranking documents given a query. It contains contrastive query–document pairs where the only difference in the queries is whether they contain negation (e.g., What should I do / avoid if my friend has a seizure?). Document rankings change depending on whether the query contains negation. The task requires models to correctly rank the documents for *both* queries, and the evaluation metric is pairwise accuracy. NevIR contains 1,283 instances in the test split, and 948 and 225 for training and validation.

NMoNLI and ScoNe-NLI are negation-focused lexical entailment benchmarks for natural language inference (NLI). Both are binary classification tasks and consider two labels: entailment and neutral (no contradictions, unlike other NLI benchmarks). NMoNLI extends SNLI (Bowman et al., 2015) by introducing negation into each instance (Geiger et al., 2020). The negation is restricted to scope over the substituted token span, thus it affects the entailment relation. ScoNe-NLI (She et al., 2023) further extends NMoNLI by constructing contrast sets containing six instances. Each contrast set contains zero, one, or two negations and each negation scope may or may not affect the NLI label. The numbers of training and test instances for NMoNLI and ScoNe-NLI are reported in Table 6 in the appendix; they were created primarily as benchmarks to evaluate LLMs on the NLI task when complex negations are present in the premise or hypothesis.

5 Experiments

We experiment with encoder-only LMs (BERT-large and RoBERTa-large) and (smaller)

LLMs (LLaMA3.2-1B and Qwen2-0.5B), and evaluate on the five negation-specific benchmarks (Section 4). Specifically, we evaluate off-the-shelf models as well as improved versions further pre-trained with NSPP and NegCue. Section 5.1 describes the further pre-training setup, and Section 5.2 details the experimental results.

5.1 NSPP Pre-training with NegCue

We further pre-train off-the-shelf versions of the LMs and LLMs for Next Sentence Polarity Prediction with NegCue. Specifically, we pre-train for NSPP with the full NegCue corpus as well as subsets consisting of positive instances belonging to each negation cue type (affixal, single- and multi-word). This setup allows us to examine how pre-training with different negation cue types affects negation understanding.

Encoder-only LMs are pre-trained under the same settings. We use a learning rate of 1e-6, batch size of 512, and early stopping. Smaller LLMs are pre-trained using QLoRA with 4-bit quantization (Dettmers et al., 2023). All NegCue instances are converted into zero-shot instruction-style prompts. We use a learning rate of 2e-4 for LLaMA3.2-1B and 1e-5 for Qwen2-0.5B. The specific prompt and additional implementation details are provided in Appendix B.

5.2 Evaluation on Downstream Tasks

We evaluate all models (off-the-shelf and further pretrained with NSPP and NegCue) in downstream tasks under two settings. Encoder-only LMs are evaluated after supervised fine-tuning with relevant data, and (smaller) LLMs are evaluated with zero-shot prompting.

Evaluating on CondaQA. For CondaQA, we report accuracy and group consistency (Section 4) with the full test split. We also partition the test instances into three subsets depending on the negation cue type (affixal, single-word, and multi-word) to investigate which cue types benefit the most (at inference time) depending on which cue types are used for pre-training. Additionally, we partition the test instances into four quartiles based on negation cue frequency to investigate whether the benefits of pre-training for NSPP with NegCue apply to all negations regardless of their frequency.

For LMs, we fine-tune both off-the-shelf and further pre-trained models on the CondaQA training split. Hyperparameters are selected via grid search over the same search space for all models, with the best configuration chosen based on validation performance. For LLMs, we evaluate on the CondaQA test set using zero-shot in-context learning (Brown et al., 2020), where all instances are presented as instruction-style prompts. Additional details are provided in Appendix C.1.

Evaluating on NeQA. The NeQA benchmark is designed for evaluating LLMs and provides a test set without an explicit training-validation split. To enable fine-tuning for LMs, we construct training and validation splits based on the dataset metadata. Specifically, each original multiple-choice question is transformed into two instances by pairing the question with each answer option and assigning a binary label indicating whether the option is correct. For LLMs, we evaluate the performance on the test set using zero-shot prompting; see Appendix C.2 for the exact prompt.

Evaluating on NevIR. NevIR is designed for information retrieval models. To adapt it for LMs, we fine-tune models using the Sentence-Transformers framework (Reimers and Gurevych, 2019) on the NevIR training set to enable document ranking, and evaluate performance on the test set. LLMs are evaluated using zero-shot prompting; see Appendix C.3 for the exact prompt. Evaluation is performed using pairwise accuracy, which measures whether a model correctly ranks documents for both the original and negated queries.

Evaluating on NLI Benchmarks. Fine-tuning LMs on the training sets of NMoNLI and ScoNe-NLI leads to near-perfect performance, with test accuracies above 99% across subsets of the test splits (Table 13). We therefore fine-tune LMs on the well-known SNLI training set (Bowman et al., 2015) and evaluate them on the NMoNLI and ScoNe-NLI test splits. For LLM evaluation, we adopt a zero-shot prompt. Additional implementation details are provided in Appendix C.4.

6 Results and Discussion

We first discuss the results with the largest benchmark, CondaQA, and then with the remaining benchmarks. Pre-training both encoder-only LMs and (smaller) LLMs for NSPP with NegCue consistently yields better results across all benchmarks.

6.1 Results on CondaQA

We analyze results on CondaQA from three complementary perspectives.

Overall Accuracy and Group Consistency.

	#Pars.	Accuracy				Group Consistency			
		All (Δ%)	Affixal	Single	Multi	All	Par.	Sco.	Aff.
Fine-tuned encoder-only LMs									
Previous Work									
BERT-large [1]	340M	46.3	n/a	n/a	n/a	2.2	44.2	14.8	12.4
RoBERTa-large [1]	355M	54.1	n/a	n/a	n/a	13.6	51.6	26.5	27.2
+ affirmative interpretations [2]		67.1	n/a	n/a	n/a	31.4	61.9	43.8	50.7
+ NSPP (not, n't, never) [3]		67.3	n/a	n/a	n/a	33.5	64.6	46.3	50.2
UnifiedQA-v2-large [1]	770M	66.7	n/a	n/a	n/a	30.2	64.0	43.7	46.5
Our results									
BERT-large (off-the-shelf)	340M	46.8	47.0	46.9	44.6	2.2	44.9	16.0	11.3
further pre-trained, NSPP with									
affixal cues		**50.1** (+7.1)*	**49.3**	**51.3**	47.8	**6.7**	46.0	**20.1**	**19.8**
single-word cues		48.3 (+3.3)*	47.3	49.7	46.4	3.7	**48.3**	17.9	13.1
multi-word cues		48.9 (+4.4)*	48.7	49.7	43.9	4.6	47.5	18.6	15.3
All cues		48.6 (+3.8)*	47.9	49.0	**51.5**	4.5	45.9	17.8	15.9
RoBERTa-large (off-the-shelf)	355M	54.1	51.3	57.2	52.9	12.7	51.4	26.8	24.9
further pre-trained with:									
affixal cues		**68.0** (+25.8)*	**65.5**	**70.7**	**67.7**	**33.4**	**66.0**	**46.0**	**50.5**
single-word cues		65.2 (+20.5)*	63.1	67.5	64.9	30.7	63.6	43.5	47.4
multi-word cues		65.4 (+21.0)*	63.1	68.2	63.5	29.1	60.6	41.6	46.6
All cues		65.3 (+20.8)*	63.9	66.8	65.6	29.1	62.3	41.9	46.4
LLMs, zero-shot									
Previous Work [1]									
UnifiedQA-v2-11B	11B	73.1	n/a	n/a	n/a	40.0	75.5	53.7	54.1
GPT-3	175B	43.7	n/a	n/a	n/a	1.3	41.33	10.7	10.9
Our results									
Llama3.2-1B (off-the-shelf)	1B	48.6	48.0	48.9	50.7	2.1	52.3	17.7	10.2
further pre-trained, NSPP with:									
affixal cues		**50.1** (+3.0)*	**49.3**	**50.7**	**51.6**	**3.3**	**53.3**	**19.5**	**12.0**
single-word cues		48.8 (+0.3)	48.3	49.3	48.4	1.8	52.4	17.7	10.3
multi-word cues		49.3 (+1.4)*	48.5	50.0	50.7	2.8	52.6	18.7	10.8
All cues		48.8 (+0.3)	48.4	48.9	50.2	2.3	52.4	17.7	10.3
Qwen2-0.5B (off-the-shelf)	0.5B	49.6	48.4	50.9	49.6	3.8	47.0	18.4	15.2
further pre-trained, NSPP with:									
affixal cues		49.9 (+0.7)	49.6	**50.7**	46.2	**5.1**	45.1	17.4	**17.4**
single-word cues		48.4 (−2.5)	48.1	49.0	44.2	4.1	44.9	16.1	15.6
multi-word cues		**50.0** (+0.8)	**49.8**	50.2	**48.9**	4.7	**46.1**	**17.6**	**17.4**
All cues		48.9 (−1.4)	48.8	49.3	45.4	4.3	45.2	17.0	16.6

Table 2: Results on CondaQA with encoder-only LMs and (smaller) LLMs and zero-shot prompts. [1] refers to Ravichander et al. (2022), [2] to Rezaei and Blanco (2024), and [3] to Rezaei and Blanco (2025). Δ indicates the relative performance with respect to the off-the-shelf models, and an asterisk (*) indicates a statistically significant difference (McNemar's test (McNemar, 1947), $p < 0.05$). Both the LMs and LLMs benefit from further pre-training with NSPP and NegCue, our corpus, especially with affixal negation cues. We outperform all previous work except UnifiedQA-v2-11B, which uses ×11 the parameters and is fine-tuned with over 1M question-answer pairs.

As shown in Table 2 (Accuracy, All), both encoder-only LMs and LLMs further pre-trained for NSPP outperform their off-the-shelf counterparts (BERT: +3.3–7.1% depending on the subset of NegCue; RoBERTa: +20.5–25.8%; Llama: +0.3–3.0%). Except for Qwen (+0.7–0.8%), the differences with respect to the off-the-shelf models are statistically significant, with affixal pre-training yielding the strongest performance.

Our results with RoBERTa and NSPP pre-training with affixal negation (Acc.: 68%) outperform previous work (Rezaei and Blanco, 2025) targeting only three frequent negation cues (*not*, *n't*, and *never*). Further, unlike the affirmative interpretations of Rezaei and Blanco (2024, 67.1%), we do not require additional compute nor double the input size at inference time. Lastly, we also outperform UnifiedQA-v2-large (66.7%) despite having twice the parameters and is fine-tuned with 1M question-answer pairs.

Cue Type Analysis. Across LMs and LLMs, pre-training on affixal and multi-word cues yields larger relative improvements over off-the-shelf models while gains from single-word pre-training

	Accuracy				Group Consistency			
	Q_1 (Δ%)	Q_2 (Δ%)	Q_3 (Δ%)	Q_4 (Δ%)	Q_1	Q_2	Q_3	Q_4
BERT-large	46.7	47.1	45.9	48.8	2.5	1.7	1.8	0.0
+ NSPP, affixal	50.0 (+7.1)	50.3 (+6.8)	49.5 (+7.8)	52.2 (+7.0)	7.4	4.6	7.1	2.0
RoBERTa-large	55.3	52.0	48.6	51.4	14.3	8.0	8.8	12.0
+ NSPP, affixal	68.7 (+24.2)	67.4 (+29.6)	66.6 (+37.0)	62.1 (+20.8)	34.5	32.1	31.0	22.0
Llama3.2-1B	48.9	48.3	47.4	48.0	2.3	2.6	0.9	0.0
+ NSPP, affixal	50.3 (+2.9)	49.5 (+2.5)	49.3 (+4.0)	49.1 (+2.3)	3.4	3.4	1.7	2.0
Qwen2-0.5B	50.2	50.0	46.1	45.7	4.9	2.1	2.6	0.0
+ NSPP, multi	50.5 (+0.6)	48.5 (−3.0)	48.9 (+6.1)	47.5 (+3.9)	4.6	3.4	6.1	2.0

Table 3: Results with the best performing NSPP pre-training per model (Table 2) on four subsets of CondaQA containing cues based on frequency (four quartiles; Q_1 contains the top-25% most frequent cues). Δ indicates the relative change with respect to the off-the-shelf model. Pre-training with NSPP is beneficial regardless of cue frequency with both LMs and LLMs. In particular, we observe substantial gains in consistency.

are consistently smaller. This pattern suggests that models already handle single-word negation reasonably well, whereas affixal and multi-word negation remain more challenging and therefore benefit more from targeted pre-training. This interpretation is supported by the accuracies calculated depending on the cue types in the test instances of CondaQA (Table 2, Columns 4–6). Across all models, accuracy is generally highest on the single-word subset, indicating stronger baseline performance on this cue type. In contrast, affixal and multi-word subsets exhibit lower baseline accuracy.

Pre-training with affixal negation is the most beneficial at test time across all negation cues. We hypothesize that this is due to the fact that there are many more unique affixal cues (Figure 2, Table 1).

Cue Frequency Analysis. We further analyze results on CondaQA by negation cue frequency (Table 3). Specifically, we create four partitions of the test split depending on cue frequency (four quartiles, Q_1–Q_4, where Q_1 contains the top-25% most frequent cues). Across off-the-shelf models, accuracy and group consistency tend to be higher on subsets containing more frequent cues, indicating that commonly occurring cues are easier for off-the-shelf models. Pre-training with NSPP results in consistent improvements regardless of cue frequency. We note that the relative improvements of pre-training (Δ%) are often larger for the 50% least frequent negation cues (Q_3 and Q_4). In other words, NSPP pre-training benefits all negation cues at test time, but especially the infrequent ones.

6.2 Results on other NLU Benchmarks

We next discuss results on the remaining four NLU benchmarks (Table 4). Both LMs and LLMs pre-trained on affixal or multi-word cues generally achieve the best performance. This trend aligns with our findings on CondaQA, indicating that targeted exposure to these cue types is beneficial.

Results on NeQA. Overall, both LMs and LLMs benefit from further pre-training on our corpora. Interestingly, BERT-large outperforms RoBERTa-large on NeQA (83% vs. 51%), despite the latter being a stronger model in many settings, consistent with the inverse scaling behavior reported by the authors of this benchmark. For LLMs, LLaMA3.2-1B further pre-trained on our corpora outperforms the much larger Cohere-medium (45% vs. 44%), suggesting that negation-focused pre-training can be more effective than increasing model size alone for negation understanding.

Results on NevIR. The results show that further pre-training on our corpora leads to clear gains on negation-sensitive information retrieval tasks. Across encoder-only LMs, all further pre-trained variants outperform their off-the-shelf counterparts (BERT: +2.3–3.3%, RoBERTa: +1.3–3.5%). For LLMs, performance varies across models. Qwen2-0.5B shows limited benefits from further pre-training under some cue settings (-10.5–+11.6%), whereas LLaMA3.2-1B benefits substantially, achieving improvements over 200%.

Results on NMoNLI and ScoNe-NLI. Across both benchmarks, further pre-training consistently benefits both LMs and LLMs: the best-performing variant outperforms the off-the-shelf counterparts. Improvements are substantially larger on NMoNLI than on ScoNe-NLI, with both LMs achieving gains exceeding 200% on NMoNLI. In contrast, Qwen2-0.5B shows limited benefits from further pre-training on both NLI benchmarks. We observe

		NeQA	NevIR	NMoNLI	ScoNe-NLI
	#Pars.	Acc. (Δ%)	Acc. (Δ%)	Acc. (Δ%)	Acc. (Δ%)
Fine-tuned encoder-only LMs					
Previous Work					
stsb-roberta-large (Weller et al., 2024)	355M	n/a	24.9	n/a	n/a
MonoT5 3B (Nogueira et al., 2020)	3B	n/a	50.6	n/a	n/a
Our results					
BERT-large (off-the-shelf)	340M	70.0	52.1	6.0	62.8
further pre-trained, NSPP with					
affixal cues		**83.0** (+18.6)	53.5 (+2.7)	8.0 (+33.3)	63.3 (+0.8)
single-word cues		80.0 (+14.3)	53.6 (+2.9)	6.0 (+0.0)	**64.6** (+2.9)
multi-word cues		81.0 (+15.7)	**53.8** (+3.3)	7.5 (+25.0)	**64.6** (+2.9)
All cues		78.0 (+11.4)	53.3 (+2.3)	**19.5** (+225.0)	64.5 (+2.7)
RoBERTa-large (off-the-shelf)	355M	42.0	60.7	6.0	64.4
further pre-trained with					
affixal cues		44.0 (+4.8)	61.5 (+1.3)	14.0 (+133.3)	61.7 (−4.2)
single-word cues		45.0 (+7.1)	61.7 (+1.6)	7.0 (+16.7)	64.3 (−0.2)
multi-word cues		**51.0** (+21.4)	**62.8** (+3.5)	**21.5** (+258.3)	**64.6** (+0.3)
All cues		**51.0** (+21.4)	61.8 (+1.8)	4.5 (−25.0)	64.3 (−0.2)
LLMs, zero-shot					
Previous Work					
text-davinci-003 (Zhang et al., 2023)	n/a	71.0	n/a	n/a	64.0
Cohere-medium (Zhang et al., 2023)	6.1B	44.0	n/a	n/a	n/a
text-davinci-002 (She et al., 2023)	n/a	n/a	n/a	n/a	66.0
Our results					
Llama3.2-1B (off-the-shelf)	1B	38.0	4.6	43.5	49.2
further pre-trained, NSPP with					
affixal cues		40.0 (+5.3)	11.0 (+139.1)	**51.0** (+17.2)	**51.2** (+4.1)
single-word cues		39.0 (+2.6)	7.8 (+69.6)	50.0 (+14.9)	49.8 (+1.2)
multi-word cues		**45.0** (+18.4)	11.1 (+141.3)	**51.0** (+17.2)	**51.2** (+4.1)
All cues		38.0 (+0.0)	**14.1** (+206.5)	50.0 (+14.9)	49.8 (+1.2)
Qwen2-0.5B (off-the-shelf)	0.5B	16.0	9.5	49.5	50.0
further pre-trained, NSPP with					
affixal cues		**18.0** (+12.5)	**10.6** (+11.6)	49.5 (+0.0)	**50.1** (+0.2)
single-word cues		17.0 (+6.3)	8.8 (−7.4)	**50.0** (+1.0)	49.7 (−0.8)
multi-word cues		16.0 (+0.0)	10.3 (+8.4)	49.5 (+0.0)	50.1 (+0.2)
All cues		17.0 (+6.3)	8.5 (−10.5)	49.5 (+0.0)	50.0 (+0.0)

Table 4: Results on four NLU benchmarks. Δ indicates relative improvements with respect to the off-the-shelf model. Both LMs and LLMs further pre-trained with NSPP and NegCue, our corpus, yield better results than their off-the-shelf counterparts across the four corpora. We also outperform results by previous work (when available) by substantial margins despite experimenting with much smaller models (e.g., NeQA, +1% accuracy despite our LLM being six times smaller; NevIR, +12.2% accuracy despite our LM being ten times smaller).

that this model often predicts a single label across many instances, a behavior that is already present in its off-the-shelf version, suggesting that these tasks are particularly challenging for this model.

7 Conclusions

Negation remains a challenging problem for modern language models, and prior approaches often rely on heavy supervision or work with large-scale models. We have shown that further pre-training on NSPP, a negation-focused task, provides a simple, model-agnostic, and effective alternative. We construct NegCue, a large-scale corpus (>1.6M instances) covering over 200 negation cues beyond frequent and commonly studied cues such as *not* and *never* (1/3 affixal, 1/3 single-word, and 1/3 multi-word negation cues).

We further pre-train both encoder-only LMs and smaller LLMs for Next Sentence Polarity Prediction with NegCue. Experiments on five negation-related benchmarks demonstrate consistent improvements over off-the-shelf models. Pre-training on affixal and multi-word cues consistently yields larger gains than single-word cues. This suggests that current models struggle more with these cue types and therefore benefit more from targeted exposure.

Limitations

This work focuses on expanding the coverage of negation cues beyond a small set of commonly studied cues such as *not*, *n't*, and *never*. Our analysis, however, is limited to English corpora, and extending it to other languages with different negation systems remains future work. While NegCue is a large-scale corpus following the natural distribution of over 200 negation cues, exploring cue-balanced variants may further clarify the impact of distributional effects.

We show that smaller LLMs benefit from further pre-training on our corpora, whereas larger models around 10B parameters do not exhibit consistent gains. Identifying effective strategies to improve negation understanding in larger models is therefore an important direction for future research. Finally, while a simple zero-shot prompting strategy already yields promising results, it would be worthwhile to investigate few-shot and alternative prompting strategies.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

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*, pages 163–173, Online. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Laurence R. Horn. 1989. *A Natural History of Negation*. University of Chicago Press, Chicago.

Laurence R. Horn and Heinrich Wansing. 2025. Negation. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Spring 2025 edition. Metaphysics Research Lab, Stanford University.

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*, pages 3869–3885, Online. Association for Computational Linguistics.

Md Mosharaf Hossain and Eduardo Blanco. 2022. Leveraging affirmative interpretations from negation improves natural language understanding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5833–5847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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*, pages 1301–1312, Online. Association for Computational Linguistics.

Rodney D. Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press.

Shrikant Joshi. 2012. Affixal negation–direct, indirect and their subtypes. *Syntaxe & sémantique*, 13(1):49–63.

Emanuele Lapponi, Erik Velldal, Lilja Øvrelid, and Jonathon Read. 2012. UiO 2: Sequence-labeling negation using dependency features. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 319–327, Montréal, Canada. Association for Computational Linguistics.

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 R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, Andrew Gritsevskiy, Daniel Wurgaft, Derik Kauffman, Gabriel Recchia, Jiacheng Liu, Joe Cavanagh, Max Weiss, Sicong Huang, The Floating Droid, Tom Tseng, Tomasz Korbak, Xudong Shen, Yuhui Zhang, Zhengping Zhou, Najoung Kim, Samuel R. Bowman, and Ethan Perez. 2024. Inverse scaling: When bigger isn't better. *Preprint*, arXiv:2306.09479.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Roser Morante and Walter Daelemans. 2012. ConanDoyle-neg: Annotation of negation cues and their scope in conan doyle stories. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1563–1568, Istanbul, Turkey. European Language Resources Association (ELRA).

Roser Morante, Sarah Schrauwen, and Walter Daelemans. 2011. Annotation of negation cues and their scope: Guidelines v1. *Computational linguistics and psycholinguistics technical report series, CTRS-003*, pages 1–42.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Abhilasha Ravichander, Matt Gardner, and Ana Marasovic. 2022. CONDAQA: A contrastive reading comprehension dataset for reasoning about negation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8729–8755, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

MohammadHossein Rezaei and Eduardo Blanco. 2024. Paraphrasing in affirmative terms improves negation understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 602–615, Bangkok, Thailand. Association for Computational Linguistics.

MohammadHossein Rezaei and Eduardo Blanco. 2025. Making language models robust against negation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8123–8142, Albuquerque, New Mexico. Association for Computational Linguistics.

Jingyuan S. She, Christopher Potts, Samuel R. Bowman, and Atticus Geiger. 2023. ScoNe: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1803–1821, Toronto, Canada. Association for Computational Linguistics.

György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. 2008. The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 38–45, Columbus, Ohio. Association for Computational Linguistics.

Chantal van Son, Emiel van Miltenburg, and Roser Morante. 2016. Building a dictionary of affixal negations. In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics (ExProM)*, pages 49–56, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*, 2:2.

Jason Wei, Najoung Kim, Yi Tay, and Quoc Le. 2023. Inverse scaling can become U-shaped. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15580–15591, Singapore. Association for Computational Linguistics.

Orion Weller, Dawn Lawrie, and Benjamin Van Durme. 2024. NevIR: Negation in neural information retrieval. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2274–2287, St. Julian's, Malta. Association for Computational Linguistics.

Wikimedia-Foundation. 2023. Wikimedia downloads.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Yuhui Zhang, Michihiro Yasunaga, Zhengping Zhou, Jeff Z. HaoChen, James Zou, Percy Liang, and Serena Yeung. 2023. Beyond positive scaling: How negation impacts scaling trends of language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7479–7498, Toronto, Canada. Association for Computational Linguistics.

A Details of NegCue Corpus

Although the CondaQA dataset provides an *original cue* field, we do not directly rely on it for cue identification, as it includes tokens such as *could* and *loss* that do not explicitly express negation. As a result, we construct a refined cue list derived from the original CondaQA cues.

Using this cue list, we traverse the full Wikipedia corpus and extract sentence pairs that meet our criteria. Among the three cue types, multi-word cues yield the smallest number of samples, with approximately 309,857 instances. To construct a roughly class-balanced corpus, we pair these negative (NEG) instances with an equal number of affirmative (AFF) sentence pairs that do not contain negation, resulting in a multi-word corpus of 619,714 samples. We then randomly split each corpus into training, validation, and test sets using a 90:5:5 ratio, ensuring that all splits contain the corresponding negation cues and exhibit similar cue distributions.

Following the same procedure, we construct two additional corpora for the single-word and affixal cue types. Finally, we merge the corresponding training, validation, and test splits of the three corpora separately to form NegCue, a unified corpus that covers all negation cues.

B Experimental Details of NSPP Further Pre-training

All models are trained on one H100 GPUs with 80 GB of memory. For encoder-based Transformer models, namely BERT-large (Devlin et al., 2019) and RoBERTa-large (Liu et al., 2019), we perform self-supervised training on the NSPP objective for two epochs, with early stopping enabled based on validation performance. All LMs are further pre-trained with the same hyperparameter settings, using a batch size of 512, a learning rate of 1e-6, and a linear learning rate scheduler with warmup, following standard practice in Transformer-based models (Vaswani et al., 2017).

In addition to these two models, we also pre-trained and evaluated the relatively smaller BERT-base and RoBERTa-base models. After further pre-training on our corpus, both models show substantial improvements on CondaQA compared to their off-the-shelf counterparts.

For smaller large language models (LLMs), including LLaMA3.2-1B-Instruct (Dubey et al., 2024) and Qwen2-0.5B-Instruct (Yang et al., 2024), we adopt a zero-shot prompt-based formulation of the NSPP task and further pre-train the models using QLoRA (Dettmers et al., 2023) with 4-bit quantization. The zero-shot prompt used for NSPP pre-training is shown in Table 5.

All LLMs are pre-trained with a batch size of 16 and a cosine learning rate scheduler with warmup; the LoRA rank is set to 32, with learning rates of 2e-4 for LLaMA and 1e-5 for Qwen, respectively. Training is conducted for a single epoch, as these models converge rapidly and reach a performance plateau early, with only marginal decreases in validation loss thereafter. To mitigate overfitting, we apply standard early stopping with a patience of 3, and additionally introduce a convergence-based stopping criterion, terminating training when the validation loss decreases by less than 0.01 for three consecutive evaluations. Under these settings, training each LLM takes approximately three hours on average.

In addition to the two smaller models, we also trained and evaluated two larger language models, Llama3.1-8B (Grattafiori et al., 2024) and Qwen2-7B (Yang et al., 2024). However, we observe that further pre-training these larger models on our corpora yields limited performance gains.

C Experimental Details with Downstream Tasks

Next, we provide a detailed description of the downstream benchmarks used for evaluation, along with experimental details such as the hyperparameter choices for fine-tuning LMs and the prompts used for zero-shot evaluation of LLMs. Table 6 summarizes the dataset statistics for each benchmark, and Table 7 reports the hyperparameter settings used for fine-tuning LMs across benchmarks.

C.1 Experimental Details with CondaQA

CondaQA (Ravichander et al., 2022) provides training, validation, and test splits, with dataset statistics summarized in Table 6. In addition to the evaluation metrics reported in the paper, namely accuracy and group consistency, we further evaluate model performance across different negation types and cue frequency ranges. Specifically, we partition the CondaQA test set into three subsets based on negation cue type: affixal, single-word, and multi-word. We also group test instances into four quartiles (Q_1–Q_4) according to the frequency of the negation cues they contain, ordered from most frequent

Instruction:
You are a helpful assistant. In this task, you will be given the first sentence (S1) from a pair of consecutive sentences.
Your goal is to determine whether the second sentence (S2) would contain a negation with respect to the first sentence (S1).

The answer should be either "yes" or "no".
Only output one of the following labels: [YES] or [NO].

Input:
Sentence 1: The unusually good behavior of the cat causes the painter to feel more affectionate, and he names the animal "Good Fortune".

Response:

Table 5: Zero-shot prompt used to further pre-train LLMs on the NSPP task with our NegCue corpus.

	Training	Validation	Test
CondaQA	5832	1110 8	7240
NeQA	374	54	100
NevIR	948	225	1383
NMoNLI	1002	n/a	200
ScoNe-NLI	6012	n/a	1200
SNLI	366,180	6564	n/a

Table 6: The number of samples in the training, validation, and test sets of the benchmarks involved in our experiments.

	Learning Rate	Batch Size	Epochs
CondaQA	5e-6, 1e-5	8, 16	10
NeQA	1e-5	16	8
NevIR	2e-5	16	20
SNLI	2e-5	16	1

Table 7: Hyperparameters used for fully supervised fine-tuning of LMs across benchmarks. For CondaQA, we perform grid search to select the optimal hyperparameter configuration. For NMoNLI and ScoNe-NLI, models are instead fine-tuned on SNLI, as in-domain fine-tuning leads to near-saturated performance.

to least frequent.

For encoder-only LMs further pre-trained on our corpus, including BERT-large and RoBERTa-large, we fine-tune the models on the CondaQA training set. We perform grid search over learning rates {5e-6, 1e-5} and batch sizes {8, 16}, selecting the best configuration for each model based on validation performance. We follow the original setting and fine-tune the models for 10 epochs. The resulting models are then evaluated on the CondaQA test set using the metrics described above.

In contrast, for LLMs, we conduct evaluation using the five test splits provided by Ravichander et al. (2022) for in-context learning (ICL). Specifically, we evaluate LLaMA3.2-1B and Qwen2-0.5B under a zero-shot setting, using the same prompt as in the original paper (shown in Table 8), with the task description following Wang et al. (2022).

C.2 Experimental Details with NeQA

NeQA dataset. NeQA is a dataset consisting of 1,718 multiple-choice questions with negation sourced from several existing benchmarks (Zhang et al., 2023). Unlike CondaQA, NeQA does not provide predefined training, validation, or test splits. In the original work, the authors randomly sample 100 instances from the dataset as a test set for evaluation. To ensure consistency and comparability with prior work, we adopt the same 100 instances as our test set.

To enable encoder-only LMs to perform the question-answering task, we fine-tune the models directly on NeQA rather than on external QA datasets. Specifically, we construct training and validation splits from the remaining NeQA instances based on the dataset metadata, with dataset statistics reported in Table 6. To prevent data leakage, we ensure that no test instance appears in either the training or validation set.

Instance transformation for LM fine-tuning. Each NeQA instance consists of a question paired with two answer options. To make the task compatible with encoder-only LMs further pre-trained on the NSPP binary classification objective, we convert each original instance into two binary classification examples. Each converted example contains a declarative statement formed by combining the question with one answer option, along with a label indicating whether the option is correct. For example, given the question *"Birds cannot ?"* with options *(A) fly* and *(B) ulster*, where the correct answer is *(B)*, we convert the instance into two examples: *"Birds cannot fly."* labeled as *incorrect*, and *"Birds cannot ulster."* labeled as *correct*. As a result, the transformed dataset contains twice as many training instances as the original NeQA sam-

In this task, you're expected to write answers to questions involving reasoning about negation. The answer to the question should be "yes", "no", "don't know" or a phrase in the passage. Questions can have only one correct answer.

Passage: In October 2010, Daniela Soleri - Paolo Soleri's daughter - resigned from the Cosanti Foundation board, citing abuse by her father. She stated that some of Soleri's inner circle had been told decades earlier, but they remained actionless about it at the time. After the resignation, Soleri stepped down as chairman, but the board made no public statement on the reasons.
Question: Is it likely that people in Paolo's inner circle talked to him about the abuse of his daughter?
Answer:

Table 8: Zero-shot prompt used to evaluate LLMs with CondaQA.

The following are multiple choice questions (with answers) about common sense.

Question: Crying is not part of ?
A. life
B. love
Answer:

Table 9: Zero-shot prompt used to evaluate LLMs with NeQA.

ples. As shown in Table 6, we use 374 NeQA instances to construct the training data, yielding a total of 748 binary classification examples for LM fine-tuning.

Evaluation Settings. After building the training and validation sets, we fine-tune BERT-large and RoBERTa-large using a unified set of hyperparameters. Specifically, we use a learning rate of 1e-5, a batch size of 16, and fine-tune the models for 8 epochs. The fine-tuned models are then evaluated on the same 100 test instances used in the original NeQA work, and accuracy is reported for comparison. In contrast, LLMs are evaluated under a zero-shot setting. To mitigate prompt sensitivity in language model evaluation (Burns et al., 2022), we adopt the same prompt used in the original NeQA work, as shown in Table 9. This ensures a fair and consistent comparison with prior results.

C.3 Experimental Details with NevIR

NevIR (Weller et al., 2024) is an information retrieval benchmark that evaluates document ranking by identifying the document most relevant to a given query from a pair of candidates. The sizes of the training, validation, and test splits are reported in Table 6. Since our NSPP objective is formulated as a binary classification task, we adopt the Sentence-Transformers framework and fine-tune the further pre-trained LMs on the NevIR training set to adapt them to the document ranking setting.

Following the original NevIR work, where several information retrieval models are fine-tuned with a learning rate of 2e-5 for 20 epochs, we adopt the same hyperparameter settings. Specifically, both encoder-only LMs are fine-tuned using a learning rate of 2e-5 and a batch size of 16 for 20 epochs, and are evaluated on the test set using pairwise accuracy, as described in Section 5.2. Similarly, for LLM evaluation, we convert NevIR instances into instruction-style prompts. The prompt used in our experiments is shown in Table 10.

C.4 Experimental Details with NLI Benchmarks

Dataset structure. We evaluate two NLI benchmarks, NMoNLI and ScoNe-NLI, both of which provide only training and test splits. The corresponding dataset statistics are reported in Table 6. ScoNe-NLI (She et al., 2023) is derived from NMoNLI (Geiger et al., 2020) and extends it by introducing contrast sets that systematically vary the scope and interaction of negation. Specifically, ScoNe-NLI consists of six contrast splits, each containing 1,002 training instances and 200 test instances, designed to isolate the effects of negation scope on entailment relations. The six splits correspond to the following cases:

- *no_negation*: no negation is present.
- *one_not_scoped*: one negation that does not scope over the relevant lexical item.
- *two_not_scoped*: two negations, neither of which scopes over the relevant lexical item.
- *two_scoped*: two negations, where one scopes over the relevant lexical item, but the second scopes over the first, canceling it out.
- *one_scoped*: one negation that scopes over the relevant lexical item.
- *one_scoped_one_not_scoped*: two negations, but only one scopes over the relevant lexical item.

Together, these six splits yield a total of 6,012 training instances and 1,200 test instances across the full benchmark, as summarized in Table 6.

Evaluation Settings. Similar to NSPP, both NMoNLI and ScoNe-NLI are formulated as binary

In this task, you will be given two candidate documents and a query. Your goal is to decide which document answers the query. ##DOCUMENT 1: In 2012 Dyson published (with William H. Press) a fundamental new result about the prisoner's dilemma in the Proceedings of the National Academy of Sciences of the United States of America. He wrote a foreword to a treatise on psychic phenomena in which he concluded that "ESP is real... but cannot be tested with the clumsy tools of science". ##DOCUMENT 2: In 2012 Dyson published (with William H. Press) a fundamental new result about the prisoner's dilemma in the Proceedings of the National Academy of Sciences of the United States of America. He wrote a foreword to a treatise on psychic phenomena in which he concluded that "ESP is real... and has been tested with the clumsy tools of science". Output "1" if the first document contains the answer, or "2" if the second document contains the answer. You MUST answer with exactly one character: "1" or "2". Query: Who said that ESP is real though not testable by science? Only output "1" or "2". Response:

Table 10: Zero-shot prompt used to evaluate LLMs with NevIR.

This is a logical and commonsense reasoning exam. The answer must be Yes or No. Only output Yes or No. Question: Assume that the man does not own a dog. Is it definitely true that the man does not own a mammal? Answer:

Table 11: Zero-shot prompt used to evaluate LLMs with NMoNLI.

Question: Only Answer yes or no. Assume that the man does not own a dog and does not own a cat. Is it then definitely true that the man does not own an animal and does not own a cat? Answer:

Table 12: Zero-shot prompt used to evaluate LLMs with ScoNe-NLI.

classification tasks. Unlike NeQA and NevIR, no task reformulation is required. We therefore directly fine-tune BERT-large and RoBERTa-large on the training sets of these two benchmarks and evaluate the resulting models on their respective test sets.

However, we observe that after fine-tuning on the in-domain training data, LMs achieve near-saturated performance on the test sets, leaving little room for meaningful comparison with prior baselines. For example, RoBERTa-large attains 100% accuracy on all six ScoNe-NLI splits after fine-tuning. The detailed results for BERT-large and RoBERTa-large fine-tuned on ScoNe-NLI are reported in Table 13. To obtain more informative comparisons, we instead fine-tune both LMs on the widely used SNLI dataset (Bowman et al., 2015). The models are fine-tuned for one epoch with a learning rate of 2e-5 and a batch size of 16, and are then directly evaluated on the NMoNLI and ScoNe-NLI test sets.

For LLM evaluation, we convert the training instances into instruction-style prompts and perform zero-shot evaluation on the test sets. The prompts used for NMoNLI and ScoNe-NLI are shown in Tables 11 and 12, respectively. We report accuracy on NMoNLI, and for ScoNe-NLI, we report the average accuracy across all six splits.

	No Negation	One Not Scoped	Two Not Scoped	Two Scoped	One Scoped	One Scoped One Not Scoped
BERT-large (off-the-shelf)	93.5	97.0	94.0	97.5	98.5	98.5
+ NSPP, Affixal	99.0	99.0	98.0	99.5	99.0	100.0
+ NSPP, Single-word	98.5	99.5	99.5	100.0	99.0	99.0
+ NSPP, Multi-word	99.0	99.0	99.0	100.0	99.5	99.0
+ NSPP, All cues	99.0	99.5	98.0	99.5	99.0	100.0
RoBERTa-large (off-the-shelf)	100.0	100.0	97.5	99.5	99.5	99.5
+ NSPP, Affixal	100.0	100.0	100.0	100.0	100.0	100.0
+ NSPP, Single-word	100.0	100.0	100.0	100.0	100.0	100.0
+ NSPP, Multi-word	100.0	100.0	100.0	100.0	100.0	100.0
+ NSPP, All cues	100.0	100.0	100.0	100.0	100.0	100.0

Table 13: Accuracy (%) on the six ScoNe-NLI test splits for BERT-large and RoBERTa-large fine-tuned on the ScoNe-NLI training set. All models, including off-the-shelf baselines, achieve near-perfect results across all splits.