Thunder-NUBench: A Benchmark for LLMs' Sentence-Level Negation Understanding

Yeonkyoung So[1], Gyuseong Lee[1], Sungmok Jung[1], Joonhak Lee[1], JiA Kang[1], Sangho Kim[1], Jaejin Lee[1,2]

[1]Graduate School of Data Science, Seoul National University
[2]Dept. of Computer Science and Engineering, Seoul National University
{kathy1028,ksnannaya,tjdahrwjd,hmjelee,jia6776,ksh4931,jaejin}@snu.ac.kr

Abstract

Negation is a fundamental linguistic phenomenon that poses ongoing challenges for Large Language Models (LLMs), particularly in tasks requiring deep semantic understanding. Current benchmarks often treat negation as a minor detail within broader tasks, such as natural language inference. Consequently, there is a lack of benchmarks specifically designed to evaluate comprehension of negation. In this work, we introduce *Thunder-NUBench* — a novel benchmark explicitly created to assess sentence-level understanding of negation in LLMs. Thunder-NUBench goes beyond identifying surface-level cues by contrasting standard negation with structurally diverse alternatives, such as local negation, contradiction, and paraphrase. This benchmark includes manually created sentence-negation pairs and a multiple-choice dataset, allowing for a comprehensive evaluation of models' understanding of negation.

1 Introduction

Negation is a fundamental and universal phenomenon found in languages worldwide. It is closely associated with various human communicative abilities, such as denial, contradiction, deception, misrepresentation, and irony. Although affirmative statements are more common, negation still plays a significant role in language; approximately 25% of sentences in English texts contain some form of negation (Sarabi and Blanco, 2016; Hossain et al., 2020; Horn and Wansing, 2025). This prevalence and its impact on meaning make accurate interpretation of negation crucial for several natural language processing (NLP) tasks, including sentiment analysis, question answering, knowledge base completion, and natural language inference (NLI) (Khandelwal and Sawant, 2020; Hosseini et al., 2021; Singh et al., 2023). Recent studies have shown that effectively managing negation is important even for multi-modal language models (Quantmeyer et al., 2024; Alhamoud et al., 2025; Park et al., 2025).

```
{Task Instruction}

Generate the standard negation of the given sentence.
Sentence: {Original Sentence}

A. {Standard Negation} – Answer
B. {Local Negation} – Negation cues included, partial scope
C. {Contradiction} – Semantically incompatible
D. {Paraphrase} – Same meaning, different form

Answer:
```

Figure 1: Illustration of the Thunder-NUBench task. Only the standard negation reverses the truth value of the original sentence, while other options differ in scope or semantics.

Meanwhile, negation poses significant challenges for both humans and language models. Research shows that people often find it more difficult to process and comprehend negated statements compared to affirmative ones (Wales and Grieve, 1969; Sarabi and Blanco, 2016). Similarly, many studies indicate that pretrained language models (PLMs) struggle to interpret negation accurately. For example, models like BERT (Devlin et al., 2019) and even large language models (LLMs) such as GPT-3 (Brown et al., 2020) often have difficulty distinguishing between negated and affirmative statements. These models tend to rely on superficial cues, which can result in incorrect outputs when negation is involved (Kassner and Schütze, 2020; Hossain et al., 2022a; Truong et al., 2023).

Despite its significance, there is a notable lack of dedicated evaluation benchmarks for understanding negation. Most existing resources either treat negation as a minor aspect of broader tasks or focus solely on narrow syntactic detection, often emphasizing encoder-based models (Hossain et al., 2020; Geiger et al., 2020; Truong et al., 2022; An-

Findings of the Association for Computational Linguistics: EACL 2026, pages 4749–4793
March 24-29, 2026

schütz et al., 2023). To address this gap, we introduce *Thunder-NUBench* (*N*egation *U*nderstanding *Bench*mark), a dataset explicitly designed to evaluate LLMs' sentence-level comprehension of negation.[1] Our benchmark is structured as a multiple-choice question (MCQ) task: given an original sentence, the model must select the correct standard negation from four options. The other three choices (local negation, contradiction, and paraphrase) are carefully designed distractors that test whether models truly grasp semantic scope and logical oppositions.

The contributions of this paper are summarized as follows:

- We define standard negation within the framework of sentential logic. Grounding standard negation in logical structure not only clarifies its role in natural language but also supports the evaluation and enhancement of reasoning in LLMs.
- We introduce a manually created benchmark that includes a dataset of sentence-negation pairs for fine-tuning, along with a multiple-choice evaluation task.
- We conduct systematic evaluations of decoder-based LLMs across model families, scales, and training methods to analyze variation in negation understanding.

Thunder-NUBench provides valuable insights into the semantic reasoning abilities of language models and serves as a robust standard for future research focused on understanding negation.

2 Related Work

Negation detection and scope resolution. Early work in negation detection and scope resolution primarily relied on rule-based systems and handcrafted heuristics, especially in domain-specific contexts like clinical texts. While these systems are effective, they lack flexibility across different domains (Chapman et al., 2001; de Albornoz et al., 2012; Ballesteros et al., 2012; Basile et al., 2012). Traditional machine learning methods, such as Support Vector Machines (SVMs) (Hearst et al., 1998) and Conditional Random Fields (CRFs) (Sutton et al., 2012), were introduced later; however, they too are limited to narrow domains (Morante et al., 2008; Morante and Daelemans, 2009; Read et al., 2012; Li and Lu, 2018).

[1]Thunder-NUBench is publicly available at https://huggingface.co/datasets/thunder-research-group/SNU_Thunder-NUBench.

More recently, deep learning approaches employing Convolutional Neural Networks (CNNs) (O'shea and Nash, 2015) and Bidirectional Long Short-Term Memory (BiLSTM) networks (Siami-Namini et al., 2019) have enhanced performance by providing improved contextual embeddings and sequence modeling (Fancellu et al., 2016; Bhatia et al., 2019). Pretrained transformer models like BERT have been employed through transfer learning techniques (e.g., NegBERT (Khandelwal and Sawant, 2020)), significantly increasing the accuracy of negation detection tasks. Nonetheless, these methods still largely focus on syntactic span detection, leaving deeper semantic understanding of negation a challenging area to tackle.

Negation-sensitive subtasks of NLU. Negation understanding has become increasingly important in natural language understanding (NLU) tasks (Hosseini et al., 2021). However, existing NLU benchmarks, such as SNLI (Bowman et al., 2015) for natural language inference (NLI), CommonsenseQA (Talmor et al., 2019) for Question Answering (QA), SST-2 (Socher et al., 2013) for sentiment analysis, STS-B (Cer et al., 2017) for textual similarity and paraphrasing, have been criticized for not adequately addressing the semantic impact of negation (Hossain et al., 2022a; Rezaei and Blanco, 2024). These datasets contain relatively few instances of negation or include negations that are not crucial to task performance, allowing language models to achieve high accuracy even when they completely ignore negation.

Recent studies, including NegNLI (Hossain et al., 2020), MoNLI (Geiger et al., 2020), NaN-NLI (Truong et al., 2022), NoFEVER-ML and NoSNLI-ML (Vrabcová et al., 2025), have introduced benchmarks for NLU that includes negation. These studies show that model performance significantly declines when negation plays a crucial role in affecting the outcome (Naik et al., 2018; Yanaka et al., 2019; Hartmann et al., 2021; Hossain et al., 2022b; Hossain and Blanco, 2022; She et al., 2023; Anschütz et al., 2023). These findings suggest that current language models tend to depend on superficial linguistic patterns rather than a genuine understanding of semantics.

Limitations of distributional semantics. Distributional semantics (Harris, 1954; Sahlgren, 2008) aims to create models that learn semantic representations based on patterns of word co-

Dimension	Negation Type	Definition	Example
Scope	**Clausal Negation (= Sentential Negation)**	Negation that applies to the entire clause or sentence. This typically involves the use of "not", or its contracted form "n't" with auxiliary verbs.	He **speaks** English fluently. → He **doesn't speak** English fluently.
	Subclausal Negation (= Constituent / Local Negation)	Negation that focuses on negating a specific part of a clause, such as a word or phrase, rather than the entire clause.	He speaks English **fluently**. → He speaks English, **but not fluently**.
Form	**Morphological Negation**	Negation expressed through affixes attached to words such as prefixes like "un-", "in-", "dis-", or suffixes like "-less".	She is **happy**. → She is **unhappy**.
	Syntactic Negation	Negation expressed through separate words (particles) in the syntax, such as "not", "never", "no", etc.	She is happy. → She is not happy.
Target	**Verbal Negation**	Negation that applies directly to the verb or verb phrase.	They **have finished** the work. → They **have not finished** the work.
	Non-verbal Negation	Negation that negates elements other than the verb.	There is **milk** in the fridge. → There is **no milk** in the fridge.

Table 1: Typology of negation.

occurrences (Boleda, 2020; Lenci et al., 2022) and capture broad semantic relationships; however, it encounters significant challenges with negation. Negated expressions, such as "not good," often appear in similar contexts as their affirmative counterparts, like "good." As a result, models tend to generate similar vector representations for these expressions, despite their opposing meanings. Previous research has pointed out this limitation, showing that PLMs struggle to capture the subtle semantic differences introduced by antonyms and the reversal of polarity (Rimell et al., 2017; Jumelet and Hupkes, 2018; Niwa et al., 2021; Jang et al., 2022; Vahtola et al., 2022). Studies have further suggested that models like BERT find it difficult to distinguish between affirmative and negated contexts (Kassner and Schütze, 2020; Ettinger, 2020).

Negations in generative language models. Recent research on understanding negation has primarily focused on bidirectional models, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), which have demonstrated strong performance in NLU and negation detection tasks. However, with the emergence of generative foundation models like GPT (Radford et al., 2018) and LLaMA (Touvron et al., 2023), attention has shifted towards evaluating how these models handle negation. Studies have shown that these generative models often exhibit a positive bias and struggle with producing or interpreting negated statements (Truong et al., 2023; Chen et al., 2023; García-Ferrero et al., 2023). Although some benchmarks, such as CONDAQA (Ravichander et al., 2022) and ScoNe (She et al., 2023), reveal these limitations, there is still a lack of robust evaluation resources specifically designed for negation understanding of generative models.

Building on previous studies, this paper focuses on sentence-level negation as a core linguistic operation. While prior negation-related NLI or QA benchmarks evaluate negation within broader inference or comprehension tasks, Thunder-NUBench specifically tests whether models can distinguish standard negation, the logical reversal of the sentence, from closely related distractors. As a result, Thunder-NUBench enables a more direct examination of sentence-level negation than is afforded by NLI or QA benchmarks, where negation is evaluated only as part of broader inference tasks.

3 Scope and Categorization of Negation

In this work, we aim to clarify the concept of negation by introducing a typology that clearly outlines its semantic boundaries and differentiates it from related, yet distinct, phenomena. This typology organizes various forms of meaning reversal into logically consistent categories, allowing for a more precise and systematic evaluation of how language models handle negation.

3.1 Typology of Negation

Negation is a fundamental semantic and syntactic operation found in natural languages, used to convey denial, rejection, or the absence of a proposition. Hereafter, we denote our negation operation for a sentence S as $\text{Neg}(S)$. In formal logic, negation flips the truth value of a proposition P: if P is

Type	Definition	
Base case	If P is an atomic proposition, Neg(P) is the proposition where the main predicate of P is negated.	
Inductive step	**Conjunction**	Neg(P and Q) $\equiv$ Neg(P) or Neg(Q), Neg(P but Q) $\equiv$ Neg(P) or Neg(Q)
	Disjunction	Neg(P or Q) $\equiv$ Neg(P) and Neg(Q)
	Implication	Neg(if P, Q) $\equiv$ Neg(Neg(P) or Q) $\equiv$ P and Neg(Q) Neg(P if and only if Q) $\equiv$ Neg(if P, Q and if Q, P)

Table 2: Standard negation. P and Q stand for propositions. In addition to *and*, *or*, and *if*, other natural language connectives such as *when* are also considered, and their negations follow the same principles depending on their function.

true, then Neg(P) is false, and vice versa. Semantically, negation creates a binary opposition between a proposition and its affirmative counterpart, meaning that each one is the opposite of the other (Horn and Wansing, 2025).

Negation can be categorized along several dimensions: scope, form, and target (see Table 1). In terms of scope, negation may affect the entire clause (referred to as *clausal negation*) or only part of it (known as *subclausal negation*). Regarding form, negation can manifest as bound morphemes, such as prefixes and suffixes (*morphological negation*), or as separate syntactic elements like "not" or "never" (*syntactic negation*). Finally, depending on its target, negation can apply to the verb (*verbal negation*) or to other elements in the sentence (*non-verbal negation*) (Zanuttini, 2001; Miestamo, 2007; Truong et al., 2022; Kletz et al., 2023).

3.2 Negation and Contradiction

Negation and contradiction are closely related concepts that are often conflated in NLP research (Jiang et al., 2021). Contradiction refers to the incompatibility of two propositions, meaning that they cannot both be true at the same time. While negation frequently serves as a primary mechanism for creating contradictions (by reversing the truth value of a proposition), contradictions can also arise from antonymy, numeric mismatches, or differences in structure and lexicon (further details can be found in Appendix A). For instance, the statements "An individual was born in France" and "An individual was born in Italy" are contradictory, but they are not negations, as the second statement does not reverse the truth of the first.

Many previous studies have overlooked the possibility that contradictions can exist independently of explicit negation. Recognizing this gap, we specifically examine the ability of LLMs to differentiate between negations and non-negated contradictions, highlighting the nuanced semantic distinctions that are involved.

3.3 Standard Negation

Standard negation refers to the typical form of negation applied to the declarative verbal main clause. It specifically negates the verb in a *main clause* (Miestamo, 2000). A main clause can function as a complete sentence on its own, consisting at a minimum of a subject and a predicate. This definition is grounded in the notion that the verb acts as the head of the clause (Miller and Miller, 2011).

Building on this traditional understanding, we treat standard negation as *the process of reversing the truth value of the verb phrase in the main clause*, which we will refer to as the *main predicate* in this paper. A verb phrase is headed by a verb and can consist of a single verb or a combination of auxiliaries, complements, and modifiers (e.g., "will call" and "is being promoted") (Lakoff, 1966). Since the main predicate conveys the core action or state of the clause, negating it effectively reverses the proposition of the entire sentence. In this context, this paper treats standard negation as a *truth-functional operation that maps the main predicate to its complement set within the semantic space*.

We further clarify the scope of standard negation within the typology presented in Table 1. Standard negation includes both *clausal negation* and *verbal negation*, as it reverses the meaning of the entire sentence by negating the main predicate. In terms of form, standard negation can employ both *syntactic* and *morphological negation*. Syntactically, standard negation often uses explicit negation particles, such as "not." Morphologically, it can involve *complementary antonyms* (for example, "alive" vs. "dead" or "true" vs. "false"), which occupy mutually exclusive semantic spaces, thus reversing the truth value of the proposition. In con-

Type	Structure Explanation	Local Negation Example
Relative clause negation	A relative clause is a type of dependent clause that gives extra details about a noun or noun phrase in the main sentence. It usually begins with a relative pronoun such as *who, which, that, whom,* or *whose*.	The man who **owns** the car is my neighbor. → The man who **does not own** the car is my neighbor.
Participle clause negation	A participle clause is a type of dependent clause that begins with a participle (a verb form ending in *-ing* or a past participle). It acts like an adverb, giving extra details about the main clause, often showing time, reason, result, or sequence of actions.	**Walking** through the park, she found a lost wallet. → **Not walking** through the park, she found a lost wallet.
Adverbial clause negation	An adverbial clause is a dependent clause that acts like an adverb, modifying a verb, adjective, or adverb. It gives information such as time, reason, condition, or contrast. These clauses are introduced by subordinating conjunctions like *because, although*, or *while*.	She stayed inside because it **was raining**. → She stayed inside because it **was not raining**.
Compound sentence with local negation	A compound sentence consists of two or more main clauses joined by coordinating conjunctions such as *and, but*, or *or*. If only one of these clauses is negated, the negation applies only locally to that clause.	He submitted the report and **attended** the meeting. → He submitted the report and **did not attend** the meeting.

Table 3: Typology of local negation.

trast, other types of antonyms, such as *gradable antonyms* (e.g., "happy" vs. "unhappy") and *relational antonyms* (e.g., "buy" vs. "sell") (Lehrer and Lehrer, 1982), do not strictly reverse truth values. Therefore, they are classified as contradictions rather than standard negation in this paper.

Atomic and Complex Propositions. While this characterization of standard negation effectively defines standard negation for atomic propositions (elementary sentences that cannot be further decomposed) (Davis and Gillon, 2004), its application to complex sentences with multiple clauses requires a more thorough approach. In this paper, we treat an atomic proposition as *a sentence that contains a single main predicate*. Specifically, for propositions composed of multiple logically connected atomic statements, the method for reversing the truth value of the entire complex proposition can be ambiguous. In natural language, such logical structures typically appear as coordinated clauses (e.g., "P and Q or R") or comma-separated lists connected by "and" or "or" (e.g., "P, Q, and R"). We treat these as equivalent to a sequence of binary conjunctions or disjunctions.

Definition of standard negation. In this paper, standard negation refers to natural-language sentential negation, which is formally treated as logical negation within the framework of sentential logic (Enderton, 2001). To address the complexities involved, we define standard negation recursively by applying it pairwise over the logical structure of a sentence until only atomic propositions remain, ensuring that the truth value of the entire sentence is reversed even when it contains multiple coordinated clauses. Our definition of Neg$(\cdot)$ is presented in Table 2. Conditionals of the form "if P, Q" are equivalent to "Neg(P) or Q" in logic, and we adhere to this equivalence when defining their negation (more details can be found in Appendix C).

3.4 Local Negation

We define *local negation* as a form of negation that specifically targets a verb phrase outside the main clause. While the term is often used interchangeably with subclausal negation, our focus is solely on local negation relating to subclausal and verbal negation. This concept applies to four types of sentence structures: relative clauses, participle clauses, adverbial clauses, and compound sentences (refer to Table 3 for more details).

In particular, conditional clauses, such as the "if P" part in "if P, Q" are categorized as adverbial clauses. In compound sentences, standard negation requires all main clauses to be negated in order to achieve sentence-level negation. If only a subset of the clauses is negated, this is considered local negation.

Local negation, in terms of structure, resembles standard negation, typically using explicit negation markers like "not." However, its scope is confined to a specific part of the sentence rather than encompassing the entire main clause. Because explicit cues such as "not" are still present, models that depend on shallow cue detection may be misled,

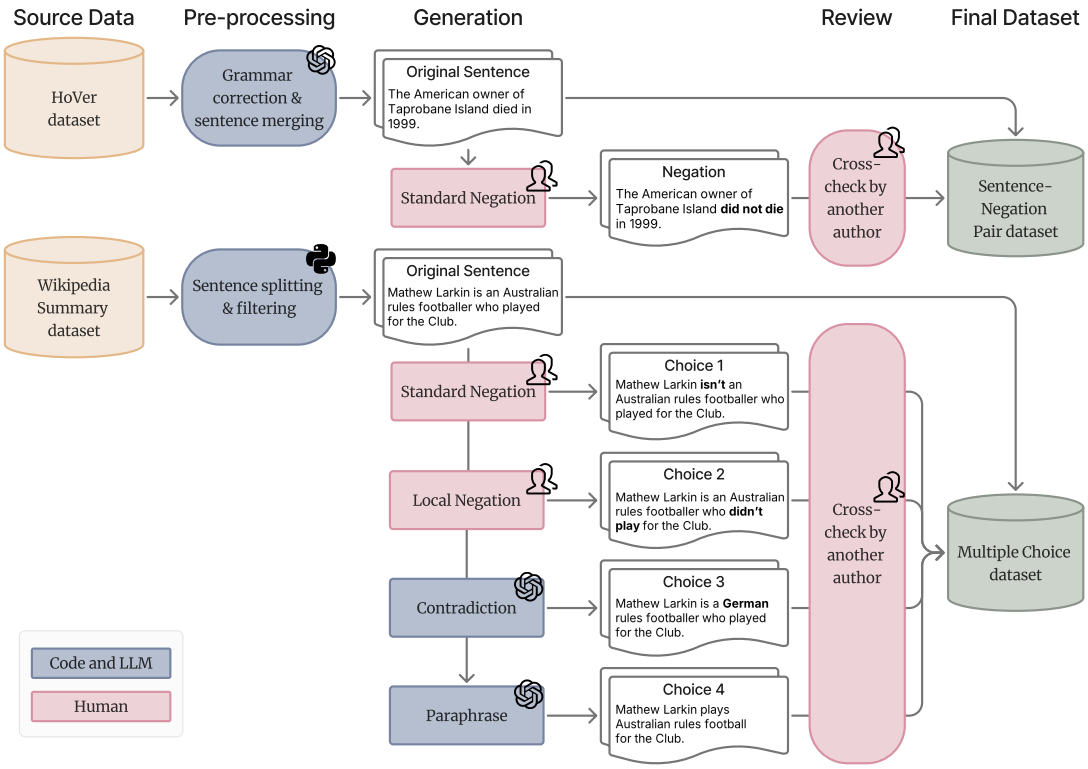

Figure 2: Dataset generation process.

failing to distinguish between standard negation and local negation.

4 Thunder-NUBench Dataset

We construct the Thunder-NUBench dataset through three main stages: (1) pre-processing, (2) generation, and (3) review. The overall workflow is illustrated in Figure 2.

Pre-processing. We begin by extracting sentences from two primary corpora: (1) the Hover dataset (Jiang et al., 2020), designed for multi-hop fact extraction and claim verification, and (2) the Wikipedia Summary dataset (Scheepers, 2017), which contains concise summaries from English Wikipedia. We chose these datasets because their factual content and complex sentence structures are well-suited for developing a dataset aimed at understanding standard negation in complex, sufficiently lengthy sentences. Additionally, we automatically correct any grammatical errors and merge or split sentences as needed to create well-formed single-sentence units.

Generation. We create two types of datasets from the pre-processed sentences: the *sentence-negation pair dataset* and the *multiple choice dataset*. In the sentence-negation pair dataset, each original sentence is paired with a manually crafted standard negation, as detailed in Section 3.3. In the multiple-choice dataset, each original sentence is presented with four options: a standard negation, a local negation, a contradiction, and a paraphrase. Each of them is described in Table 4. Together, these categories assess whether models truly understand semantic negation rather than relying on superficial cues.

Standard and local negation options are manually created rather than generated by LLMs. We have observed that LLMs often struggle to produce correct standard negations, frequently resulting in subclausal or local negations instead. They can also generate incorrect local negations, even when explicitly prompted to do otherwise. Since precise negation is essential to our benchmark, these options must be developed by humans to ensure the quality of the dataset. In contrast, contradiction and paraphrase options are initially created automatically using carefully designed prompts with the OpenAI API (OpenAI, 2025) and are then refined during the review process. Details of the GPT models and code used at each stage of the data generation process are provided in Appendix K.

Review. All constructed data undergo a multi-stage human review process (see Appendix I). A

Category	Description
Standard Negation	This category involves reversing the truth value of the main clause, which is the primary focus of the benchmark.
Local Negation	In this case, negation is applied to a subordinate clause or a partial structure, which does not reverse the entire sentence.
Contradiction	This category introduces conflicts with the original meaning through semantic changes, such as the use of antonyms, different numbers, or other entities, without employing explicit negation.
Paraphrase	Here, the original meaning is preserved while the surface form is altered. Examples of paraphrases are intentionally constructed to vary the sentence structure and word choice significantly, ensuring that no additional information is added. As a result, the original sentence still entails its paraphrase. This category tests whether models mistakenly consider different surface forms as meaning reversals, even when the semantic meanings remain equivalent.

Table 4: Multiple choice categories included in Thunder-NUBench.

different author, separated from the creator, cross-checks each instance, and any disagreements are addressed in regular meetings to ensure consistency. Options for contradictions are reviewed only after the corresponding standard and local negations are finalized, as they must not overlap semantically. Consequently, the earlier negations are re-examined during the contradiction review and are cross-checked by multiple authors.

The guidelines for data generation and review are continuously updated, and any previously created data are revised accordingly (see Appendix J). This protocol ensures rigorous quality control and consistency throughout the benchmark.

Dataset statistics. The final dataset includes a training set of sentence-negation pairs and a multiple-choice evaluation set (see Table 5). For few-shot prompting, we construct a demonstration set of 50 examples. These are carefully selected to have unique Wikipedia page indices to avoid any overlap with the test set. Furthermore, to provide the model with a balanced overview of the task, we match the distribution of local negation types (`choice2_type`) in the demonstration set to that of the overall dataset. This ensures that the demonstrations are representative and prevents the model from developing a biased strategy for specific negation types.

Dataset	Split	Count
Sentence-Negation	Train	3,772
Multiple Choice	Demonstration	50
	Test	1,261
	Total	5,083

Table 5: Thunder-NUBench statistics.

5 Experiments

5.1 Evaluation Setup

We evaluate models under two common Multiple-Choice Question Answering (MCQA) settings: (1) a completion-based evaluation, where the model assigns probabilities to each candidate by appending it as a continuation of the prompt, and (2) an option-selection evaluation, where the model selects from labeled options (A/B/C/D). To mitigate known issues such as selection or position bias in the option-selection setting, we randomly shuffle the order of the options (using random seed 42). In both settings, we use two instruction variants: a definition-based instruction (referred to as the *definition* instruction) and a step-by-step instruction (referred to as the *detailed* instruction). Details of the prompt templates and formatting are provided in Appendix M.

We report results for a diverse set of pretrained and instruction-tuned models, evaluated under zero-shot and few-shot settings with 1, 5, and 10 examples, as well as supervised fine-tuning (SFT). For SFT, models are trained on the sentence-negation pair dataset introduced in Thunder-NUBench. The full list of models, evaluation protocols, and fine-tuning details is provided in Appendix N.

5.2 Overall Performance and Key Findings

Effect of Model Size. Performance tends to improve as model size increases, indicating that negation understanding benefits from larger models. The improvement is more pronounced in the option-selection evaluation than in the completion-based setting, with larger models showing higher accuracy gains (as seen in Figure 3). This suggests that option selection benefits more from larger models, since it requires comparing multiple candidate answers rather than generating a single continuation.

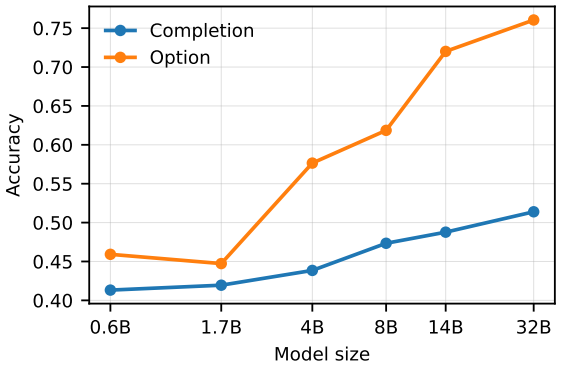

Figure 3: Zero-shot accuracy of instruction-tuned Qwen3 models under the detailed prompt setting.

Effect of Instruction Tuning. Instruction tuning generally improves performance across most model families and evaluation settings. As shown in Figure 4, this improvement is observed in both completion-based and option-selection evaluations. We further observe that instruction tuning yields larger improvements under the detailed prompt setting than under the definition-based instruction. This suggests that instruction-tuned models follow step-by-step instructions more reliably, leading to larger gains in negation understanding.

However, the effect of instruction tuning is not uniform across all models. In particular, some Qwen3 models exhibit comparable or higher performance in their pretrained versions, indicating that instruction tuning does not always guarantee gains for negation understanding.

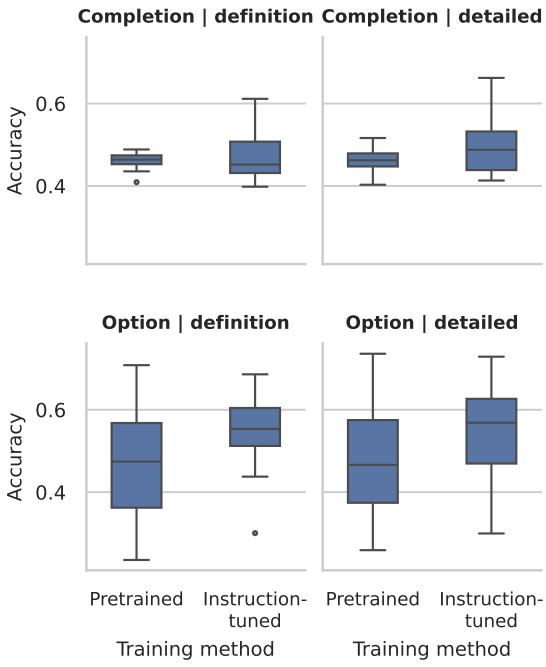

Figure 4: Boxplots showing the distribution of zero-shot accuracy for pretrained and instruction-tuned models across evaluation settings, using both definition-based and detailed prompt instructions.

Effect of Few-Shot Learning. Overall, increasing the number of in-context demonstrations tends to improve performance across few-shot settings. This indicates that few-shot learning generally benefits sentence-level negation understanding. Figure 5 illustrates this trend for the Llama-3.1-8B-Instruct model under definition-based instruction. The initial introduction of demonstrations leads to a clear performance improvement, while the incremental benefits decrease as more examples are added.

This trend, however, is not uniform. For some API-based models under the detailed instruction, additional demonstrations do not always yield further gains. One possible reason is that the detailed instructions already provide strong guidance for API-based large models, and adding demonstrations may introduce noise or encourage unnecessary pattern following.

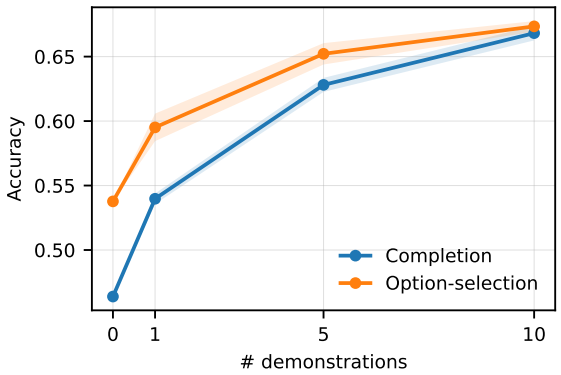

Figure 5: Mean accuracy of Llama-3.1-8B-Instruct under the definition instruction across different numbers of in-context demonstrations.

Effect of Supervised Fine-Tuning. Supervised fine-tuning (SFT) generally leads to improved zero-shot performance across most models, indicating that task-specific training further enhances negation understanding. The performance gains from SFT are more pronounced in the completion-based evaluation than in the option-selection setting, suggesting that the effects of fine-tuning may interact with the evaluation format. We observe that SFT does not degrade models' general language abilities, as confirmed by evaluations on general

Evaluation Formats	Instruction Formats	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
					Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
completion-based	definition	baseline	zero-shot	0.536	73.82	23.96	2.22	25.00	31.82	58.50	48.71
			5-shot	0.367	85.31	14.25	0.43	18.91	22.40	40.14	48.06
		after SFT	zero-shot	0.229	86.51	12.46	1.04	10.90	14.29	23.13	33.55
	detailed	baseline	zero-shot	0.468	71.86	26.44	1.69	23.08	29.87	47.62	38.71
			5-shot	0.347	84.70	14.84	0.46	19.55	23.05	37.76	41.29
		after SFT	zero-shot	0.211	87.97	10.90	1.13	8.97	13.31	21.09	33.23
option-selection	definition	baseline	zero-shot	0.462	70.67	18.52	10.81	27.56	24.68	53.40	30.00
			5-shot	0.332	81.62	11.22	7.16	25.64	18.83	40.48	27.42
		after SFT	zero-shot	0.356	71.94	15.37	12.69	25.00	19.81	44.56	17.10
	detailed	baseline	zero-shot	0.486	71.45	14.85	13.70	33.01	25.97	56.12	29.03
			5-shot	0.353	77.53	11.24	11.24	27.56	19.81	39.12	26.77
		after SFT	zero-shot	0.270	65.69	21.70	12.61	14.74	14.61	29.25	15.16

Table 6: Error distribution and confusion analysis of Llama-3.1-8B-Instruct model across various evaluation settings.

benchmarks (Appendix P).

Detailed results for all models and training settings are provided in Appendix O. In addition, we conducted a human evaluation, showing that although humans perform well on average, distinguishing standard negation from closely related alternatives is not uniformly easy across participants. Detailed results are provided in Appendix R.

5.3 Error Analysis

We analyze model errors to evaluate the ability of our models to differentiate standard negation from similar semantic variants. Each type of local negation in our dataset is explicitly labeled based on its sentence structure, as defined in Table 3.

We measure the *confusion rate*, defined as the proportion of examples within each subtype where the model incorrectly selects the local negation option instead of the correct standard negation. For example, if 320 items are labeled as participle clause negation and the model incorrectly chooses the local negation option instead of the correct standard negation option in 32 of these cases, the confusion rate for participle clause negation would be 10%. Complete analysis results are provided in Appendix Q.

We focus on the results of Llama-3.1-8B-Instruct model as shown in Table 6. In the completion-based setting, error rates generally decrease from zero-shot to 5-shot and continue to improve after SFT, with most errors concentrated in local negation options. Within local negation, compound sentences exhibit the highest confusion but also show the largest relative improvement after SFT.

In the option-selection setting, errors are likewise dominated by local negation, while the model shows a higher tendency to confuse paraphrase options compared to the completion-based setting. Comparatively, the option-selection setting results in lower confusion rates for adverbial clause negation. In addition, performance improvements in the option-selection setting do not consistently follow the expected progression across training configurations. In some cases, SFT yields higher error rates than the few-shot baselines, and this pattern is observed in some other models as well.

Overall, these patterns highlight how different evaluation settings and model configurations lead to distinct types of errors, and how the addition of more examples or SFT affects error distribution.

6 Conclusion

In this work, we introduce Thunder-NUBench, a benchmark designed to evaluate LLMs' sentence-level understanding of negation, going beyond surface cue detection. By distinguishing between standard negation, local negation, contradiction, and paraphrase, Thunder-NUBench offers a comprehensive assessment of semantic comprehension. Our experiments demonstrate that while supervised fine-tuning and in-context learning can help reduce specific errors, these approaches still struggle to differentiate standard negation from closely related semantic variants. Thunder-NUBench serves as a valuable diagnostic tool for analyzing the limitations of models' understanding of negation and stands as a robust benchmark for future research. Its design enables evaluation across diverse model families and settings, making it broadly applicable for studying semantic reasoning in LLMs.

Limitations

Thunder-NUBench is exclusively constructed in English, despite negation being a universal linguistic phenomenon demonstrated in diverse forms across languages. The syntactic and semantic expressions of negation may vary in other languages, meaning that our current findings may not generalize to multilingual or cross-lingual settings. In future work, we aim to extend the research to a broader range of languages to enable cross-linguistic evaluation of negation understanding in language models.

Although we built the dataset using two distinct sources (HoVer and Wikipedia summaries), both are derived from encyclopedic, formal domains, which may not fully represent the variety of sentence structures and informal language found in real-world use cases. Moreover, while all examples were systematically generated and reviewed, some bias may persist due to subjective decisions in the human review process. We attempt to mitigate this through cross-checking by an independent group of authors, but some residual bias may remain.

Thunder-NUBench primarily focuses on standard (sentence-level) negation and its distinction from local negation, contradiction, and paraphrase. Other important negation phenomena, such as double negation and negative polarity items (NPIs), are not directly addressed in this benchmark. Our current focus is on establishing a strong foundation for evaluating models' understanding of standard negation. However, we aim to expand the evaluation to a broader range of negation phenomena in future work.

Ethical Considerations

This work does not involve the use of crowdsourcing methods. Instead, all data included in the Thunder-NUBench benchmark has been carefully reviewed by the authors to ensure quality, relevance, and adherence to ethical standards. The datasets and tools used for training and evaluation are publicly available and used in compliance with their respective licenses.

When leveraging OpenAI's text generation models, we take additional care to avoid generating or including any content that is harmful, biased, or violates privacy. All generated examples are manually reviewed to meet ethical and safety standards. We ensure no personally identifiable information or offensive content is present in the final dataset.

The Thunder-NUBench dataset is released under the CC BY-NC-SA 4.0 license, ensuring transparency, reproducibility, and accessibility for future research. We believe our work contributes positively to developing trustworthy and interpretable language models.

Acknowledgments

We thank the anonymous reviewers and the meta-reviewer for their valuable feedback on this paper. We also sincerely thank Sungeun Hahm, Suyoung Park, Jongmin Kim, Yelim Ahn, Hyunji M. Park, Seorin Kim, and Jisoo Kim for their valuable contributions to the initial design of the negation dataset.

This work was partially supported by the National Research Foundation of Korea (NRF) under Grant No. RS-2023-00222663 (Center for Optimizing Hyperscale AI Models and Platforms), and by the Institute for Information and Communications Technology Promotion (IITP) under Grant No. 2018-0-00581 (CUDA Programming Environment for FPGA Clusters) and No. RS-2025-02304554 (Efficient and Scalable Framework for AI Heterogeneous Cluster Systems), all funded by the Ministry of Science and ICT (MSIT) of Korea. It was also partially supported by the Korea Health Industry Development Institute (KHIDI) under Grant No. RS-2025-25454559 (Frailty Risk Assessment and Intervention Leveraging Multimodal Intelligence for Networked Deployment in Community Care), funded by the Ministry of Health and Welfare (MOHW) of Korea. Additional support was provided by the BK21 Plus Program for Innovative Data Science Talent Education (Department of Data Science, Seoul National University, No. 5199990914569) and the BK21 FOUR Program for Intelligent Computing (Department of Computer Science and Engineering, Seoul National University, No. 4199990214639), both funded by the Ministry of Education (MOE) of Korea. This work was also partially supported by the Artificial Intelligence Industrial Convergence Cluster Development Project, funded by the MSIT and Gwangju Metropolitan City. Research facilities were provided by the Institute of Computer Technology (ICT) at Seoul National University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman,

Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Kumail Alhamoud, Shaden Alshammari, Yonglong Tian, Guohao Li, Philip HS Torr, Yoon Kim, and Marzyeh Ghassemi. 2025. Vision-language models do not understand negation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29612–29622.

Miriam Anschütz, Diego Miguel Lozano, and Georg Groh. 2023. This is not correct! negation-aware evaluation of language generation systems. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 163–175.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Miguel Ballesteros, Alberto Díaz, Virginia Francisco, Pablo Gervás, Jorge Carrillo de Albornoz, and Laura Plaza. 2012. Ucm-2: a rule-based approach to infer the scope of negation via dependency parsing. 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 288–293.

Valerio Basile, Johan Bos, Kilian Evang, Noortje Venhuizen, and 1 others. 2012. Ugroningen: Negation detection with discourse representation structures. In *Proceedings of 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*, pages 301–309. Association for Computational Linguistics.

Parminder Bhatia, E Busra Celikkaya, and Mohammed Khalilia. 2019. End-to-end joint entity extraction and negation detection for clinical text. In *International Workshop on Health Intelligence*, pages 139–148. Springer.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234.

Samuel 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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

Jiangjie Chen, Wei Shi, Ziquan Fu, Sijie Cheng, Lei Li, and Yanghua Xiao. 2023. Say what you mean! large language models speak too positively about negative commonsense knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9890–9908.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Steven Davis and Brendan S Gillon. 2004. *Semantics: A reader*. Oxford University Press.

Jorge Carrillo de Albornoz, Laura Plaza, Alberto Díaz, and Miguel Ballesteros. 2012. Ucm-i: A rule-based syntactic approach for resolving the scope of negation. 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 282–287.

Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. In *Proceedings of acl-08: Hlt*, pages 1039–1047.

Viviane Déprez, Susagna Tubau, Anne Cheylus, and M Teresa Espinal. 2015. Double negation in a negative concord language: An experimental investigation. *Lingua*, 163:75–107.

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.

Herbert B. Enderton. 2001. *A Mathematical Introduction to Logic*. Academic Press.

Orlando Espino and Ruth MJ Byrne. 2012. It is not the case that if you understand a conditional you know how to negate it. *Journal of Cognitive Psychology*, 24(3):329–334.

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.

Federico Fancellu, Adam Lopez, and Bonnie Webber. 2016. Neural networks for negation scope detection. In *Proceedings of the 54th annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 495–504.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Iker García-Ferrero, Begoña Altuna, Javier Álvez, Itziar Gonzalez-Dios, and German Rigau. 2023. This is not a dataset: A large negation benchmark to challenge large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8596–8615.

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.

Lila R Gleitman. 1965. Coordinating conjunctions in english. *Language*, 41(2):260–293.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Mareike Hartmann, Miryam de Lhoneux, Daniel Hershcovich, Yova Kementchedjhieva, Lukas Nielsen, Chen Qiu, and Anders Søgaard. 2021. A multilingual benchmark for probing negation-awareness with minimal pairs. In *CoNLL 2021-25th Conference on Computational Natural Language Learning*, pages 244–257. Association for Computational Linguistics.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Kees Hengeveld. 1986. Copular verbs in a functional grammar of spanish.

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 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.

Md Mosharaf Hossain, Dhivya Chinnappa, and Eduardo Blanco. 2022a. An analysis of negation in natural language understanding corpora. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 716–723.

Md Mosharaf Hossain, Luke Holman, Anusha Kakileti, Tiffany Kao, Nathan Brito, Aaron Mathews, and Eduardo Blanco. 2022b. A question-answer driven approach to reveal affirmative interpretations from verbal negations. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 490–503.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. 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)*.

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Myeongjun Jang, Frank Mtumbuka, and Thomas Lukasiewicz. 2022. Beyond distributional hypothesis: Let language models learn meaning-text correspondence. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2030–2042.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao,

Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Liwei Jiang, Antoine Bosselut, Chandra Bhagavatula, and Yejin Choi. 2021. "i'm not mad": Commonsense implications of negation and contradiction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4380–4397.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. 2020. Hover: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of lstms to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231.

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*, pages 7811–7818.

Aditya Khandelwal and Suraj Sawant. 2020. Negbert: A transfer learning approach for negation detection and scope resolution. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5739–5748.

David Kletz, Pascal Amsili, and Marie Candito. 2023. The self-contained negation test set. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 212–221.

George Lakoff. 1966. Criterion for verb phrase constituency.

Adrienne Lehrer and Keith Lehrer. 1982. Antonymy. *Linguistics and philosophy*, pages 483–501.

Alessandro Lenci, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. A comparative evaluation and analysis of three generations of distributional semantic models. *Language resources and evaluation*, 56(4):1269–1313.

Hao Li and Wei Lu. 2018. Learning with structured representations for negation scope extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 533–539.

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*.

Matti Miestamo. 2000. Towards a typology of standard negation. *Nordic journal of linguistics*, 23(1):65–88.

Matti Miestamo. 2007. Negation–an overview of typological research. *Language and linguistics compass*, 1(5):552–570.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

James Edward Miller and Jim Miller. 2011. *A critical introduction to syntax*. A&C Black.

Roser Morante and Walter Daelemans. 2009. A metalearning approach to processing the scope of negation. In *Proceedings of the thirteenth conference on computational natural language learning (CoNLL-2009)*, pages 21–29.

Roser Morante, Anthony Liekens, and Walter Daelemans. 2008. Learning the scope of negation in biomedical texts. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 715–724.

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.

Ha Thanh Nguyen, Randy Goebel, Francesca Toni, Kostas Stathis, and Ken Satoh. 2023. A negation detection assessment of gpts: analysis with the xnot360 dataset. *arXiv preprint arXiv:2306.16638*.

Ayana Niwa, Keisuke Nishiguchi, and Naoaki Okazaki. 2021. Predicting antonyms in context using bert. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 48–54.

OpenAI. 2025. Text generation and prompting. Accessed on May 16, 2025.

Keiron O'shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Junsung Park, Jungbeom Lee, Jongyoon Song, Sangwon Yu, Dahuin Jung, and Sungroh Yoon. 2025. Know" no"better: A data-driven approach for enhancing negation awareness in clip. *arXiv preprint arXiv:2501.10913*.

Vincent Quantmeyer, Pablo Mosteiro, and Albert Gatt. 2024. How and where does clip process negation? In *Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR)*, pages 59–72.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Abhilasha Ravichander, Matt Gardner, and Ana Marasović. 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.

Jonathon Read, Erik Velldal, Lilja Øvrelid, and Stephan Oepen. 2012. Uio1: Constituent-based discriminative ranking for negation resolution. 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 310–318.

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.

Laura Rimell, Amandla Mabona, Luana Bulat, and Douwe Kiela. 2017. Learning to negate adjectives with bilinear models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 71–78.

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of linguistics*, 20:33–53.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *COMMUNICATIONS OF THE ACM*, 64(9).

Zahra Sarabi and Eduardo Blanco. 2016. Understanding negation in positive terms using syntactic dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1108–1118.

Thijs Scheepers. 2017. Improving the compositionality of word embeddings. Master's thesis, Universiteit van Amsterdam, Science Park 904, Amsterdam, Netherlands, 11.

Jingyuan S She, Christopher Potts, Samuel 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.

Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. 2019. The performance of lstm and bilstm in forecasting time series. In *2019 IEEE International conference on big data (Big Data)*, pages 3285–3292. IEEE.

Rituraj Singh, Rahul Kumar, and Vivek Sridhar. 2023. Nlms: Augmenting negation in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13104–13116.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Charles Sutton, Andrew McCallum, and 1 others. 2012. An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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 4149–4158.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (* SEM 2023)*, pages 101–114.

Thinh Hung Truong, Julia Otmakhova, Timothy Baldwin, Trevor Cohn, Jey Han Lau, and Karin Verspoor. 2022. Not another negation benchmark: The nan-nli test suite for sub-clausal negation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 883–894.

Teemu Vahtola, Mathias Creutz, and Jörg Tiedemann. 2022. It is not easy to detect paraphrases: Analysing semantic similarity with antonyms and negation using the new semantoneg benchmark. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 249–262.

Ton Van der Wouden. 1996. Litotes and downward monotonicity. *Negation: a notion in focus*, 7:145.

Tereza Vrabcová, Marek Kadlčík, Petr Sojka, Michal Štefánik, and Michal Spiegel. 2025. Towards the roots of the negation problem: A multilingual NLI dataset and model scaling analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25537–25551, Suzhou, China. Association for Computational Linguistics.

RJ Wales and R Grieve. 1969. What is so difficult about negation? *Perception & Psychophysics*, 6(6):327–332.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Raffaella Zanuttini. 2001. Sentential negation. *The handbook of contemporary syntactic theory*, pages 511–535.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

A Typology of Contradiction

Contradictions in natural language can arise in diverse ways that go beyond simple negation. Following the typology of De Marneffe et al. (2008), contradictions can be grouped into seven categories: antonymy, explicit negation, numeric mismatch, factive/modal inconsistencies, structural reversals, lexical incompatibilities, and conflicts based on world knowledge. These categories reflect the fact that contradiction covers a broader semantic scope than negation alone. Table 7 summarizes these types with definitions and examples.

B Copular Verbs

Copular verbs, also known as linking verbs, are verbs that connect the subject of a sentence to a subject complement, which can be a noun, adjective, or other expression that describes or identifies the subject. Unlike action verbs, copular verbs do not express actions but rather states or conditions. The most common copular verb in English is "to be" in its various forms (am, is, are, was, were). Other examples include "seem," "appear," "become," "feel," "look," "sound," "taste," and "smell" when used to describe the subject's state (Hengeveld, 1986).

As discussed in Section 3.3, standard negation in this work targets the main predicate of a clause. For sentences with copular verbs, this means that the entire verb phrase, including the copular verb and its complement, is subject to negation. For example, in the sentence "She is a doctor," the main predicate is "is a doctor." Negating this sentence results in "She is not a doctor," where the negation applies to the entire predicate, not just the verb "is."

Negation of a verb phrase including a copular verb can be realized either syntactically (e.g., "is **not** an expert") or by replacing the complement with its complementary antonym (e.g., "is a **non-expert**"), both of which result in the reversal of the main predicate's truth value. Although such constructions may superficially appear to be non-verbal negation, especially when the complement is a noun or adjective, they are, in fact, instances of verbal negation, since the negation applies to the predicate as a whole.

C Negation of Implications

Negating implications presents challenges, as natural language intuitions often diverge from the rules of formal logic. Let's say there is a conditional statement, "If I study hard, I will pass the bar exam." Formally, let P denote "I study hard" and Q denote "I will pass the bar exam." In classical logic, the conditional "if P, Q" can be false only when P is true and Q is false. This implies that the negation of the conditional is "P and Neg(Q)" ("I study hard and I won't pass the bar exam,) while the conditional itself is equivalent to "Neg(P) or Q" ("I don't study hard or I will pass the exam) (Nguyen

Contradiction Type	Definition	Example
Antonym	Contradiction caused by opposing meanings of aligned words.	The policy was a **success**. → The policy was a **failure**.
Negation	One sentence explicitly negates a statement in the other.	She **attended** the meeting. → She **did not attend** the meeting.
Numeric	Inconsistent numbers, dates, or quantities in related statements.	Totally, **ten** people were injured. → Totally, **five** people were injured.
Factive/Modal	Conflict in implied facts or modal possibilities due to verbs or auxiliaries.	He **managed to** enter the building. → He **did not enter** the building.
Structure	Syntactic rearrangement or argument swapping causes contradiction.	**Alice** hired **Bob**. → **Bob** hired **Alice**.
Lexical	Contradiction through incompatible verbs or phrases, not strictly antonyms.	The manager **praised** her performance. → The manager **expressed disappointment in** her performance.
World Knowledge	Contradiction relies on common-sense or background knowledge.	The Eiffel Tower is in **Paris**. → The Eiffel Tower is in **Berlin**.

Table 7: Contradiction types from De Marneffe et al. (2008). Contradiction covers a broader scope than negation.

et al., 2023).

Psychological studies confirm that people often accept both "if P, Neg(Q)" ("If I study hard, I won't pass the exam.") and "if Neg(P), Q" ("If I don't study hard, I will pass the exam."). However, the former can be interpreted as "Neg(P) or Neg(Q)", and the latter "P or Q", both of which are not equivalent to the original statement's negation, "P and Neg(Q)". "if Neg(P), Neg(Q)" ("If I don't study hard, I won't pass the exam.") is not the correct negation as well, as it is equivalent to "P or Neg(Q)" (Espino and Byrne, 2012).

While humans often struggle to distinguish the correct negation of a conditional from invalid alternatives, the logical form is unambiguous. We therefore include conditional statements in our benchmark to test whether language models, like humans, are prone to intuitive but invalid interpretations, or whether they can correctly apply truth-functional reasoning.

Note that implications in natural language are not limited to the explicit "if P, Q" form, but may also appear with connectives such as *when*, *as long as*, or *unless*, which functionally convey conditional meaning and are treated under the same negation principle.

D Compound Sentences and Coordinating Conjunction

A compound sentence consists of two or more independent clauses joined by a coordinating conjunction. Each clause can stand alone, but they are combined to express related ideas (Gleitman, 1965).

Coordinating conjunctions connect elements of equal grammatical rank. The seven common ones in English are: *for, and, nor, but, or, yet, so* (often remembered as FANBOYS). Among these, *and*, *or*, and *but* are indisputably used to coordinate clauses. The others can be ambiguous or function in non-coordinating roles(e.g., indicating cause or result rather than logical structure). These are the examples using *and*, *or*, and *but* to connect sentences equally.

- "She studied hard, **and** she passed the exam."
- "I wanted to go, **but** it was raining."
- "You can call me, **or** you can send an email."

We consider only the coordinating conjunctions *and*, *or*, and *but* as indicators of compound sentences, in which two or more independent clauses are equally connected. Although *but* introduces a contrast semantically, in terms of logical structure, it functions as a conjunction equivalent to *and*; therefore, its negation follows the same principle.

E Local Negation Constructions Excluded

In constructing the Thunder-NUBench dataset, we consider various types of local (i.e., subclausal)

negation, where negation applies to a phrase or constituent rather than the main predicate. However, several constructions are excluded due to their semantic ambiguity, syntactic irregularity, or misalignment with the benchmark's focus on verbal negation.

Infinitive Phrase Negation. Infinitive phrases (e.g., "to go") can be negated with "not" (e.g., "not to go" or "to not go"). Unlike the clause-level structures that define our local negation category, infinitive phrases are not full clauses but simply part of a verb phrase, making them less compatible with our definition. Moreover, although grammatically correct, this construction is relatively rare and sounds awkward depending on the context.

- **Original**: George wants to go to the park.
- **Negated (infinitive)**: George wants not to go / George wants to not go to the park.

For these reasons, we exclude infinitive phrase negation from the benchmark.

Appositive Clause Negation. Appositive clauses are noun phrases that provide descriptive clarification. Attempting to negate an appositive typically involves lexical replacement rather than syntactic negation.

- **Original:** My brother, a talented musician, plays the guitar.
- **Negated (appositive):** My brother, not a talented musician, plays the guitar.

Such changes alter descriptive content rather than reversing the meaning of the predicate, and often fall into the domain of contradiction. Accordingly, they are excluded from the dataset.

Prepositional Phrase Negation. Negating a prepositional phrase often involves replacing the preposition with its antonym (e.g., "with" $\rightarrow$ "without", "in" $\rightarrow$ "outside"), which results in a sentence that differs in content, rather than reversing the meaning of the predicate.

- **Original**: She went to the park with her bird.
- **Negated (preposition)**: She went to the park without her bird.

Since such modifications do not negate the verb but instead change the nature of an adjunct or argument, they fall outside the scope of standard negation or local negation in this work and are excluded.

In all of the above cases, the negation does not target the whole verb phrase but rather peripheral elements within the sentence. As the ThunderNUBench is designed to evaluate verbal negation, these local or phrase-level forms of negation were intentionally left out.

F Double Negation

Double negation refers to the use of two forms of grammatical negation within a single sentence. In standard English, only one negative form should be present in a subject-predicate construction; the presence of two negatives is generally considered nonstandard and often results in an unintended meaning. For example, while "He's going nowhere" is correct, "He's not going nowhere" is ungrammatical. Another example is "I won't bake no cake," which combines verb negation ("won't") with object negation ("no cake"), resulting in a grammatically incorrect construction (Déprez et al., 2015).

In English, certain double negation constructions convey affirmative meanings rather than intensifying negation, effectively paraphrasing the original positive statement (e.g., $\neg\neg p \approx p$) (Van der Wouden, 1996). This rhetorical device, known as litotes, often manifests in expressions such as "not bad," implying "good," or "not unhappy," implying "happy." Leveraging this phenomenon, we have generated paraphrase candidates for our dataset using such double negation patterns. For example,

- **Sentence:** His characteristic style fuses samba, funk, rock and bossa nova with lyrics that blend humor and satire with often esoteric subject matter.
  Double Negation: His characteristic style **does not fail to fuse** samba, funk, rock, and bossa nova with lyrics that blend humor and satire with often esoteric topics.
- **Sentence:** It covers a broad range of fields, including the humanities, social sciences, exact sciences, applied sciences, and life sciences.
  Double Negation: It **does not exclude** a broad range of fields, including the humanities, social sciences, exact sciences, applied sciences, and life sciences.
- **Sentence:** Sanders was honoured to meet with many world dignitaries and representatives of UNESCO member nations, and delighted when delegates from UNESCO, visited Toowoomba in 2018 in return.
  Double Negation: Sanders **was not unhappy** to meet with many world dignitaries and representatives of UNESCO member nations, and not displeased when delegates from UNESCO visited Toowoomba in 2018 in return.

However, upon closer examination, these paraphrase candidates do not always preserve the exact meaning of the original sentence. The antonyms used (e.g., "exclude" for "cover," "unhappy" for "honoured") are not always true complementary antonyms, which does not effectively negate the meaning. Moreover, the litotes construction ("does not fail to fuse") tends to add an emphatic nuance, rather than being a perfect semantic equivalent. Therefore, the boundary between paraphrasing and double negation is ambiguous, and their relationship requires more careful analysis. Given these issues, and because our primary focus is on standard negation, we ultimately decide to exclude double negation constructions as paraphrase candidates from our dataset.

G HoVer Dataset

The HoVer (**Ho**ppy **Ver**ification) dataset is developed for the tasks of multi-hop evidence retrieval and factual claim verification. In HoVer, each claim requires supporting evidence that spans multiple English Wikipedia articles to determine whether the claim is substantiated or not. The dataset is distributed under a CC BY-SA 4.0 License, and it can be accessed via its official homepage[2]. Table 8 offers an overview of the dataset's structure. The data is split into training, validation, and test sets, containing 18,171, 4,000, and 4,000 examples respectively.

HoVer is constructed on top of the HotpotQA dataset, which is designed to evaluate multi-hop reasoning in question answering. HotpotQA itself is a large-scale collection of Wikipedia-based QA pairs created to address the limitations of prior QA datasets, which often fail to require complex reasoning or explanatory answers (Yang et al., 2018). The construction of HoVer involves rewriting HotpotQA question-answer pairs into claim statements, which are then validated and labeled by annotators. Claims are extended to require multi-hop evidence from up to four Wikipedia articles and are systematically modified to increase complexity. Final labels are assigned as `SUPPORTED` or `NOT-SUPPORTED` (Jiang et al., 2020).

H Wikipedia Summary Dataset

The Wikipedia Summary Dataset contains the titles and introductory summaries of English Wikipedia articles, extracted in September 2017. A summary or introduction in this context refers to the content from the article title up to the content outline (i.e., before the first section heading). The dataset was originally released via GitHub[3], but is now accessible through the Hugging Face Hub[4]. The dataset license is not explicitly mentioned, but as the original Wikipedia data is distributed under the CC BY-SA 4.0, it is assumed that the dataset would be distributed under the same license. For licensing details, refer to the Wikimedia Terms of Use [5]. Table 9 offers an overview of the dataset's structure. The dataset comprises approximately 430,000 articles, only providing the training set (Scheepers, 2017).

I Human Review Protocol

To ensure high-quality data construction, we implement a rigorous quality control protocol that combines generation, independent review, and iterative consensus building. The process involves the following key steps:

- **Task allocation and independence.** Authors are assigned distinct portions of the dataset, but no author is permitted to review the data they have generated. This ensures that each instance is subject to at least one independent review.
- **Sequential authoring across choices.** For the multiple-choice dataset, construction proceeds in four stages: standard negation, local negation, contradiction, and paraphrase. At each stage, different authors are responsible for creating the new option, while reviewers who have not authored that option perform the verification.
- **Cross-checking and layered review.** Each newly created option is reviewed by at least one other author, and reviewers also revisit earlier options in the same instance. For example, when reviewing the paraphrased sentence, the reviewer also checks that standard negation, local negation, and contradiction sentences are correct. As a result, every instance undergoes multiple rounds of verification across stages, such that all authors ultimately examine data they have not created themselves.

[2] https://hover-nlp.github.io/

[3] https://github.com/tscheepers/Wikipedia-Summary-Dataset

[4] https://huggingface.co/datasets/jordiclive/wikipedia-summary-dataset

[5] https://foundation.wikimedia.org/wiki/Policy:Terms_of_Use

Column	Detail	Example
id	Unique claim identifier	0
uid	User/annotator identifier	330ca632-e83f-4011-b11b-0d0158145036
claim	The statement to be verified, often requiring multi-article evidence	Skagen Painter Peder Severin Krøyer favored naturalism along with Theodor Esbern Philipsen and the artist Ossian Elgström studied with in the early 1900s.
supporting_facts	List of Wikipedia article titles and sentence indices providing evidence	[{ "key": "Kristian Zahrtmann", "value": 0 }, { "key": "Kristian Zahrtmann", "value": 1 }, { "key": "Peder Severin Krøyer", "value": 1 }, { "key": "Ossian Elgström", "value": 2 }]
label	Whether the claim is supported	1: SUPPORTED or 0: NOT_SUPPORTED
num_hops	Number of articles required for verification	2∼4
hpqa_id	Reference to the original HotpotQA pair	5ab7a86d5542995dae37e986

Table 8: Details of HoVer dataset structure with examples.

Column	Detail	Example
title	Article title from Wikipedia.	Alain Connes
description	A brief description or category for the article (when available).	French mathematician
summary	The extracted summary or introduction section of the article, typically more concise than the full text.	Alain Connes (; born 1 April 1947) is a French mathematician...
full_text	The complete article text (when included), encompassing the full body of the Wikipedia page.	Alain Connes (; born 1 April 1947) is a French mathematician...
__index_level_0__	Index number for each entry in the dataset.	3

Table 9: Details of Wikipedia Summary dataset structure with examples.

- **Guideline refinement and retroactive correction.** Generation and reviewing guidelines are continuously updated based on discussion of ambiguous or problematic cases. Whenever the guidelines changes, all previously created data are revisited to ensure compliance, promoting consistency across the dataset.
- **Consensus and adjudication.** Disagreements are discussed in weekly meetings and, if necessary, adjudicated by a lead reviewer, ensuring that no instance remains unresolved.

Overall, this iterative and layered procedure ensures that every instance in the multiple-choice dataset is independently reviewed across multiple stages, leading to stable guidelines and a consistent dataset.

Edit-based reliability and quality control statistics. Because our review process is revision-based rather than label-based, standard inter-annotator agreement metrics (e.g., Cohen's κ) are not directly applicable. Instead of assigning categorical labels, reviewers have directly edited or deleted problematic instances to enforce the dataset guidelines.

We therefore report edit-based quality control statistics as a proxy for annotation reliability, including (i) the number of instances escalated to adjudication meetings and (ii) the number of instances that required edits after cross-checking. All cases requiring edits have been discussed in meetings and resolved by consensus, ensuring consistency across the dataset.

Subset		# items	# Brought to meeting	# Edited	Edit rate
Sentence-Negation	Standard Negation	3,992	469	72	1.80%
Multiple Choice	Standard/ Local Negation	1,102	126	34	3.09%
	Contradiction/ Paraphrase	1,102	409	117	10.62%
	Implication (add-on)	201	49	16	7.96%

Table 10: Edits include rewriting or deleting instances that violate guidelines.

Overall, the relatively low edit rates after cross-checking indicate that most instances satisfied the guidelines upon independent review, while meeting-based adjudication played a critical role in resolving ambiguous or complex cases.

J Detailed Principles and Examples of the Thunder-NUBench

While the main text already defines the core notions of standard and local negation (Section 3) and explains how they are applied throughout dataset construction (Section 4), here we provide more detailed illustrations.

Paraphrasing before Negation. Before negating, the main verb or other components may be paraphrased with synonyms, provided that the sentence's tense, structure, and meaning remain strictly equivalent before applying standard negation. Authors refer to the Merriam-Webster Thesaurus [6]. For example,

- **Original Sentence**: Toumour is a village and rural commune in Niger **located near** the Niger–Nigeria **border**.
  - **Paraphrased Sentence**: Toumour is a village and rural commune in Niger **that is found close to** the Niger–Nigeria **boundary**.
    → **Standard Negation after Paraphrase**: Toumour **isn't** a village and rural commune in Niger that is found close to the Niger–Nigeria boundary.
  - **Explanation**: In this example, the participle clause "located near the Niger–Nigeria border" is rephrased as a relative clause "that is found close to the Niger–Nigeria boundary." Since both constructions serve as modifiers and preserve the same semantic role, we treat them as equivalent in meaning for the purpose of standard negation.
- **Original Sentence**: The armed forces **said** Boko Haram **attacked** their military post on March 15, 2020, which they responded to by repelling the attack, killing 50 insurgents.
  - **Paraphrased Sentence**: The armed forces **stated** that Boko Haram **assaulted** their military post on March 15, 2020, which they responded to by repelling the attack, killing 50 insurgents.
    → **Standard Negation after Paraphrase**: The armed forces **didn't state** that Boko Haram assaulted their military post on March 15, 2020, which they responded to by repelling the attack, killing 50 insurgents.
  - **Explanation**: In this example, the reporting verb "said" is paraphrased as "stated," and the verb "attacked" is replaced with the synonym "assaulted." These substitutions preserve the original tense and meaning, allowing standard negation to be applied without altering the semantic content of the sentence.

Negation of Simple Sentences. For simple, declarative sentences, standard negation is achieved by inserting "not" after the auxiliary or main verb, or by replacing the predicate with its complementary antonym. For example, "She is happy." → "She is not happy."; "The room is occupied." → "The room is unoccupied."

Negation in Compound Sentences. When multiple clauses or propositions are coordinated (e.g., with "and", "or", "but"), standard negation is logically applied, governed by De Morgan's laws. Here, "but" is treated as a coordinating conjunction equivalent to "and" in terms of logical structure, so its negation follows the same principle.

- Conjunction "P and/but Q": the negation is "Neg(P) or Neg(Q)".
- Disjunction "P or Q": the negation is "Neg(P) and Neg(Q)".

For example, "He passed the test and received an award." is negated as "He did not pass the test or did not receive an award."

When application of logical negation produces unnatural language, sentences may be split or slightly rephrased for fluency, provided logical meaning is preserved. For example,

- **Original:** "He finished the report and submitted the assignment."
- **Standard Negation:** "He did not finish the report or did not submit the assignment."
- **Standard Negation, but Splitted:** "He did not finish the report. Or, he did not submit the assignment."

Coordinated Elements in the Sentence. When a sentence contains coordinated elements (such as subjects, objects, or predicates connected by "and"

[6] https://www.merriam-webster.com/

or "or"), standard negation typically follows logical principles derived from De Morgan's Laws. However, whether logical negation applies to each individual component or to the entire predicate as a whole depends on whether the coordination expresses multiple independent propositions or a single collective event.

- If the coordination introduces semantically distinct propositions, that is, each conjunct could form a complete sentence on its own, negation must be applied to each proposition individually. For example, "My sister and I studied hard."
  This sentence can be interpreted as: "My sister studied hard and I studied hard."
  Therefore, the correct standard negation is: "My sister did not study hard, or I did not study hard."

- Conversely, if the coordination connects elements that jointly participate in a single action or state (e.g., a shared subject or a collective predicate), then the sentence is treated as a simple clause, and the predicate as a whole is negated. Logical decomposition is not appropriate. For example, "My sister and I share clothes."
  This expresses a single collective action involving both participants.
  Therefore, the correct standard negation is: "My sister and I do not share clothes."
  (NOT: "My sister does not share clothes, or I do not share clothes.")

- This distinction is crucial: even if two noun phrases are coordinated, if the sentence semantically decomposes into separate atomic propositions, standard negation must apply to each atomic proposition. Otherwise, it applies to the whole predicate as one unit.

- Other examples of semantically collective predicates where logical splitting is not appropriate include: "be the same", "have in common", "do something together", "combine", "unite", etc. These describe inherently joint or relational properties, not independent propositions. For example, "Clarence Brown and Peter Glenville are from the same country." should be negated as "Clarence Brown and Peter Glenville are not from the same country."

Use of Antonyms. When replacing predicates with antonyms in standard negation, only complementary antonyms are appropriate, as they provide a clear binary opposition, ensuring logical consistency of negation. Gradable and relational antonyms are unsuitable for standard negation because their antonyms do not represent the logical complement of the original predicate. In other words, replacing a predicate p with its antonym does not produce $\neg p$ in a truth-conditional sense.

Specifically, unlike complementary antonyms, which form mutually exclusive pairs (i.e., $p \cup \neg p = U$ and $p \cap \neg p = \emptyset$), gradable and relational antonyms do not partition the meaning space cleanly, and thus fail to reverse the truth value reliably.

- **Complementary Antonyms**: Also called binary/contradictory antonyms. These antonyms represent mutually exclusive pairs with no intermediate states. The presence of one implies the absence of the other. Examples include:
  - alive / dead
  - true / false
  - present / absent
  - occupied / vacant

  Using complementary antonyms in negation ensures a direct and unambiguous reversal of the original proposition's truth value.

- **Gradable Antonyms**: These antonyms exist on a continuum and allow for varying degrees between the two extremes. Negating one does not necessarily affirm the other. Examples include:
  - hot / cold
  - happy / sad
  - tall / short
  - young / old

  Due to their scalar nature, gradable antonyms are inappropriate for standard negation, as they do not provide a definitive binary opposition.

- **Relational Antonyms**: Also known as converse antonyms, these pairs describe a reciprocal relationship where one implies the existence of the other. Examples include:
  - parent / child

- teacher / student
- buy / sell
- employer / employee

Relational antonyms are context-dependent and do not represent direct opposites in a binary sense, making them unsuitable for standard negation purposes.

General Principles of Standard Negation.

- The negated sentence must preserve all elements (subject, tense, objects, adjuncts, etc.) of the original, except for the truth value of the main predicate.
- When naturalness and logical negation conflict, logical correctness takes priority, but minimal rephrasing is allowed for fluency.
- If the negated clause creates a contradiction with other parts of the sentence, the contradictory clause must be removed. For example, the standard negation of the sentence "While the spatial size of the entire universe is unknown, it is possible to measure the size of the observable universe, which is approximately 93 billion light-years in diameter."
  will be "While the spatial size of the entire universe is unknown, it isn't possible to measure the size of the observable universe."
  The relative clause must be removed because its content directly contradicts the negated main clause.

Common Negation Errors and Corrections.

- **Original sentence**: His characteristic style fuses samba, funk, rock **and** bossa nova with lyrics that blend humor and satire with often esoteric subject matter.
  - **Incorrect negation**: His distinctive style **doesn't fuse** samba, funk, rock **or** bossa nova with lyrics that blend humor and satire with often esoteric subject matter.
  - **Correct negation**: His distinctive style **doesn't fuse** samba, funk, rock **and** bossa nova with lyrics that blend humor and satire with often esoteric subject matter.
  - **Explanation**: The verb "fuse" implies a combination of all listed elements. "and" must be preserved.
- **Original sentence**: The mascot of Avon Center School is the "Koalaty Kid," **while** the mascot at Prairieview **is** an eagle **and** the mascot at Woodview **is** an owl.
  - **Incorrect negation**: Avon Center School's mascot is not the "Koalaty Kid," Prairieview's mascot **is not** an eagle, **or** Woodview's mascot **is not** an owl.
  - **Correct negation**: Avon Center School's mascot is not the "Koalaty Kid," **while** the mascot at Prairieview **is** an eagle **and** the mascot at Woodview **is** an owl.
  - **Explanation**: Two clauses connected by while are not coordinated propositions (as with *and* or *or*), but instead express contrastive information. Therefore, applying logical negation across both clauses is incorrect. Negation should apply only to the main clause (here, the first statement), while the contrasting clause remains affirmative.

K Code for Data Construction

K.1 Sentence-Negation Pair Dataset

To construct the sentence-negation pair dataset, we begin by randomly sampling sentences labeled as "supported facts" from the HoVer dataset. Since the original data often contains grammatical errors, we utilize OpenAI's API (OpenAI, 2025) to automatically correct these issues. In cases where the selected text consists of multiple sentences, we merge or split them as needed to ensure that each example is a single sentence, aligning with our sentence-level task objective.

We select different model versions depending on the complexity of each task. For sentence merging, which demands nuanced contextual understanding and complex syntax, we use `GPT-4`. For grammar correction, where edits are more straightforward, `GPT-3.5` is sufficient.

```
def grammar_fix(claim):
    messages = [{"role": "system", "content": "Fix
     grammatical errors."},
    {"role": "user", "content": f"If there are
     errors, please fix the sentence: {claim} \n If
     there aren't, return the original sentence.
     Provide only the resulting sentence without any
      additional explanation or introduction."}]
    response = client.chat.completions.create(model=
     "gpt-3.5-turbo", messages=messages)
    fixed_text = response.choices[0].message.content
     .strip()
    return fixed_text
```

Listing 1: Fixing Grammar with OpenAI API.

```
def merge_sentences_with_gpt(claim):
    messages = [{"role": "system", "content": "Merge
      sentences into a single one."},
    {"role": "user", "content": f"Merge these
     sentences: {claim} \n Provide only the
     resulting sentence without any additional
     explanation or introduction."}]
    response = client.chat.completions.create(model=
     "gpt-4-turbo-preview", messages=messages)
    merged_text = response.choices[0].message.
     content.strip()
    return merged_text
```

Listing 2: Merging Sentences with OpenAI API.

K.2 Multiple Choice Dataset

To construct the multiple-choice dataset, we first segment the "summary" column of the Wikipedia Summary dataset, which often contains multiple sentences in a single entry, into individual sentences. To focus on the challenges of negation in complex sentences, we filter out sentences that are too short. This process is done with Python code.

Since conditional sentences (e.g., "If P, Q") are rarely present in the Wikipedia summary dataset, we adopt a two-step approach: (1) prompting the model to generate conditional variants from given sentences (using OpenAI API, `GPT-4o-mini`), and (2) manually filtering or lightly editing the results to obtain valid conditionals.

Subsequently, we automatically generate contradictions and paraphrases for each sentence via the OpenAI API (`GPT-4o`) as well, followed by human review. The following scripts illustrate the procedures.

```
import pandas as pd
import re
from datasets import load_dataset
import random

df = pd.DataFrame(load_dataset("jordiclive/wikipedia
     -summary-dataset")['train'].shuffle(seed=42).
     select(range(10000))
df = df.drop(columns=['full_text'])

def split_into_sentences(text):
    sentences = re.split(r'(?<=[.!?]) +', text)
    return sentences

df['sentence'] = df['summary'].apply(
     split_into_sentences)
df = df.explode('sentence')
df = df[df['sentence'].apply(lambda x: len(x.split()
     ) >= 30)]
df = df.reset_index(drop=True)
df.to_csv("file/wikipedia_summary_sentences.csv",
     index=False)
```

Listing 3: Sentence extraction and preprocessing from Wikipedia summaries.

```
def generate_conditionals(sentence):
    prompt = f"""
    Based on the sentence below, write a conditional
      sentence that uses the main topic of the
     sentence.
    The conditional sentence should express a
     hypothetical situation or cause-effect
     relationship related to the topic. It can be
     slightly complex in structure.
    For example:
    - If it rains tomorrow, I will stay home.
    Sentence:
    '{sentence}'
    """

    completion = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=[
            {"role": "system", "content": "You are a
      helpful assistant that specializes in
     generating conditional sentences."},
            {"role": "user", "content": prompt}
        ]
    )

    return completion.choices[0].message.content
```

Listing 4: Conditionals sentence generation.

```
def generate_contradiction(sentence):
    prompt = f"""
    You will be given a sentence. Generate a
     contradictory sentence that directly conflicts
     with the original sentence without using
     standard negation.

    Definitions:
    - Standard negation: Directly negating the main
     verb or using words like 'not', 'no', 'never',
     or negative contractions such as \"isn't\", \"
     doesn't\", or \"can't\".
    - Contradiction: A sentence that logically
     conflicts with the original statement. The
     contradiction must be such that both sentences
     cannot logically be true at the same time under
      any circumstances.

    Important:
    - Do not change the main verb from the original
     sentence.
    - Do not use 'never' or other negative words to
     form the contradiction.
    - Ensure the contradicted sentence logically
     excludes the possibility of the original
     sentence being true simultaneously.

    Examples:
    Original sentence: \"The tallest student won the
      award.\"
    Contradicted sentence: \"The shortest student
     won the award.\"

    Original sentence: \"The room was completely
     dark.\"
    Contradicted sentence: \"The room was brightly
     lit.\"

    Original sentence: \"The event took place in the
      morning.\"
    Contradicted sentence: \"The event took place in
      the evening.\"

    Original sentence: \"All people are dying.\"
    Contradicted sentence: \"Some people are dying
     .\"

    Now, generate a contradictory sentence without
     standard negation, without changing the main
     verb, and ensuring the two sentences are
     logically incompatible, for the following:

    Original sentence: \"{sentence}\"

    Contradicted sentence:
    """
```

```
    completion = client.chat.completions.create(
        model="gpt-4o",
        messages=[
            {"role": "system", "content": "You are a
     helpful assistant tasked with generating
     logical contradictions. Do not use negation to
     make contradiction."},
            {"role": "user", "content": prompt}
        ]
    )

    return completion.choices[0].message.content
```

Listing 5: Contradiction generation.

```
def generate_paraphrase(sentence):
    prompt = f"""
        Paraphrase the following sentence using
     synonyms or slight structural variations
     without changing its meaning.
        Do not add or remove any main verbs. Keep
     the original intent of the sentence intact.

        Original sentence: "{sentence}"

        Paraphrased sentence:
        """

    completion = client.chat.completions.create(
        model="gpt-4o",
        messages=[
            {"role": "system", "content": "You are a
     helpful assistant skilled at generating
     paraphrases while keeping the meaning of
     sentences unchanged."},
            {
                "role": "user",
                "content": prompt
            }
        ]
    )

    return completion.choices[0].message.content
```

Listing 6: Paraphrase generation.

L Thunder-NUBench Dataset Structure

Thunder-NUBench consists of two subsets: a sentence-negation pair dataset for supervised fine-tuning and a multiple-choice dataset for evaluation. Both datasets are built on English text and reviewed by authors following strict guidelines.

L.1 Sentence-Negation Pair Dataset

This subset contains pairs of affirmative and corresponding standard negation sentences. It includes the following fields:

- `index`: the index of the data.
- `premise`: the original sentence.
- `hypothesis`: its logically negated form.

L.2 Multiple-Choice Dataset

This evaluation set presents each original sentence with four candidate transformations.

- `wikipedia_index`: the original index of the Wikipedia Summary dataset.
- `index`: the index of the data.
- `sentence`: the original sentence.
- `choice1`: standard negation (correct answer).
- `choice2`: local negation (subclausal negation).
- `choice2_type`: specifies the type of local negation.
- `choice2_element`: a short description of the phrase or clause that was negated (e.g., "being built", "which crashed").
- `choice3`: contradiction (non-negated, semantically incompatible).
- `choice4`: paraphrase (semantically equivalent).

The details of `choice2_type` and distribution on demonstration and test sets are described in Table 11. It follows the definition in Table 3.

In addition to the Wikipedia Summary dataset, we supplement the evaluation set with conditionals (e.g., If P, Q) by manually searching Wikipedia articles where such constructions are more likely to occur (e.g., Newton's laws of motion). Among the 100 conditional sentences included across demonstration and test sets, 20 are collected through manual search (marked with indices beginning with "S" in `wikipedia_index`). Meanwhile, the remaining 80 are sampled from the Wikipedia Summary dataset and converted into conditional form using the script in Listing 4 (Appendix K).

M Prompt Templates Used in Evaluation

We design and employ two instruction formats for in-context learning. These instructions differ in the level of guidance they provide, ranging from a task definition to a detailed procedural instruction. Specifically, we use the following two instruction styles. These instruction formats correspond to the `task instruction` component shown in Figure 1.

1. **Definition instruction**: A concise definition of standard negation, which specifies the target operation,
   `"Standard negation is sentential negation that reverses the truth value of the sentence by negating the main predicate(s) of the main clause(s). Keep the rest of the sentence content unchanged."`

2. **Detailed instruction**: A step-by-step instruction that presents standard negation as an explicit procedure. This prompt describes how to identify the main clause and its main predicate, preserve other sentence elements, apply syntactic negation or complementary antonyms where appropriate, and negate complex sentences by reversing their logical struc-

choice2_type	Definition	Demonstration Set	Test Set
relative_part	negation inside relative clauses (e.g., "who did not attend. . . ").	12	312
pp_part	negation in participle clauses (e.g., "not walking through the park. . . ").	12	308
adverb_part	negation in adverbial clauses (e.g., "because it was not raining").	12	310
compound_part	negation applied to one clause within a compound sentence.	12	294
non-applicable	used when the sentence structure does not support a valid local negation variant under our definition.	2	37
	Total	50	1,261

Table 11: Choice 2 Types and Distributions.

ture. The full prompt format is shown in Listing 7.

```
Standard negation reverses the truth value of
     the main predicate in the main clause
     while keeping all other elements of the
     main clause unchanged.
Do not negate subordinate clauses or modify
     other parts of the sentence.

To do this:
1) Identify the main clause and its main verb
     (main predicate). Ignore subordinate
     clauses.
2) Preserve all other main-clause content.
3) Insert a negative particle such as "not"
     into the main verb, or replace it with a
     complementary antonym only if it forms an
      absolute binary (e.g., alive/dead, true/
     false, possible/impossible).
4) If the sentence contains multiple
     propositions connected by logical
     operators (e.g., and, or, conditional
     constructions), negate it in a way that
     reverses the entire proposition (e.g., A
     and B -> not A or not B; If A then B -> A
      and not B).
```

Listing 7: Detailed instruction format.

Below, we show the prompt formats used for completion-based and option-selection evaluations, including prompt–response structures for each evaluation setting.

```
Prompt] "{instruction format}

Generate the standard negation of the given
     sentence.
Sentence: Chromosome 2 is the second-largest human
     chromosome, spanning more than 242 million
     base pairs and representing almost eight
     percent of the total DNA in human cells.
Negation:"

Response] "Chromosome 2 isn't the second-largest
     human chromosome, which measures more than 242
      million base pairs and represents almost
     eight percent of the entire DNA in human cells
     ."
```

Listing 8: Completion-based Format.

```
Prompt] "Given the following instruction and
     candidate answers, choose the single best
     answer.

{instruction format}

Generate the standard negation of the given
     sentence.
Sentence: Chromosome 2 is the second-largest human
     chromosome, spanning more than 242 million
     base pairs and representing almost eight
     percent of the total DNA in human cells.

A. Chromosome 2 isn't the second-largest human
     chromosome, which measures more than 242
     million base pairs and represents almost eight
      percent of the entire DNA in human cells.
B. Chromosome 2 is the second-largest human
     chromosome, which doesn't span more than 242
     million base pairs or represent nearly eight
     percent of the whole DNA in human cells.
C. Chromosome 2 is the smallest human chromosome,
     spanning fewer than 50 million base pairs and
     representing less than two percent of the
     total DNA in human cells.
D. Chromosome 2 is the second-biggest human
     chromosome, with over 242 million base pairs,
     making up nearly 8% of all DNA in human cells.

Your response should be one of A, B, C, D.
Only output the letter.
Answer:"

Response] "A"
```

Listing 9: Option-selection Format.

N Evaluation Setup

We use 8 NVIDIA GeForce RTX 3090 GPUs with 24GB of memory and CUDA 12.4 for all fine-tuning and evaluation.

N.1 Models

We evaluate a diverse set of pretrained and instruction-tuned models across multiple model families and scales, ranging from about a billion to 14 billion parameters. Our experiments include models from Gemma (Team et al., 2024),

Qwen (Yang et al., 2025), LLaMA (Grattafiori et al., 2024), and Mistral families (Jiang et al., 2023), as well as API-based models from OpenAI (Achiam et al., 2023; Hurst et al., 2024) and Anthropic (Anthropic, 2024). API models are evaluated only in the option-selection setting, as they do not provide access to token-level log probabilities required for completion-based evaluation.

- **Gemma models:**
  - gemma-2-2b, gemma-2-2b-it
  - gemma-2-9b, gemma-2-9b-it
- **Qwen models:**
  - Qwen3-0.6B-Base, Qwen3-0.6B
  - Qwen3-1.7B-Base, Qwen3-1.7B
  - Qwen3-4B-Base, Qwen3-4B
  - Qwen3-8B-Base, Qwen3-8B
  - Qwen3-14B-Base, Qwen3-14B
  - Qwen3-32B
- **LLaMA models:**
  - Llama-3.1-8B, Llama-3.1-8B-Instruct
  - Llama-3.2-1B, Llama-3.2-1B-Instruct
  - Llama-3.2-3B, Llama-3.2-3B-Instruct
- **Mistral models:**
  - Mistral-7B-v0.3, Mistral-7B-Instruct-v0.3
  - Mistral-Nemo-Base-2407 (12B size), Mistral-Nemo-Instruct-2407 (12B size)
  - Mistral-Small-24B-Base-2501, Mistral-Small-24B-Instruct-2501
- **API models:**
  - GPT: GPT-4o mini, GPT-4o, GPT-4.1 mini, GPT-4.1
  - Claude: Haiku 4.5, Sonnet 4.5

N.2 Zero-shot and few-shot evaluation

For each model, we evaluate performance in both zero-shot and few-shot settings using the Language Model Evaluation Harness (Gao et al., 2024). In the few-shot scenario, we use examples from the demonstration set as in-context demonstrations. Results are averaged over five random seeds (42, 1234, 3000, 5000, and 7000) and are reported for one, five, and ten examples from the demonstration set (1-shot, 5-shot, and 10-shot). We present the performance results on the test set for each model and prompt configuration.

N.3 Supervised fine-tuning

We conduct Supervised Fine-Tuning (SFT) using the LLaMA-Factory framework (Zheng et al., 2024) on the Sentence-Negation Pair dataset from Thunder-NUBench. The dataset is formatted in the Alpaca instruction style (Taori et al., 2023). To achieve parameter-efficient training, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2022) with a rank of 8, targeting all linear layers. The fine-tuning process is carried out for three epochs, using a batch size of 1, a gradient accumulation step of 8, cosine learning rate scheduling, and bfloat16 precision. We apply supervised fine-tuning only to models with fewer than 10B parameters, in order to focus on analyzing the effects of SFT on relatively smaller models, where instruction tuning is expected to have a more pronounced impact. After the SFT, we evaluate the model's zero-shot performance to directly assess its ability to generalize from instruction tuning without being influenced by in-context examples. It is important to note that API models are not fine-tuned, as they are not compatible with the LLaMA-Factory framework.

O Total Model Performance on Thunder-NUBench

This section provides the full set of results on Thunder-NUBench for all evaluated models and training settings. Following the prompt templates in Appendix M, we report results separately for the definition instruction and detailed instruction styles.

For each instruction style, models are evaluated under three conditions: (1) zero-shot, (2) few-shot with 1, 5, and 10 in-context demonstrations (averaged across random seeds), and (3) supervised fine-tuning (SFT). For SFT, models are evaluated in the zero-shot setting to isolate the effect of task-specific training without the influence of in-context examples.

Tables 14 and 15 report the results of open-weight models under the definition and detailed instruction styles, respectively. Tables 16 and 17 present the corresponding results for API models under the same instruction settings.

P General Benchmark Performance after SFT

To assess whether supervised fine-tuning (SFT) on Thunder-NUBench affects performance on broader natural language understanding tasks, we evaluate models on six widely used benchmarks: ARC-Challenge, ARC-Easy, GSM8K, HellaSwag, OpenBookQA, and WinoGrande (Clark et al., 2018; Cobbe et al., 2021; Zellers et al., 2019; Mihaylov et al., 2018; Sakaguchi et al., 2021). These datasets

cover a diverse range of domains, including commonsense reasoning, scientific knowledge, mathematics, and reading comprehension.

Table 18 and Table 19 summarize the results. We find that performance on general benchmarks remains broadly stable after SFT, indicating that fine-tuning on Thunder-NUBench does not substantially harm general capabilities.

Q Total Error Analysis of Model Predictions on the Thunder-NUBench

This section extends the error analysis presented in Section 5.3, providing the complete results. In particular, we examine (i) incorrect choice distributions and (ii) confusion rates for local negation categories, comparing models of different sizes and training paradigms (pretrained vs. instruction-tuned) under both zero-shot and few-shot conditions.

	definition		**detailed**	
	completion	**option**	**completion**	**option**
Gemma2	Table 20	Table 21	Table 22	Table 23
Qwen3	Table 24	Table 25	Table 26	Table 27
Llama3	Table 28	Table 29	Table 30	Table 31
Mistral	Table 32	Table 33	Table 34	Table 35
API		Table 36		Table 37

Table 12: Complete prediction analysis by model family.

We report few-shot results using a fixed random seed (1234). Averaging over multiple seeds was avoided, as it could obscure specific error patterns and make confusion analysis less interpretable. Each table follows the same instruction format, reporting error rates, incorrect choice distributions, and confusion rates across local negation subcategories under zero-shot, few-shot, and SFT conditions.

For API models, we also track cases labeled as "Answer Format Wrong." This category captures instances where the model's output does not follow the required answer format (selecting strictly one of A, B, C, or D). Because such responses cannot be mapped to a specific incorrect option, they are not included in the incorrect choice distribution or confusion rate. Instead, we report their raw counts alongside the other error statistics. This also serves as an indicator of the model's ability to follow output-format instructions.

R Human Evaluation

The human evaluation was approved by the Institutional Review Board (IRB No. 2512/004-015). All participants provided informed consent through an official IRB-approved consent form prior to participation and were fully informed of the study's purpose, procedures, potential risks and benefits, and their right to withdraw at any time without penalty. Participants were also notified that only aggregate statistics of their evaluation scores would be reported.

All responses were collected anonymously, and no personally identifiable information was included in the dataset. Participants were compensated KRW 50,000 for a two-hour session, which exceeds the minimum hourly wage (KRW 10,320 per hour) and constitutes adequate compensation. To minimize fatigue and ensure reliable responses, participants were given sufficient rest during the evaluation, and the total duration was limited to approximately two hours per session. Participants were required to be native or highly proficient English users (CEFR[7] C-level or above), as verified during recruitment. A total of 11 adult participants aged in their 20s to 30s took part in the study, including 6 highly proficient English users at the CEFR C level and 5 native or dominant English speakers.

A total of 50 items were randomly sampled from the multiple-choice evaluation split of Thunder-NUBench, with stratified sampling to preserve the distribution of distractor types. Specifically, the sample included 12 items each for relative clause, participle clause, compound sentence, and adverbial clause (local negation type), and 2 items without local negation. All items were presented using the detailed instruction format. Participants were asked to select the option corresponding to standard negation for each item. No example items were provided during the evaluation.

Mean	**Median**	**Min**	**Max**	**SD**
0.78	0.82	0.58	0.96	0.139

Table 13: Human evaluation results.

Table 13 summarizes the results of the human evaluation conducted with 11 participants. Human accuracy ranges from 0.58 to 0.96, with a mean of 0.78. This range indicates that distinguishing stan-

[7]Common European Framework of Reference for languages.

dard negation from closely related alternatives can be challenging even for human readers, consistent with prior observations that negation processing requires careful semantic and syntactic reasoning. The average human accuracy is higher than the zero-shot performance of most models reported in this paper, suggesting that humans generally handle sentence-level negation more reliably without task-specific training.

evaluation setting	zero-shot		1-shot		5-shot		10-shot		after SFT	
	completion	option	completion (±SD)	option (±SD)	completion (±SD)	option (±SD)	completion (±SD)	option (±SD)	completion	option
gemma-2-2b	0.436	0.236	0.480 (±0.006)	0.328 (±0.004)	0.565 (±0.010)	0.373 (±0.009)	0.599 (±0.004)	0.398 (±0.011)	0.781	0.260
gemma-2-2b-it	0.516	0.504	0.514 (±0.008)	0.501 (±0.008)	0.569 (±0.007)	0.510 (±0.012)	0.594 (±0.003)	0.540 (±0.007)	0.741	0.543
gemma-2-9b	0.470	0.468	0.499 (±0.005)	0.530 (±0.009)	0.565 (±0.007)	0.588 (±0.006)	0.597 (±0.005)	0.626 (±0.005)	0.771	0.751
gemma-2-9b-it	0.508	0.554	0.562 (±0.007)	0.631 (±0.005)	0.653 (±0.007)	0.705 (±0.007)	0.689 (±0.005)	0.724 (±0.011)	0.767	0.814
Qwen3-0.6B-Base	0.453	0.349	0.505 (±0.008)	0.406 (±0.007)	0.581 (±0.005)	0.428 (±0.004)	0.616 (±0.005)	0.465 (±0.010)	0.708	0.479
Qwen3-0.6B	0.416	0.438	0.420 (±0.006)	0.466 (±0.004)	0.487 (±0.003)	0.502 (±0.006)	0.530 (±0.006)	0.528 (±0.007)	0.645	0.555
Qwen3-1.7B-Base	0.481	0.526	0.525 (±0.007)	0.559 (±0.007)	0.577 (±0.004)	0.641 (±0.004)	0.604 (±0.001)	0.676 (±0.007)	0.699	0.493
Qwen3-1.7B	0.398	0.532	0.406 (±0.006)	0.547 (±0.009)	0.476 (±0.007)	0.574 (±0.008)	0.514 (±0.005)	0.578 (±0.007)	0.668	0.608
Qwen3-4B-Base	0.454	0.602	0.489 (±0.002)	0.611 (±0.006)	0.535 (±0.004)	0.653 (±0.003)	0.560 (±0.007)	0.659 (±0.009)	0.733	0.709
Qwen3-4B	0.410	0.562	0.461 (±0.008)	0.619 (±0.005)	0.548 (±0.010)	0.676 (±0.005)	0.597 (±0.007)	0.708 (±0.006)	0.715	0.722
Qwen3-8B-Base	0.474	0.690	0.505 (±0.004)	0.692 (±0.003)	0.566 (±0.009)	0.741 (±0.008)	0.603 (±0.009)	0.758 (±0.007)	0.703	0.768
Qwen3-8B	0.442	0.589	0.487 (±0.011)	0.664 (±0.009)	0.556 (±0.008)	0.712 (±0.006)	0.597 (±0.006)	0.740 (±0.005)	0.722	0.784
Qwen3-14B-Base	0.489	0.708	0.547 (±0.006)	0.766 (±0.013)	0.613 (±0.005)	0.806 (±0.005)	0.653 (±0.007)	0.821 (±0.004)	-	-
Qwen3-14B	0.440	0.661	0.492 (±0.008)	0.677 (±0.009)	0.569 (±0.004)	0.732 (±0.007)	0.622 (±0.006)	0.751 (±0.003)	-	-
Qwen3-32B	0.470	0.730	0.534 (±0.011)	0.768 (±0.010)	0.650 (±0.009)	0.792 (±0.009)	0.690 (±0.006)	0.817 (±0.003)	-	-
Llama-3.1-8B	0.464	0.474	0.495 (±0.006)	0.511 (±0.005)	0.593 (±0.010)	0.580 (±0.011)	0.643 (±0.008)	0.629 (±0.010)	0.774	0.559
Llama-3.1-8B-Instruct	0.464	0.538	0.540 (±0.005)	0.595 (±0.012)	0.628 (±0.006)	0.652 (±0.009)	0.668 (±0.006)	0.673 (±0.005)	0.771	0.644
Llama-3.2-1B	0.409	0.259	0.431 (±0.007)	0.263 (±0.011)	0.492 (±0.006)	0.250 (±0.010)	0.526 (±0.004)	0.262 (±0.009)	0.750	0.259
Llama-3.2-1B-Instruct	0.431	0.301	0.468 (±0.006)	0.376 (±0.013)	0.526 (±0.007)	0.351 (±0.009)	0.548 (±0.009)	0.336 (±0.012)	0.761	0.286
Llama-3.2-3B	0.435	0.362	0.469 (±0.005)	0.443 (±0.004)	0.563 (±0.006)	0.472 (±0.003)	0.588 (±0.006)	0.483 (±0.004)	0.757	0.385
Llama-3.2-3B-Instruct	0.452	0.512	0.456 (±0.007)	0.539 (±0.007)	0.548 (±0.005)	0.499 (±0.002)	0.590 (±0.004)	0.493 (±0.005)	0.795	0.651
Mistral-7B-v0.3	0.467	0.485	0.464 (±0.006)	0.494 (±0.015)	0.557 (±0.014)	0.631 (±0.007)	0.599 (±0.005)	0.653 (±0.007)	0.784	0.480
Mistral-7B-Instruct-v0.3	0.611	0.657	0.643 (±0.006)	0.643 (±0.003)	0.694 (±0.005)	0.638 (±0.009)	0.715 (±0.008)	0.650 (±0.008)	0.793	0.692
Mistral-Nemo-Base-2407 (12B)	0.460	0.412	0.490 (±0.010)	0.520 (±0.007)	0.600 (±0.003)	0.568 (±0.008)	0.649 (±0.007)	0.599 (±0.009)	-	-
Mistral-Nemo-Instruct-2407 (12B)	0.487	0.604	0.528 (±0.004)	0.630 (±0.007)	0.633 (±0.009)	0.660 (±0.006)	0.677 (±0.007)	0.655 (±0.006)	-	-
Mistral-Small-24B-Base-2501	0.474	0.568	0.500 (±0.003)	0.618 (±0.005)	0.597 (±0.006)	0.721 (±0.005)	0.647 (±0.007)	0.783 (±0.003)	-	-
Mistral-Small-24B-Instruct-2501	0.526	0.686	0.569 (±0.006)	0.705 (±0.005)	0.637 (±0.009)	0.731 (±0.011)	0.688 (±0.006)	0.780 (±0.003)	-	-
total average	**0.464**	**0.519**	**0.499**	**0.559**	**0.577**	**0.599**	**0.615**	**0.622**	**0.740**	**0.554**

Table 14: Zero-shot, few-shot, and SFT evaluation results on Thunder-NUBench with the **definition instruction**. SD denotes standard deviation across random seeds or runs. Few-shot results are averaged over five random seeds (42, 1234, 3000, 5000, and 7000) and one, five, and ten demonstration examples (1-, 5-, 10-shot). Red text indicates the model with the highest performance in each column.

	zero-shot		1-shot		5-shot		10-shot		after SFT	
evaluation setting	**completion**	**option**	**completion (±SD)**	**option (±SD)**	**completion (±SD)**	**option (±SD)**	**completion (±SD)**	**option (±SD)**	**completion**	**option**
gemma-2-2b	0.438	0.276	0.494 (±0.005)	0.343 (±0.005)	0.570 (±0.004)	0.360 (±0.018)	0.604 (±0.004)	0.378 (±0.010)	0.768	0.259
gemma-2-2b-it	0.505	0.470	0.488 (±0.013)	0.474 (±0.009)	0.545 (±0.007)	0.487 (±0.011)	0.580 (±0.002)	0.500 (±0.009)	0.733	0.663
gemma-2-9b	0.469	0.466	0.513 (±0.005)	0.530 (±0.009)	0.570 (±0.006)	0.568 (±0.006)	0.604 (±0.005)	0.606 (±0.007)	0.772	0.797
gemma-2-9b-it	0.542	0.569	0.597 (±0.005)	0.635 (±0.005)	0.653 (±0.006)	0.695 (±0.010)	0.690 (±0.003)	0.715 (±0.009)	0.764	0.805
Qwen3-0.6B-Base	0.424	0.342	0.487 (±0.008)	0.397 (±0.004)	0.564 (±0.006)	0.416 (±0.008)	0.607 (±0.006)	0.461 (±0.010)	0.723	0.325
Qwen3-0.6B	0.413	0.459	0.411 (±0.004)	0.414 (±0.009)	0.487 (±0.003)	0.468 (±0.003)	0.535 (±0.007)	0.495 (±0.008)	0.679	0.526
Qwen3-1.7B-Base	0.462	0.483	0.511 (±0.007)	0.542 (±0.004)	0.574 (±0.005)	0.620 (±0.002)	0.597 (±0.005)	0.653 (±0.003)	0.724	0.609
Qwen3-1.7B	0.420	0.447	0.442 (±0.008)	0.502 (±0.013)	0.490 (±0.006)	0.547 (±0.007)	0.522 (±0.006)	0.556 (±0.010)	0.642	0.573
Qwen3-4B-Base	0.485	0.585	0.503 (±0.005)	0.581 (±0.006)	0.546 (±0.004)	0.636 (±0.006)	0.566 (±0.006)	0.649 (±0.009)	0.729	0.711
Qwen3-4B	0.439	0.577	0.489 (±0.010)	0.635 (±0.003)	0.559 (±0.008)	0.652 (±0.004)	0.600 (±0.005)	0.685 (±0.003)	0.688	0.711
Qwen3-8B-Base	0.455	0.668	0.513 (±0.004)	0.700 (±0.007)	0.557 (±0.005)	0.737 (±0.004)	0.591 (±0.009)	0.755 (±0.009)	0.714	0.823
Qwen3-8B	0.473	0.619	0.498 (±0.014)	0.695 (±0.006)	0.548 (±0.007)	0.693 (±0.007)	0.586 (±0.007)	0.723 (±0.007)	0.740	0.760
Qwen3-14B-Base	0.508	0.736	0.553 (±0.008)	0.771 (±0.010)	0.628 (±0.004)	0.797 (±0.007)	0.660 (±0.007)	0.816 (±0.006)	-	-
Qwen3-14B	0.488	0.720	0.522 (±0.007)	0.718 (±0.011)	0.601 (±0.008)	0.762 (±0.009)	0.636 (±0.005)	0.766 (±0.005)	-	-
Qwen3-32B	0.514	0.761	0.570 (±0.011)	0.814 (±0.004)	0.657 (±0.003)	0.808 (±0.001)	0.695 (±0.004)	0.821 (±0.005)	-	-
Llama-3.1-8B	0.451	0.431	0.486 (±0.006)	0.484 (±0.003)	0.564 (±0.008)	0.560 (±0.010)	0.620 (±0.004)	0.611 (±0.010)	0.802	0.672
Llama-3.1-8B-Instruct	0.532	0.514	0.589 (±0.004)	0.588 (±0.010)	0.643 (±0.007)	0.626 (±0.013)	0.679 (±0.007)	0.648 (±0.004)	0.789	0.730
Llama-3.2-1B	0.403	0.259	0.439 (±0.009)	0.263 (±0.007)	0.497 (±0.007)	0.249 (±0.009)	0.527 (±0.005)	0.264 (±0.008)	0.747	0.259
Llama-3.2-1B-Instruct	0.416	0.300	0.468 (±0.003)	0.347 (±0.009)	0.523 (±0.008)	0.339 (±0.010)	0.542 (±0.009)	0.330 (±0.009)	0.703	0.263
Llama-3.2-3B	0.447	0.374	0.478 (±0.003)	0.419 (±0.006)	0.568 (±0.006)	0.470 (±0.006)	0.590 (±0.004)	0.479 (±0.008)	0.761	0.364
Llama-3.2-3B-Instruct	0.488	0.488	0.464 (±0.010)	0.518 (±0.014)	0.540 (±0.007)	0.492 (±0.005)	0.583 (±0.011)	0.481 (±0.006)	0.791	0.582
Mistral-7B-v0.3	0.479	0.502	0.469 (±0.005)	0.472 (±0.016)	0.555 (±0.012)	0.627 (±0.009)	0.597 (±0.006)	0.652 (±0.006)	0.792	0.477
Mistral-7B-Instruct-v0.3	0.662	0.634	0.661 (±0.009)	0.643 (±0.009)	0.699 (±0.004)	0.635 (±0.008)	0.712 (±0.009)	0.643 (±0.008)	0.795	0.711
Mistral-Nemo-Base-2407 (12B)	0.466	0.422	0.498 (±0.008)	0.522 (±0.015)	0.595 (±0.005)	0.557 (±0.006)	0.642 (±0.005)	0.593 (±0.007)	-	-
Mistral-Nemo-Instruct-2407 (12B)	0.511	0.627	0.546 (±0.006)	0.643 (±0.005)	0.636 (±0.008)	0.658 (±0.005)	0.676 (±0.005)	0.654 (±0.006)	-	-
Mistral-Small-24B-Base-2501	0.516	0.575	0.529 (±0.006)	0.633 (±0.007)	0.610 (±0.008)	0.731 (±0.004)	0.656 (±0.005)	0.786 (±0.005)	-	-
Mistral-Small-24B-Instruct-2501	0.581	0.729	0.638 (±0.012)	0.758 (±0.007)	0.668 (±0.006)	0.757 (±0.007)	0.702 (±0.005)	0.804 (±0.005)	-	-
total average	**0.481**	**0.520**	**0.513**	**0.557**	**0.580**	**0.591**	**0.615**	**0.612**	**0.740**	**0.564**

Table 15: Zero-shot, few-shot, and SFT evaluation results on Thunder-NUBench with the **detailed instruction**. SD denotes standard deviation across random seeds or runs. Few-shot results are averaged over five random seeds (42, 1234, 3000, 5000, and 7000) and one, five, and ten demonstration examples (1-, 5-, 10-shot). Red text indicates the model with the highest performance in each column.

models	zero-shot	1-shot (±SD)	5-shot (±SD)	10-shot (±SD)
GPT-4o mini	0.713	0.737 (±0.011)	0.773 (±0.005)	0.795 (±0.011)
GPT-4o	0.782	0.785 (±0.005)	0.800 (±0.012)	0.824 (±0.006)
GPT-4.1 mini	0.757	0.800 (±0.006)	0.840 (±0.007)	0.851 (±0.007)
GPT-4.1	0.881	0.863 (±0.003)	0.874 (±0.003)	0.891 (±0.001)
Haiku 4.5	0.772	0.775 (±0.005)	0.816 (±0.008)	0.837 (±0.006)
Sonnet 4.5	0.876	0.880 (±0.006)	0.875 (±0.002)	0.893 (±0.009)
total average	**0.797**	**0.807**	**0.830**	**0.849**

Table 16: Zero-shot and few-shot results of API models on Thunder-NUBench with the **definition instruction**. SD denotes standard deviation across random seeds or runs. Few-shot results are averaged over five random seeds (42, 1234, 3000, 5000, and 7000) and one, five, and ten demonstration examples (1-, 5-, 10-shot). Red text indicates the model with the highest performance in each column.

models	zero-shot	1-shot(±SD)	5-shot (±SD)	10-shot (±SD)
GPT-4o mini	0.756	0.768 (±0.007)	0.774 (±0.005)	0.790 (±0.011)
GPT-4o	0.863	0.847 (±0.006)	0.825 (±0.012)	0.826 (±0.006)
GPT-4.1 mini	0.902	0.873 (±0.004)	0.870 (±0.007)	0.870 (±0.007)
GPT-4.1	0.936	0.915 (±0.004)	0.901 (±0.003)	0.899 (±0.001)
Haiku 4.5	0.865	0.870 (±0.004)	0.870 (±0.008)	0.879 (±0.006)
Sonnet 4.5	0.915	0.914 (±0.004)	0.882 (±0.002)	0.825 (±0.009)
total average	**0.873**	**0.865**	**0.854**	**0.848**

Table 17: Zero-shot and few-shot results of API models on Thunder-NUBench with the **detailed instruction**. SD denotes standard deviation across random seeds or runs. Few-shot results are averaged over five random seeds (42, 1234, 3000, 5000, and 7000) and one, five, and ten demonstration examples (1-, 5-, 10-shot). Red text indicates the model with the highest performance in each column.

tasks	ARC-Challenge	ARC-Easy	GSM8K	HellaSwag	OpenBookQA	WinoGrande	average
metric	**acc**	**acc**	**exact_match**	**acc_norm**	**acc_norm**	**acc**	
gemma-2-2b	0.468 (+0.001)	0.797 (- 0.002)	0.259 (+0.002)	0.740 (- 0.001)	0.422 (- 0.002)	0.680 (+0.002)	0.561 (+0.000)
gemma-2-2b-it	0.498 (+0.003)	0.809 (+0.002)	0.438 (- 0.014)	0.716 (- 0.001)	0.434 (- 0.002)	0.680 (+0.007)	0.596 (- 0.001)
gemma-2-9b	0.616 (+0.005)	0.874 (+0.002)	0.672 (+0.035)	0.800 (+0.012)	0.472 (- 0.006)	0.741 (+0.009)	0.696 (+0.010)
gemma-2-9b-it	0.629 (+0.008)	0.858 (+0.010)	0.820 (+0.013)	0.800 (+0.000)	0.502 (- 0.008)	0.762 (- 0.010)	0.729 (+0.002)
Qwen3-0.6B-Base	0.326 (- 0.006)	0.668 (- 0.007)	0.510 (- 0.005)	0.517 (+0.002)	0.340 (+0.000)	0.594 (- 0.009)	0.493 (- 0.004)
Qwen3-0.6B	0.317 (- 0.010)	0.607 (- 0.011)	0.409 (+0.019)	0.473 (- 0.009)	0.318 (- 0.006)	0.560 (- 0.006)	0.447 (- 0.004)
Qwen3-1.7B-Base	0.416 (- 0.003)	0.733 (+0.013)	0.723 (- 0.034)	0.665 (+0.000)	0.392 (- 0.002)	0.639 (+0.006)	0.595 (- 0.003)
Qwen3-1.7B	0.403 (+0.009)	0.726 (+0.017)	0.692 (- 0.012)	0.604 (+0.004)	0.366 (- 0.004)	0.608 (+0.010)	0.567 (+0.004)
Qwen3-4B-Base	0.477 (+0.016)	0.789 (+0.006)	0.842 (- 0.008)	0.737 (+0.004)	0.408 (- 0.010)	0.706 (- 0.002)	0.660 (+0.001)
Qwen3-4B	0.507 (+0.017)	0.804 (+0.012)	0.840 (+0.026)	0.684 (+0.003)	0.402 (- 0.004)	0.661 (+0.004)	0.650 (+0.010)
Qwen3-8B-Base	0.532 (+0.002)	0.817 (+0.005)	0.850 (- 0.017)	0.785 (+0.003)	0.410 (+0.012)	0.723 (+0.002)	0.686 (+0.001)
Qwen3-8B	0.555 (+0.008)	0.835 (+0.005)	0.880 (+0.002)	0.750 (+0.006)	0.416 (+0.010)	0.677 (+0.017)	0.686 (+0.008)
Llama-3.1-8B	0.513 (+0.015)	0.814 (+0.014)	0.502 (+0.027)	0.790 (+0.007)	0.450 (+0.016)	0.739 (+0.004)	0.635 (+0.014)
Llama-3.1-8B-Instruct	0.515 (+0.015)	0.819 (+0.006)	0.784 (- 0.012)	0.792 (- 0.004)	0.432 (+0.008)	0.735 (+0.004)	0.680 (+0.003)
Llama-3.2-1B	0.317 (- 0.012)	0.654 (- 0.010)	0.068 (- 0.011)	0.636 (- 0.004)	0.372 (+0.000)	0.603 (+0.010)	0.442 (- 0.005)
Llama-3.2-1B-Instruct	0.358 (- 0.007)	0.683 (+0.002)	0.334 (- 0.002)	0.607 (- 0.006)	0.342 (+0.006)	0.597 (+0.004)	0.487 (- 0.001)
Llama-3.2-3B	0.422 (- 0.007)	0.743 (- 0.003)	0.262 (+0.025)	0.736 (+0.007)	0.428 (- 0.004)	0.698 (- 0.005)	0.548 (+0.002)
Llama-3.2-3B-Instruct	0.436 (+0.003)	0.741 (+0.007)	0.649 (- 0.008)	0.705 (- 0.003)	0.362 (+0.002)	0.678 (- 0.001)	0.595 (+0.000)
Mistral-7B-v0.3	0.488 (+0.015)	0.796 (+0.011)	0.371 (- 0.053)	0.804 (+0.009)	0.436 (- 0.006)	0.736 (- 0.002)	0.605 (- 0.004)
Mistral-7B-Instruct-v0.3	0.572 (- 0.008)	0.842 (- 0.007)	0.498 (- 0.043)	0.829 (- 0.004)	0.474 (- 0.016)	0.744 (- 0.014)	0.660 (- 0.015)
						total average	**0.601 (+0.001)**

Table 18: General benchmark performance after supervised fine-tuning (SFT) on Thunder-NUBench with the **definition instruction**. Here, *acc* denotes accuracy, *acc_norm* is normalized accuracy, and *exact_match* requires an exact string match with the reference answer. For each model and task, the main value reports the performance of the pre-SFT model, while the value in parentheses indicates the change after applying SFT.

tasks	ARC-Challenge	ARC-Easy	GSM8K	HellaSwag	OpenBookQA	WinoGrande	average
metric	acc	acc	exact_match	acc_norm	acc_norm	acc	
gemma-2-2b	0.468 (+0.000)	0.797 (+0.000)	0.259 (+0.000)	0.740 (+0.000)	0.422 (+0.000)	0.680 (+0.000)	0.561 (+0.000)
gemma-2-2b-it	0.498 (- 0.001)	0.809 (- 0.002)	0.438 (+0.000)	0.716 (- 0.001)	0.434 (+0.002)	0.680 (+0.002)	0.596 (+0.000)
gemma-2-9b	0.616 (+0.004)	0.874 (+0.001)	0.672 (+0.035)	0.800 (+0.013)	0.472 (- 0.008)	0.741 (+0.006)	0.696 (+0.009)
gemma-2-9b-it	0.629 (+0.006)	0.858 (+0.011)	0.820 (+0.005)	0.800 (+0.001)	0.502 (+0.002)	0.762 (- 0.004)	0.729 (+0.004)
Qwen3-0.6B-Base	0.326 (+0.000)	0.668 (+0.000)	0.510 (+0.000)	0.517 (+0.000)	0.340 (+0.000)	0.594 (+0.000)	0.493 (+0.000)
Qwen3-0.6B	0.317 (- 0.009)	0.607 (+0.016)	0.409 (+0.014)	0.473 (- 0.012)	0.318 (- 0.002)	0.560 (- 0.001)	0.447 (+0.001)
Qwen3-1.7B-Base	0.416 (+0.002)	0.733 (+0.003)	0.723 (- 0.035)	0.665 (+0.000)	0.392 (- 0.004)	0.639 (+0.002)	0.595 (- 0.005)
Qwen3-1.7B	0.403 (+0.003)	0.726 (+0.021)	0.692 (- 0.024)	0.604 (+0.003)	0.366 (+0.000)	0.608 (+0.009)	0.567 (+0.002)
Qwen3-4B-Base	0.477 (+0.026)	0.789 (+0.009)	0.842 (- 0.036)	0.737 (+0.005)	0.408 (- 0.004)	0.706 (+0.004)	0.660 (+0.001)
Qwen3-4B	0.507 (+0.008)	0.804 (+0.009)	0.840 (+0.008)	0.684 (+0.008)	0.402 (+0.006)	0.661 (+0.006)	0.650 (+0.008)
Qwen3-8B-Base	0.532 (+0.024)	0.817 (+0.011)	0.850 (- 0.040)	0.785 (+0.004)	0.410 (+0.014)	0.723 (- 0.002)	0.686 (+0.002)
Qwen3-8B	0.555 (+0.002)	0.835 (+0.000)	0.880 (- 0.002)	0.750 (+0.005)	0.416 (+0.006)	0.677 (+0.016)	0.686 (+0.005)
Llama-3.1-8B	0.513 (+0.003)	0.814 (+0.012)	0.502 (- 0.001)	0.790 (+0.010)	0.450 (+0.018)	0.739 (+0.001)	0.635 (+0.007)
Llama-3.1-8B-Instruct	0.515 (+0.009)	0.819 (+0.003)	0.784 (- 0.013)	0.792 (- 0.006)	0.432 (+0.006)	0.735 (+0.009)	0.680 (+0.001)
Llama-3.2-1B	0.317 (- 0.009)	0.654 (- 0.005)	0.068 (- 0.024)	0.636 (- 0.008)	0.372 (- 0.004)	0.603 (+0.017)	0.442 (- 0.006)
Llama-3.2-1B-Instruct	0.358 (- 0.004)	0.683 (+0.002)	0.334 (- 0.008)	0.607 (- 0.007)	0.342 (+0.008)	0.597 (+0.000)	0.487 (- 0.002)
Llama-3.2-3B	0.422 (- 0.005)	0.743 (- 0.003)	0.262 (+0.014)	0.736 (+0.004)	0.428 (- 0.006)	0.698 (+0.004)	0.548 (+0.001)
Llama-3.2-3B-Instruct	0.436 (+0.001)	0.741 (+0.006)	0.649 (+0.001)	0.705 (- 0.001)	0.362 (+0.000)	0.678 (+0.000)	0.595 (+0.001)
Mistral-7B-v0.3	0.488 (+0.013)	0.796 (+0.011)	0.371 (- 0.094)	0.804 (+0.004)	0.436 (+0.008)	0.736 (+0.005)	0.605 (- 0.009)
Mistral-7B-Instruct-v0.3	0.572 (- 0.009)	0.842 (- 0.004)	0.498 (- 0.057)	0.829 (- 0.003)	0.474 (- 0.006)	0.744 (- 0.014)	0.660 (- 0.016)
						total average	**0.601 (+0.000)**

Table 19: General benchmark performance after supervised fine-tuning (SFT) on Thunder-NUBench with the **detailed instruction**. Here, *acc* denotes accuracy, *acc_norm* is normalized accuracy, and *exact_match* requires an exact string match with the reference answer. For each model and task, the main value reports the performance of the pre-SFT model, while the value in parentheses indicates the change after applying SFT.

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
gemma-2-2b	**baseline**	**zero-shot**	0.564	74.96	19.41	5.63	25.64	32.47	63.27	53.87
		1-shot	0.523	76.52	19.55	3.94	28.85	30.84	58.84	47.42
		5-shot	0.420	81.47	17.01	1.51	21.47	25.00	51.02	44.19
		10-shot	0.399	81.71	16.30	1.99	21.47	21.43	46.94	45.16
	after SFT	**zero-shot**	0.219	82.25	17.03	0.72	11.86	12.99	26.19	23.55
gemma-2-2b-it	**baseline**	**zero-shot**	0.485	75.29	22.09	2.62	22.12	23.38	48.64	56.77
		1-shot	0.480	77.36	21.32	1.32	29.17	24.68	50.00	49.68
		5-shot	0.427	78.07	20.07	1.86	23.08	23.05	46.94	44.84
		10-shot	0.408	77.24	21.60	1.17	21.47	20.78	41.50	46.45
	after SFT	**zero-shot**	0.259	82.87	15.60	1.53	11.86	15.58	28.57	32.90
gemma-2-9b	**baseline**	**zero-shot**	0.530	75.30	19.61	5.09	24.36	31.49	61.56	48.06
		1-shot	0.500	78.73	17.62	3.65	26.28	29.87	58.84	48.06
		5-shot	0.431	83.79	14.00	2.21	22.44	27.27	50.00	49.68
		10-shot	0.402	84.22	14.60	1.18	21.15	27.27	44.22	47.42
	after SFT	**zero-shot**	0.229	85.47	12.11	2.42	10.26	13.96	22.45	34.19
gemma-2-9b-it	**baseline**	**zero-shot**	0.493	76.01	22.06	1.93	24.68	27.92	52.72	49.68
		1-shot	0.439	77.26	21.66	1.08	24.36	24.68	46.94	44.52
		5-shot	0.352	78.38	20.72	0.90	17.31	17.86	31.97	46.77
		10-shot	0.302	77.95	21.00	1.05	14.10	16.23	26.19	40.65
	after SFT	**zero-shot**	0.233	87.41	10.88	1.70	9.29	14.94	26.19	33.87

Table 20: Error rates, incorrect choice distributions, and local negation confusion rates for the **Gemma2** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
gemma-2-2b	**baseline**	**zero-shot**	0.765	33.20	26.66	40.15	22.12	29.55	29.25	23.87
		1-shot	0.668	40.38	13.54	46.08	21.79	26.62	34.01	29.03
		5-shot	0.623	53.69	12.47	33.84	27.88	34.42	40.48	35.48
		10-shot	0.599	58.81	10.86	30.33	31.09	35.39	43.20	35.81
	after SFT	**zero-shot**	0.740	32.48	34.62	32.90	22.76	27.92	25.51	22.90
gemma-2-2b-it	**baseline**	**zero-shot**	0.496	74.88	8.80	16.32	37.18	26.30	66.67	24.19
		1-shot	0.496	70.93	11.34	17.73	30.45	30.84	55.78	29.03
		5-shot	0.473	67.67	12.56	19.77	31.09	33.12	43.88	24.52
		10-shot	0.467	69.10	12.39	18.51	34.62	31.82	42.18	24.84
	after SFT	**zero-shot**	0.457	68.75	6.77	24.48	29.17	30.84	50.68	19.68
gemma-2-9b	**baseline**	**zero-shot**	0.532	62.89	7.60	29.51	36.22	29.87	57.82	15.16
		1-shot	0.472	62.35	4.71	32.94	34.94	24.35	48.30	14.52
		5-shot	0.402	71.79	6.11	22.09	33.01	23.70	45.92	17.10
		10-shot	0.378	71.43	4.62	23.95	33.33	20.78	40.82	16.77
	after SFT	**zero-shot**	0.249	64.01	16.24	19.75	21.15	11.36	23.81	9.68
gemma-2-9b-it	**baseline**	**zero-shot**	0.447	73.00	22.02	4.97	39.42	22.40	51.70	21.61
		1-shot	0.370	80.69	13.95	5.36	35.26	22.73	42.86	22.58
		5-shot	0.286	83.66	11.63	4.71	28.85	17.53	31.63	20.97
		10-shot	0.272	82.51	12.54	4.96	28.85	17.53	28.23	18.06
	after SFT	**zero-shot**	0.186	73.62	21.70	4.68	11.86	10.71	25.51	9.03

Table 21: Error rates, incorrect choice distributions, and local negation confusion rates for the **Gemma2** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
gemma-2-2b	**baseline**	**zero-shot**	0.562	74.89	19.04	6.06	26.60	34.74	60.88	52.26
		1-shot	0.507	76.37	19.41	4.23	26.60	30.84	56.46	46.45
		5-shot	0.425	82.09	16.04	1.87	20.83	25.97	52.72	45.16
		10-shot	0.392	81.98	15.99	2.02	20.19	22.40	46.26	44.19
	after SFT	**zero-shot**	0.232	82.59	16.72	0.68	11.22	13.64	27.21	27.42
gemma-2-2b-it	**baseline**	**zero-shot**	0.495	78.37	18.75	2.88	24.68	27.60	54.76	53.55
		1-shot	0.493	79.74	18.81	1.45	29.17	28.25	52.38	52.90
		5-shot	0.450	80.81	17.61	1.58	25.64	24.35	51.70	49.03
		10-shot	0.422	80.26	18.42	1.32	22.76	22.40	46.60	48.39
	after SFT	**zero-shot**	0.267	81.90	16.62	1.48	12.50	14.61	32.65	30.97
gemma-2-9b	**baseline**	**zero-shot**	0.531	74.63	20.60	4.78	25.00	33.77	62.59	43.23
		1-shot	0.487	78.50	17.59	3.91	25.96	30.52	59.18	42.90
		5-shot	0.425	83.21	14.37	2.43	21.47	25.65	51.36	48.06
		10-shot	0.393	83.23	15.35	1.41	20.83	25.97	43.54	44.84
	after SFT	**zero-shot**	0.228	86.06	11.85	2.09	8.97	14.61	23.81	33.55
gemma-2-9b-it	**baseline**	**zero-shot**	0.458	76.95	21.14	1.91	18.59	27.60	56.80	43.23
		1-shot	0.401	77.62	21.58	0.79	20.51	20.78	45.24	42.26
		5-shot	0.345	78.85	20.23	0.92	16.35	17.53	33.67	44.84
		10-shot	0.308	78.87	19.85	1.29	14.10	17.21	26.53	42.26
	after SFT	**zero-shot**	0.236	85.91	11.41	2.68	10.26	13.64	26.87	33.23

Table 22: Error rates, incorrect choice distributions, and local negation confusion rates for the **Gemma2** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
gemma-2-2b	**baseline**	**zero-shot**	0.724	35.93	29.46	34.61	23.72	31.49	29.25	22.90
		1-shot	0.660	42.91	15.02	42.07	22.76	28.90	35.37	30.00
		5-shot	0.629	53.47	12.48	34.05	28.21	34.74	40.82	35.16
		10-shot	0.612	57.64	11.53	30.83	32.05	34.42	41.84	37.42
	after SFT	**zero-shot**	0.741	32.55	34.58	32.87	22.76	27.92	25.51	23.23
gemma-2-2b-it	**baseline**	**zero-shot**	0.531	82.06	7.47	10.46	42.95	38.96	67.69	30.97
		1-shot	0.539	70.84	8.84	20.32	32.37	31.82	63.27	30.97
		5-shot	0.508	64.06	12.81	23.12	30.13	32.47	45.92	26.13
		10-shot	0.504	61.64	12.11	26.26	31.41	29.22	43.20	24.84
	after SFT	**zero-shot**	0.337	72.00	10.12	17.88	21.15	25.65	35.71	18.06
gemma-2-9b	**baseline**	**zero-shot**	0.534	73.70	8.47	17.83	42.63	32.47	66.33	21.94
		1-shot	0.476	64.83	6.50	28.67	36.22	25.32	50.00	16.45
		5-shot	0.422	70.11	5.45	24.44	33.65	22.73	48.30	18.06
		10-shot	0.403	69.88	3.94	26.18	34.62	20.45	44.22	17.42
	after SFT	**zero-shot**	0.203	75.00	11.72	13.28	22.12	12.01	16.33	12.26
gemma-2-9b-it	**baseline**	**zero-shot**	0.431	83.27	11.58	5.15	35.90	23.70	63.61	26.13
		1-shot	0.361	83.30	10.77	5.93	35.58	21.75	43.54	23.55
		5-shot	0.305	84.11	10.42	5.47	30.45	21.43	32.31	21.61
		10-shot	0.284	82.12	12.85	5.03	30.13	18.51	28.57	19.03
	after SFT	**zero-shot**	0.195	69.92	17.07	13.01	16.67	13.96	12.59	12.90

Table 23: Error rates, incorrect choice distributions, and local negation confusion rates for the **Gemma2** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Qwen3-0.6B-Base	**baseline**	**zero-shot**	0.547	72.32	23.48	4.20	29.49	34.74	50.68	48.71
		1-shot	0.487	74.27	22.15	3.58	31.09	28.90	46.60	42.90
		5-shot	0.412	77.07	19.46	3.47	23.40	21.75	43.54	42.58
		10-shot	0.390	79.27	18.90	1.83	22.76	22.08	40.82	42.26
	after SFT	**zero-shot**	0.292	83.97	15.22	0.82	18.59	17.53	30.27	34.84
Qwen3-0.6B	**baseline**	**zero-shot**	0.584	74.46	20.52	5.03	26.92	32.14	64.63	56.45
		1-shot	0.571	76.81	19.17	4.03	29.81	34.42	62.24	55.16
		5-shot	0.515	79.82	16.95	3.24	25.96	28.57	57.48	58.06
		10-shot	0.470	80.74	15.71	3.55	25.00	25.97	48.64	57.10
	after SFT	**zero-shot**	0.355	84.82	13.84	1.34	20.51	20.78	38.10	45.16
Qwen3-1.7B-Base	**baseline**	**zero-shot**	0.519	73.70	22.02	4.28	28.21	34.74	56.80	38.71
		1-shot	0.467	74.87	22.92	2.21	27.88	30.19	50.34	36.45
		5-shot	0.416	76.57	21.14	2.29	22.12	26.95	42.86	40.00
		10-shot	0.396	79.56	18.84	1.60	23.72	24.35	41.84	40.32
	after SFT	**zero-shot**	0.301	89.47	9.74	0.79	19.23	24.35	36.05	31.94
Qwen3-1.7B	**baseline**	**zero-shot**	0.602	67.46	24.64	7.91	28.53	37.99	65.31	36.77
		1-shot	0.601	71.50	24.01	4.49	31.41	34.74	64.29	47.74
		5-shot	0.517	76.61	20.64	2.75	26.92	28.25	58.84	50.65
		10-shot	0.483	78.33	19.70	1.97	27.88	24.35	54.08	50.32
	after SFT	**zero-shot**	0.332	83.77	14.56	1.67	15.71	24.68	37.41	37.42
Qwen3-4B-Base	**baseline**	**zero-shot**	0.546	76.34	20.46	3.19	31.73	32.47	62.59	46.13
		1-shot	0.515	74.73	22.03	3.24	30.13	30.52	54.42	44.19
		5-shot	0.467	74.19	22.41	3.40	26.28	27.27	48.64	41.29
		10-shot	0.443	76.21	21.65	2.15	25.64	25.65	47.96	40.65
	after SFT	**zero-shot**	0.267	89.61	10.09	0.30	16.03	20.13	30.95	31.94
Qwen3-4B	**baseline**	**zero-shot**	0.590	73.66	20.83	5.51	31.73	40.58	68.37	39.68
		1-shot	0.534	76.97	18.13	4.90	35.26	33.77	62.24	39.03
		5-shot	0.435	80.66	17.34	2.01	25.00	25.97	52.72	41.61
		10-shot	0.408	81.91	16.34	1.75	23.72	26.95	48.30	39.35
	after SFT	**zero-shot**	0.285	85.79	12.81	1.39	16.03	18.83	24.83	40.97
Qwen3-8B-Base	**baseline**	**zero-shot**	0.526	73.76	22.47	3.77	27.56	33.12	61.56	38.71
		1-shot	0.502	72.67	23.85	3.48	27.24	28.57	56.12	39.35
		5-shot	0.420	73.91	22.68	3.40	20.83	23.05	44.22	40.32
		10-shot	0.401	77.62	20.40	1.98	22.44	25.32	40.14	40.65
	after SFT	**zero-shot**	0.297	85.07	13.60	1.33	16.67	19.81	35.03	33.23
Qwen3-8B	**baseline**	**zero-shot**	0.558	71.59	22.87	5.54	27.24	34.74	65.31	38.71
		1-shot	0.501	77.22	18.20	4.59	32.37	32.14	61.22	34.84
		5-shot	0.436	80.73	16.73	2.55	26.92	30.84	48.98	39.03
		10-shot	0.397	81.20	17.00	1.80	25.32	26.95	44.56	36.45
	after SFT	**zero-shot**	0.278	83.48	15.10	1.42	14.74	16.56	31.97	32.90
Qwen3-14B-Base	**baseline**	**zero-shot**	0.512	70.08	23.57	6.36	24.68	30.84	53.74	39.35
		1-shot	0.450	72.01	23.06	4.93	24.68	26.62	45.58	37.42
		5-shot	0.380	75.57	21.50	2.92	20.83	22.73	37.76	37.42
		10-shot	0.347	76.89	21.05	2.06	20.51	20.13	31.97	37.42
Qwen3-14B	**baseline**	**zero-shot**	0.560	72.52	21.10	6.37	27.88	32.47	65.31	42.90
		1-shot	0.500	78.92	16.64	4.44	32.37	31.17	59.18	40.97
		5-shot	0.429	80.41	17.19	2.40	25.64	26.95	47.28	42.90
		10-shot	0.381	82.29	15.62	2.08	23.72	25.00	36.05	44.52
Qwen3-32B	**baseline**	**zero-shot**	0.531	76.68	17.64	5.68	27.24	34.42	67.01	40.32
		1-shot	0.462	77.49	17.70	4.81	26.28	28.25	55.78	38.06
		5-shot	0.347	81.46	15.79	2.75	18.91	22.73	39.80	35.48
		10-shot	0.316	83.17	14.32	2.51	19.87	20.78	32.31	35.48

Table 24: Error rates, incorrect choice distributions, and local negation confusion rates for the **Qwen3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Qwen3-0.6B-Base	**baseline**	**zero-shot**	0.651	50.55	7.06	42.39	25.32	37.99	39.80	32.90
		1-shot	0.603	55.26	6.84	37.89	25.64	37.01	48.98	26.45
		5-shot	0.577	55.57	3.30	41.13	27.56	33.12	47.28	24.84
		10-shot	0.525	60.12	3.78	36.10	29.81	30.52	45.92	24.52
	after SFT	**zero-shot**	0.521	67.43	11.42	21.16	32.37	39.94	42.18	30.65
Qwen3-0.6B	**baseline**	**zero-shot**	0.562	69.96	7.90	22.14	37.82	38.64	54.08	32.26
		1-shot	0.532	59.31	14.75	25.93	26.28	29.87	47.28	27.42
		5-shot	0.493	71.66	5.80	22.54	32.37	30.52	54.76	28.71
		10-shot	0.469	69.88	6.60	23.52	32.37	29.87	49.32	24.19
	after SFT	**zero-shot**	0.445	67.02	10.34	22.64	24.68	27.27	47.96	23.87
Qwen3-1.7B-Base	**baseline**	**zero-shot**	0.474	65.22	9.87	24.92	31.41	23.70	52.04	21.29
		1-shot	0.444	72.50	13.57	13.93	33.97	30.84	38.78	29.35
		5-shot	0.361	75.82	12.53	11.65	29.81	25.32	36.39	21.61
		10-shot	0.313	76.14	11.42	12.44	25.96	20.78	34.69	17.10
	after SFT	**zero-shot**	0.507	58.84	12.83	28.33	33.97	26.30	43.54	19.68
Qwen3-1.7B	**baseline**	**zero-shot**	0.468	70.85	15.76	13.39	26.92	31.82	46.26	32.26
		1-shot	0.451	67.31	14.24	18.45	24.04	28.57	47.62	25.81
		5-shot	0.424	72.47	11.80	15.73	26.60	23.70	51.70	25.48
		10-shot	0.421	72.13	9.98	17.89	25.96	23.70	53.40	23.23
	after SFT	**zero-shot**	0.392	68.83	22.27	8.91	30.13	27.92	30.27	22.90
Qwen3-4B-Base	**baseline**	**zero-shot**	0.398	78.69	15.74	5.58	38.14	22.40	45.58	23.55
		1-shot	0.386	77.82	19.30	2.87	40.71	29.22	36.73	17.42
		5-shot	0.350	75.74	19.27	4.99	40.38	26.95	24.15	17.42
		10-shot	0.336	77.36	17.45	5.19	39.10	26.95	24.49	16.45
	after SFT	**zero-shot**	0.291	89.37	8.45	2.18	32.05	18.18	43.20	14.52
Qwen3-4B	**baseline**	**zero-shot**	0.439	82.28	15.55	2.17	38.46	30.19	56.46	24.52
		1-shot	0.375	80.97	17.34	1.69	33.97	25.32	46.26	20.32
		5-shot	0.321	84.20	12.35	3.46	32.69	23.70	33.67	21.61
		10-shot	0.301	83.64	13.19	3.17	30.77	24.68	26.87	21.29
	after SFT	**zero-shot**	0.278	87.14	9.43	3.43	28.53	22.73	37.76	11.29
Qwen3-8B-Base	**baseline**	**zero-shot**	0.310	79.03	13.81	7.16	29.17	20.78	33.33	18.06
		1-shot	0.313	76.96	13.42	9.62	31.73	26.95	23.81	16.77
		5-shot	0.254	76.25	15.00	8.75	28.21	21.10	16.67	13.55
		10-shot	0.250	78.41	13.02	8.57	27.56	20.45	16.67	15.81
	after SFT	**zero-shot**	0.232	72.7	20.82	6.48	19.55	17.21	18.71	14.19
Qwen3-8B	**baseline**	**zero-shot**	0.411	84.17	12.55	3.28	41.67	31.82	49.32	20.32
		1-shot	0.326	82.00	13.14	4.87	35.90	26.30	33.33	14.84
		5-shot	0.279	83.81	10.80	5.40	34.29	25.32	21.09	15.48
		10-shot	0.256	82.35	11.15	6.50	30.77	23.38	16.67	15.81
	after SFT	**zero-shot**	0.217	84.62	11.72	3.66	25.00	22.40	15.65	12.26
Qwen3-14B-Base	**baseline**	**zero-shot**	0.292	81.52	15.22	3.26	23.72	16.56	39.12	19.35
		1-shot	0.236	75.17	18.79	6.04	21.15	16.56	26.53	9.35
		5-shot	0.190	73.22	23.01	3.77	18.59	14.61	13.27	10.65
		10-shot	0.177	75.34	21.97	2.69	17.95	14.61	9.52	12.58
Qwen3-14B	**baseline**	**zero-shot**	0.340	79.67	17.52	2.80	32.37	22.08	45.92	11.94
		1-shot	0.316	82.96	12.28	4.76	30.77	21.75	43.20	13.23
		5-shot	0.263	85.20	12.99	1.81	31.09	25.32	26.19	9.68
		10-shot	0.250	84.76	12.70	2.54	33.01	21.75	20.75	11.61
Qwen3-32B	**baseline**	**zero-shot**	0.270	81.23	16.72	2.05	25.00	24.35	26.53	14.84
		1-shot	0.232	80.82	18.15	1.03	24.68	23.05	19.73	9.68
		5-shot	0.201	81.10	18.50	0.39	24.36	19.48	14.97	8.39
		10-shot	0.181	77.63	21.93	0.44	19.55	16.88	12.24	9.03

Table 25: Error rates, incorrect choice distributions, and local negation confusion rates for the **Qwen3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Qwen3-0.6B-Base	**baseline**	**zero-shot**	0.577	71.53	22.97	5.50	29.81	35.39	54.42	50.97
		1-shot	0.502	73.46	21.80	4.74	29.17	30.84	50.34	42.26
		5-shot	0.426	77.28	19.18	3.54	23.40	23.70	44.22	44.84
		10-shot	0.401	79.01	18.61	2.38	22.12	23.70	42.18	42.90
	after SFT	**zero-shot**	0.277	83.67	15.47	0.86	17.31	17.53	28.23	32.58
Qwen3-0.6B	**baseline**	**zero-shot**	0.587	77.70	17.57	4.73	27.24	41.56	62.93	57.10
		1-shot	0.590	76.48	18.82	4.70	30.77	38.64	64.29	53.23
		5-shot	0.517	78.53	17.18	4.29	25.00	31.82	57.82	53.55
		10-shot	0.471	81.31	15.15	3.54	25.64	27.92	52.38	52.58
	after SFT	**zero-shot**	0.321	84.44	14.07	1.48	17.63	20.13	35.03	39.35
Qwen3-1.7B-Base	**baseline**	**zero-shot**	0.538	71.24	22.71	6.05	29.81	33.12	58.50	37.42
		1-shot	0.479	74.17	22.85	2.98	28.21	30.52	53.06	35.48
		5-shot	0.417	77.38	20.53	2.09	23.40	26.30	41.84	41.94
		10-shot	0.401	80.40	18.22	1.39	23.72	24.68	41.84	42.90
	after SFT	**zero-shot**	0.276	85.63	12.93	1.44	17.95	24.03	31.97	23.87
Qwen3-1.7B	**baseline**	**zero-shot**	0.581	67.76	25.27	6.97	27.88	37.66	63.27	34.52
		1-shot	0.570	71.35	24.34	4.31	29.49	33.77	61.22	44.19
		5-shot	0.504	76.54	21.10	2.36	26.60	26.95	58.16	48.06
		10-shot	0.485	77.45	20.42	2.12	27.56	25.32	54.42	48.39
	after SFT	**zero-shot**	0.358	82.71	15.74	1.55	18.59	26.30	39.12	38.39
Qwen3-4B-Base	**baseline**	**zero-shot**	0.515	75.50	21.57	2.93	27.56	30.84	55.10	47.42
		1-shot	0.494	75.12	22.15	2.73	29.49	29.55	52.72	41.94
		5-shot	0.449	74.20	22.79	3.00	25.00	27.27	44.22	41.29
		10-shot	0.440	76.22	21.80	1.98	25.64	25.97	45.58	41.61
	after SFT	**zero-shot**	0.271	86.26	12.87	0.88	16.67	19.48	31.63	29.03
Qwen3-4B	**baseline**	**zero-shot**	0.562	73.73	20.34	5.93	28.21	38.96	65.31	39.35
		1-shot	0.495	74.68	20.19	5.13	28.85	29.87	57.82	36.77
		5-shot	0.431	78.12	19.12	2.76	23.08	25.65	51.70	39.35
		10-shot	0.398	79.08	18.53	2.39	21.79	25.65	46.60	36.45
	after SFT	**zero-shot**	0.313	83.76	14.47	1.78	16.67	21.10	28.91	41.29
Qwen3-8B-Base	**baseline**	**zero-shot**	0.545	74.09	21.69	4.22	29.17	33.44	61.22	43.55
		1-shot	0.489	73.42	23.18	3.40	27.56	28.25	55.44	37.74
		5-shot	0.436	73.09	23.27	3.64	22.44	24.35	44.90	40.32
		10-shot	0.412	76.35	21.15	2.50	22.76	25.65	40.48	41.29
	after SFT	**zero-shot**	0.286	86.43	12.47	1.11	16.35	20.45	34.01	31.61
Qwen3-8B	**baseline**	**zero-shot**	0.527	71.84	23.04	5.12	26.60	35.06	61.22	34.19
		1-shot	0.484	75.41	20.49	4.10	31.73	30.19	57.14	32.26
		5-shot	0.442	80.43	17.59	1.97	27.56	30.52	49.32	39.68
		10-shot	0.408	81.36	16.70	1.94	25.64	28.57	45.24	38.06
	after SFT	**zero-shot**	0.260	83.84	14.33	1.83	13.78	17.21	27.55	31.61
Qwen3-14B-Base	**baseline**	**zero-shot**	0.493	70.69	23.67	5.64	25.00	31.17	46.60	41.29
		1-shot	0.443	74.37	22.58	3.05	25.64	27.27	46.26	37.10
		5-shot	0.369	76.99	20.65	2.37	20.19	22.08	35.37	39.68
		10-shot	0.335	77.07	21.75	1.18	20.51	19.81	28.91	37.42
Qwen3-14B	**baseline**	**zero-shot**	0.512	69.35	23.37	7.28	25.64	33.12	56.46	32.26
		1-shot	0.470	75.89	18.89	5.23	31.09	30.84	55.78	30.32
		5-shot	0.397	79.60	18.60	1.80	23.72	25.65	44.90	36.45
		10-shot	0.361	79.56	18.24	2.20	23.72	24.03	33.67	37.10
Qwen3-32B	**baseline**	**zero-shot**	0.486	70.31	20.72	8.97	24.68	35.06	60.88	21.61
		1-shot	0.418	74.95	18.60	6.45	23.72	27.60	53.06	25.81
		5-shot	0.341	81.40	15.58	3.02	20.51	23.38	40.14	30.97
		10-shot	0.309	82.01	15.17	2.83	19.23	21.75	32.99	30.65

Table 26: Error rates, incorrect choice distributions, and local negation confusion rates for the **Qwen3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Qwen3-0.6B-Base	**baseline**	**zero-shot**	0.658	48.80	4.46	46.75	27.88	37.66	38.78	28.39
		1-shot	0.606	53.14	7.46	39.40	25.00	36.04	47.62	24.84
		5-shot	0.588	52.77	3.24	43.99	26.60	33.12	43.20	25.48
		10-shot	0.537	61.00	3.55	35.45	30.45	33.77	46.60	24.84
	after SFT	**zero-shot**	0.675	37.25	30.79	31.96	23.08	27.92	31.63	21.29
Qwen3-0.6B	**baseline**	**zero-shot**	0.541	74.93	7.92	17.16	43.91	40.91	57.48	25.48
		1-shot	0.588	53.37	15.36	31.27	25.96	29.87	45.58	28.71
		5-shot	0.535	65.04	6.81	28.15	31.09	31.49	52.72	29.03
		10-shot	0.502	65.72	6.64	27.65	31.73	29.87	52.72	22.58
	after SFT	**zero-shot**	0.474	64.55	20.57	14.88	27.56	29.22	46.60	23.55
Qwen3-1.7B-Base	**baseline**	**zero-shot**	0.517	57.21	13.50	29.29	26.92	28.25	49.32	18.39
		1-shot	0.457	69.97	13.89	16.15	31.73	32.79	40.14	27.42
		5-shot	0.382	71.78	12.86	15.35	29.49	25.97	38.10	20.00
		10-shot	0.343	72.29	11.55	16.17	25.32	23.70	36.73	17.10
	after SFT	**zero-shot**	0.391	68.15	13.18	18.66	30.45	26.30	41.16	12.58
Qwen3-1.7B	**baseline**	**zero-shot**	0.553	71.74	18.94	9.33	37.50	37.99	53.74	34.84
		1-shot	0.508	68.12	14.06	17.81	30.45	31.82	55.44	25.81
		5-shot	0.450	72.31	10.41	17.28	27.56	25.97	56.80	24.84
		10-shot	0.455	71.25	10.10	18.64	30.13	25.65	55.78	23.23
	after SFT	**zero-shot**	0.427	66.05	26.90	7.05	29.81	29.87	34.35	22.58
Qwen3-4B-Base	**baseline**	**zero-shot**	0.416	79.01	14.89	6.11	36.86	24.68	48.64	25.81
		1-shot	0.413	76.20	18.23	5.57	41.35	27.92	42.86	18.06
		5-shot	0.366	75.49	17.57	6.94	39.42	25.97	30.27	18.06
		10-shot	0.355	76.06	17.00	6.94	40.71	26.30	27.21	16.77
	after SFT	**zero-shot**	0.290	84.38	11.78	3.84	28.85	15.91	39.80	16.77
Qwen3-4B	**baseline**	**zero-shot**	0.424	81.27	17.23	1.50	37.18	29.22	51.36	24.84
		1-shot	0.364	79.52	18.52	1.96	35.26	24.03	41.16	19.35
		5-shot	0.347	82.88	14.61	2.51	32.37	26.30	38.10	22.26
		10-shot	0.317	82.00	14.50	3.50	33.97	23.70	29.93	19.68
	after SFT	**zero-shot**	0.289	81.32	17.58	1.10	28.21	21.43	35.37	12.26
Qwen3-8B-Base	**baseline**	**zero-shot**	0.332	78.04	12.41	9.55	29.81	24.68	36.05	16.77
		1-shot	0.305	77.86	13.80	8.33	30.45	26.62	25.51	15.16
		5-shot	0.258	75.08	16.31	8.62	27.24	20.13	18.37	13.87
		10-shot	0.251	78.23	12.93	8.83	27.24	19.81	18.03	15.81
	after SFT	**zero-shot**	0.177	74.89	16.14	8.97	17.63	14.61	9.52	12.58
Qwen3-8B	**baseline**	**zero-shot**	0.381	86.07	12.27	1.66	43.59	33.12	37.41	21.29
		1-shot	0.305	83.38	14.03	2.60	36.54	24.03	26.87	17.42
		5-shot	0.297	81.87	13.60	4.53	34.62	26.30	21.77	17.42
		10-shot	0.276	81.32	12.93	5.75	31.73	23.70	18.71	18.06
	after SFT	**zero-shot**	0.240	85.48	10.56	3.96	29.17	21.43	19.05	14.84
Qwen3-14B-Base	**baseline**	**zero-shot**	0.264	79.58	15.92	4.50	20.83	16.56	27.89	21.61
		1-shot	0.242	73.11	21.97	4.92	21.79	16.88	20.75	13.55
		5-shot	0.210	75.09	21.51	3.40	22.12	15.58	13.95	13.23
		10-shot	0.182	77.29	20.52	2.18	18.59	15.58	8.50	14.84
Qwen3-14B	**baseline**	**zero-shot**	0.280	79.04	19.55	1.42	28.85	23.05	25.17	14.19
		1-shot	0.291	80.65	15.53	3.81	31.73	21.10	29.93	14.19
		5-shot	0.241	86.51	11.18	2.30	30.13	20.78	20.75	14.19
		10-shot	0.241	84.21	13.82	1.97	31.73	21.75	16.33	13.55
Qwen3-32B	**baseline**	**zero-shot**	0.240	78.81	19.54	1.66	20.83	24.35	18.03	14.52
		1-shot	0.189	80.67	18.49	0.84	22.44	20.45	10.88	8.71
		5-shot	0.194	80.82	18.78	0.41	21.47	20.45	12.93	9.68
		10-shot	0.175	80.54	19.00	0.45	19.23	18.18	9.18	11.29

Table 27: Error rates, incorrect choice distributions, and local negation confusion rates for the **Qwen3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Llama-3.1-8B	**baseline**	**zero-shot**	0.536	76.92	17.60	5.47	25.32	30.19	60.20	55.16
		1-shot	0.501	80.06	16.14	3.80	24.68	29.22	57.14	55.16
		5-shot	0.391	83.77	14.20	2.03	21.79	23.05	43.54	47.10
		10-shot	0.356	83.07	15.14	1.78	19.23	21.43	38.78	42.90
	after SFT	**zero-shot**	0.226	88.07	10.18	1.75	10.58	15.91	24.15	31.61
Llama-3.1-8B-Instruct	**baseline**	**zero-shot**	0.536	73.82	23.96	2.22	25.00	31.82	58.50	48.71
		1-shot	0.457	79.51	19.10	1.39	24.36	28.57	48.98	48.39
		5-shot	0.367	85.31	14.25	0.43	18.91	22.40	40.14	48.06
		10-shot	0.332	85.17	14.35	0.48	20.51	20.78	32.99	42.26
	after SFT	**zero-shot**	0.229	86.51	12.46	1.04	10.90	14.29	23.13	33.55
Llama-3.2-1B	**baseline**	**zero-shot**	0.591	71.01	24.03	4.97	21.79	29.55	63.95	58.71
		1-shot	0.570	72.32	23.37	4.31	25.32	32.47	63.95	49.35
		5-shot	0.514	75.77	20.52	3.70	20.83	27.27	63.95	49.68
		10-shot	0.468	78.14	18.64	3.22	19.87	25.97	59.86	46.13
	after SFT	**zero-shot**	0.250	76.83	21.59	1.59	13.46	14.61	24.15	27.10
Llama-3.2-1B-Instruct	**baseline**	**zero-shot**	0.569	68.48	27.89	3.63	25.00	30.84	58.16	47.42
		1-shot	0.526	69.83	27.75	2.41	25.32	26.95	54.76	45.16
		5-shot	0.472	70.92	25.38	3.70	21.15	24.68	51.36	41.61
		10-shot	0.447	72.70	23.94	3.37	22.12	22.40	47.96	42.26
	after SFT	**zero-shot**	0.239	81.06	17.94	1.00	13.14	15.26	21.43	30.00
Llama-3.2-3B	**baseline**	**zero-shot**	0.565	74.58	19.94	5.48	24.36	31.49	62.93	55.81
		1-shot	0.538	78.17	18.88	2.95	31.09	31.82	60.88	50.32
		5-shot	0.429	79.85	17.56	2.59	22.44	27.27	47.96	44.19
		10-shot	0.413	81.96	15.16	2.88	23.72	26.95	45.92	43.55
	after SFT	**zero-shot**	0.244	84.36	14.33	1.30	10.58	16.88	24.83	32.58
Llama-3.2-3B-Instruct	**baseline**	**zero-shot**	0.548	76.70	21.71	1.59	27.88	33.12	54.42	58.39
		1-shot	0.536	76.33	22.34	1.33	31.41	31.17	52.72	53.87
		5-shot	0.448	78.76	20.18	1.06	24.68	26.95	40.82	53.23
		10-shot	0.414	79.12	19.73	1.15	24.04	24.35	37.41	49.35
	after SFT	**zero-shot**	0.205	84.50	13.18	2.33	8.33	10.06	25.17	28.06

Table 28: Error rates, incorrect choice distributions, and local negation confusion rates for the **Llama3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Llama-3.1-8B	**baseline**	**zero-shot**	0.526	62.29	8.14	29.56	28.21	24.68	54.42	28.71
		1-shot	0.493	70.42	6.11	23.47	36.22	30.52	47.28	29.68
		5-shot	0.422	84.59	4.32	11.09	41.35	26.95	49.32	30.00
		10-shot	0.372	88.27	4.48	7.25	36.86	25.32	40.48	32.90
	after SFT	**zero-shot**	0.441	62.77	17.63	19.60	27.24	20.45	41.16	25.81
Llama-3.1-8B-Instruct	**baseline**	**zero-shot**	0.462	70.67	18.52	10.81	27.56	24.68	53.40	30.00
		1-shot	0.404	70.53	18.07	11.39	26.92	23.70	41.84	25.48
		5-shot	0.332	81.62	11.22	7.16	25.64	18.83	40.48	27.42
		10-shot	0.325	85.85	9.27	4.88	28.85	19.81	37.41	29.35
	after SFT	**zero-shot**	0.356	71.94	15.37	12.69	25.00	19.81	44.56	17.10
Llama-3.2-1B	**baseline**	**zero-shot**	0.741	32.44	34.69	32.87	22.76	27.92	25.51	22.90
		1-shot	0.719	30.35	33.44	36.20	20.83	20.78	25.85	22.58
		5-shot	0.751	33.16	34.85	32.00	24.36	24.68	27.89	25.81
		10-shot	0.738	33.08	33.83	33.08	25.00	25.65	25.85	24.19
	after SFT	**zero-shot**	0.741	32.44	34.69	32.87	22.76	27.92	25.51	22.90
Llama-3.2-1B-Instruct	**baseline**	**zero-shot**	0.699	39.68	27.89	32.43	26.28	34.09	29.59	24.52
		1-shot	0.608	53.59	19.17	27.25	25.32	39.29	41.50	28.71
		5-shot	0.645	47.36	22.88	29.77	27.56	38.64	31.97	27.74
		10-shot	0.678	43.98	25.26	30.76	28.21	33.77	33.33	27.74
	after SFT	**zero-shot**	0.714	35.78	29.22	35.00	23.72	30.52	26.53	24.52
Llama-3.2-3B	**baseline**	**zero-shot**	0.638	50.37	26.37	23.26	23.08	35.39	40.82	33.55
		1-shot	0.555	68.43	14.00	17.57	33.33	34.74	46.94	41.94
		5-shot	0.527	79.70	10.83	9.47	38.14	37.66	51.02	46.77
		10-shot	0.516	83.23	10.15	6.62	40.38	36.36	49.66	50.65
	after SFT	**zero-shot**	0.615	46.97	28.77	24.26	22.44	30.52	35.71	30.65
Llama-3.2-3B-Instruct	**baseline**	**zero-shot**	0.488	80.00	7.97	12.03	27.24	35.39	64.29	35.16
		1-shot	0.458	77.68	9.86	12.46	24.36	28.25	60.88	34.52
		5-shot	0.502	74.25	8.85	16.90	29.49	27.27	67.35	30.97
		10-shot	0.498	74.04	8.44	17.52	31.09	26.62	64.29	31.29
	after SFT	**zero-shot**	0.349	70.45	6.59	22.95	15.71	16.88	45.58	24.19

Table 29: Error rates, incorrect choice distributions, and local negation confusion rates for the **Llama3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Llama-3.1-8B	**baseline**	**zero-shot**	0.549	73.99	19.08	6.94	26.60	28.57	61.56	51.61
		1-shot	0.516	75.42	20.28	4.30	25.96	25.32	58.50	51.61
		5-shot	0.424	79.07	17.76	3.18	22.12	23.05	47.62	46.13
		10-shot	0.381	81.08	16.63	2.29	20.51	22.08	41.50	43.87
	after SFT	**zero-shot**	0.198	88.00	11.60	0.40	9.62	13.31	21.09	28.06
Llama-3.1-8B-Instruct	**baseline**	**zero-shot**	0.468	71.86	26.44	1.69	23.08	29.87	47.62	38.71
		1-shot	0.408	78.99	19.65	1.36	25.00	26.95	43.20	38.06
		5-shot	0.347	84.70	14.84	0.46	19.55	23.05	37.76	41.29
		10-shot	0.316	84.46	15.04	0.50	21.15	19.48	31.29	38.39
	after SFT	**zero-shot**	0.211	87.97	10.90	1.13	8.97	13.31	21.09	33.23
Llama-3.2-1B	**baseline**	**zero-shot**	0.597	69.85	24.70	5.44	24.04	32.47	67.01	49.68
		1-shot	0.565	72.75	22.89	4.35	26.28	30.52	64.63	49.03
		5-shot	0.510	76.21	20.22	3.58	20.83	26.95	64.97	48.71
		10-shot	0.470	78.08	18.55	3.37	20.51	25.65	60.20	46.13
	after SFT	**zero-shot**	0.253	80.25	18.50	1.25	14.42	15.91	24.83	28.71
Llama-3.2-1B-Instruct	**baseline**	**zero-shot**	0.584	68.89	27.72	3.40	27.56	34.42	57.48	47.10
		1-shot	0.528	69.67	27.18	3.15	24.36	27.60	53.06	47.42
		5-shot	0.478	71.31	24.71	3.98	20.83	24.03	53.40	43.23
		10-shot	0.453	72.33	23.99	3.68	21.79	21.43	50.00	42.58
	after SFT	**zero-shot**	0.297	79.41	18.98	1.60	14.74	18.83	33.33	30.65
Llama-3.2-3B	**baseline**	**zero-shot**	0.553	73.17	19.66	7.17	23.72	35.06	60.20	48.71
		1-shot	0.522	77.05	19.30	3.65	27.88	30.52	59.86	48.39
		5-shot	0.427	80.67	16.54	2.79	22.12	28.25	45.92	46.13
		10-shot	0.413	82.73	14.40	2.88	25.00	27.60	44.56	44.19
	after SFT	**zero-shot**	0.240	81.13	16.89	1.99	9.29	14.94	24.49	31.61
Llama-3.2-3B-Instruct	**baseline**	**zero-shot**	0.512	76.63	21.98	1.39	25.64	34.09	47.62	54.84
		1-shot	0.527	77.26	21.54	1.20	30.45	30.84	49.66	57.10
		5-shot	0.456	80.35	18.61	1.04	24.36	28.57	42.86	55.48
		10-shot	0.424	81.68	17.20	1.12	24.68	27.27	38.78	52.26
	after SFT	**zero-shot**	0.209	82.89	15.21	1.90	9.62	12.34	26.87	22.90

Table 30: Error rates, incorrect choice distributions, and local negation confusion rates for the **Llama3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Llama-3.1-8B	**baseline**	**zero-shot**	0.569	66.43	8.64	24.93	32.69	31.49	59.18	33.55
		1-shot	0.519	73.85	5.20	20.95	40.71	35.71	48.30	33.55
		5-shot	0.448	85.84	3.89	10.27	43.27	31.82	49.32	34.52
		10-shot	0.385	88.25	4.12	7.63	38.14	27.92	40.48	33.55
	after SFT	**zero-shot**	0.328	69.81	18.84	11.35	22.76	20.78	35.71	15.81
Llama-3.1-8B-Instruct	**baseline**	**zero-shot**	0.486	71.45	14.85	13.70	33.01	25.97	56.12	29.03
		1-shot	0.402	67.26	17.16	15.58	28.85	21.10	36.39	25.48
		5-shot	0.353	77.53	11.24	11.24	27.56	19.81	39.12	26.77
		10-shot	0.349	83.41	9.09	7.50	31.41	19.48	39.12	30.32
	after SFT	**zero-shot**	0.270	65.69	21.70	12.61	14.74	14.61	29.25	15.16
Llama-3.2-1B	**baseline**	**zero-shot**	0.741	32.44	34.69	32.87	22.76	27.92	25.51	22.90
		1-shot	0.726	29.29	33.66	37.05	21.47	18.83	24.49	22.90
		5-shot	0.744	33.05	34.75	32.20	24.04	26.30	25.85	25.16
		10-shot	0.742	32.48	34.19	33.33	23.72	25.00	25.85	24.84
	after SFT	**zero-shot**	0.741	32.44	34.69	32.87	22.76	27.92	25.51	22.90
Llama-3.2-1B-Instruct	**baseline**	**zero-shot**	0.700	40.43	27.86	31.71	25.96	35.39	29.59	25.81
		1-shot	0.638	50.31	22.36	27.33	24.68	38.31	40.14	29.68
		5-shot	0.658	46.63	23.37	30.00	27.24	38.64	31.63	29.03
		10-shot	0.681	42.03	25.73	32.25	27.88	33.12	31.97	25.16
	after SFT	**zero-shot**	0.738	32.69	33.23	34.09	22.44	27.92	25.85	23.23
Llama-3.2-3B	**baseline**	**zero-shot**	0.626	58.81	18.76	22.43	27.24	41.56	47.96	35.48
		1-shot	0.581	67.21	13.80	18.99	33.65	39.61	44.90	42.90
		5-shot	0.526	79.94	10.71	9.35	37.18	38.64	51.02	46.77
		10-shot	0.512	83.41	9.30	7.29	38.78	36.69	50.34	50.32
	after SFT	**zero-shot**	0.636	47.51	26.68	25.81	22.76	36.36	35.37	30.32
Llama-3.2-3B-Instruct	**baseline**	**zero-shot**	0.512	79.26	9.91	10.84	31.09	36.69	65.65	35.16
		1-shot	0.466	80.07	10.22	9.71	28.53	30.19	60.88	35.16
		5-shot	0.509	76.64	8.88	14.49	30.77	29.55	67.35	34.52
		10-shot	0.509	74.61	7.79	17.60	32.05	28.25	65.65	31.94
	after SFT	**zero-shot**	0.418	58.82	10.25	30.93	17.63	18.18	48.64	18.06

Table 31: Error rates, incorrect choice distributions, and local negation confusion rates for the **Llama3** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Mistral-7B-v0.3	**baseline**	**zero-shot**	0.533	75.74	20.24	4.02	25.96	31.82	60.20	49.35
		1-shot	0.533	78.27	18.15	3.57	29.81	33.77	59.18	50.00
		5-shot	0.427	82.71	15.80	1.49	22.12	25.65	51.02	47.42
		10-shot	0.394	84.31	13.28	2.41	21.47	24.68	47.96	43.55
	after SFT	**zero-shot**	0.217	86.81	12.09	1.10	8.97	12.66	26.53	29.68
Mistral-7B-Instruct-v0.3	**baseline**	**zero-shot**	0.389	65.10	34.08	0.82	16.67	18.83	29.25	39.68
		1-shot	0.350	71.66	27.44	0.91	16.35	18.18	33.33	35.81
		5-shot	0.305	72.47	27.27	0.26	12.50	15.58	28.91	34.52
		10-shot	0.282	73.24	25.92	0.85	12.82	13.96	26.19	32.26
	after SFT	**zero-shot**	0.207	83.91	14.18	1.92	8.01	11.04	24.49	28.39
Mistral-Nemo-Base-2407 (12B)	**baseline**	**zero-shot**	0.540	75.62	21.44	2.94	25.64	33.44	60.20	50.00
		1-shot	0.509	76.48	20.72	2.80	27.24	30.19	56.80	47.10
		5-shot	0.399	81.11	16.70	2.19	21.15	23.70	42.86	46.13
		10-shot	0.346	82.11	16.74	1.15	18.91	21.75	32.65	43.87
Mistral-Nemo-Instruct-2407 (12B)	**baseline**	**zero-shot**	0.513	75.43	21.64	2.94	24.04	29.22	56.80	50.32
		1-shot	0.468	76.95	21.36	1.69	23.08	26.30	52.38	47.42
		5-shot	0.369	78.28	20.43	1.29	19.23	20.45	38.44	41.29
		10-shot	0.324	79.41	19.36	1.23	17.31	17.53	30.61	40.65
Mistral-Small-24B-Base-2501	**baseline**	**zero-shot**	0.526	72.70	21.57	5.73	24.68	29.87	60.20	43.87
		1-shot	0.502	78.52	18.48	3.00	26.60	29.87	58.16	48.71
		5-shot	0.394	82.70	14.69	2.62	20.51	24.35	45.24	44.84
		10-shot	0.348	83.37	14.58	2.05	17.63	21.75	37.76	42.90
Mistral-Small-24B-Instruct-2501	**baseline**	**zero-shot**	0.474	78.43	19.57	2.01	25.96	29.87	53.06	45.16
		1-shot	0.426	79.70	18.81	1.49	23.40	24.03	48.30	44.84
		5-shot	0.355	81.43	17.67	0.89	18.91	20.78	38.10	41.61
		10-shot	0.309	83.55	15.68	0.77	16.03	18.51	31.63	40.32

Table 32: Error rates, incorrect choice distributions, and local negation confusion rates for the **Mistral** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Mistral-7B-v0.3	**baseline**	**zero-shot**	0.516	51.23	17.08	31.69	24.68	20.78	37.41	26.45
		1-shot	0.521	53.42	10.20	36.38	27.88	23.38	36.73	27.10
		5-shot	0.370	73.61	7.73	18.67	25.96	21.43	31.63	33.23
		10-shot	0.336	73.82	7.78	18.40	25.00	19.81	26.19	31.29
	after SFT	**zero-shot**	0.520	54.57	24.85	20.58	27.88	29.87	33.67	25.81
Mistral-7B-Instruct-v0.3	**baseline**	**zero-shot**	0.343	77.37	14.32	8.31	26.28	20.78	34.35	28.39
		1-shot	0.355	69.42	20.09	10.49	28.53	23.05	29.25	20.97
		5-shot	0.347	71.00	17.81	11.19	25.64	22.08	31.29	22.90
		10-shot	0.351	72.46	20.77	6.77	24.68	22.08	32.65	25.81
	after SFT	**zero-shot**	0.308	77.58	17.01	5.41	23.72	19.48	35.37	20.32
Mistral-Nemo-Base-2407 (12B)	**baseline**	**zero-shot**	0.588	64.10	15.25	20.65	35.90	33.44	49.66	36.77
		1-shot	0.488	83.41	10.24	6.34	39.10	33.12	54.42	41.61
		5-shot	0.438	89.49	6.34	4.17	39.10	32.47	48.98	41.29
		10-shot	0.400	89.88	6.15	3.97	33.97	29.55	40.14	44.52
Mistral-Nemo-Instruct-2407 (12B)	**baseline**	**zero-shot**	0.396	89.38	7.82	2.81	33.33	25.32	52.38	35.48
		1-shot	0.368	89.22	8.62	2.16	30.77	24.68	46.94	33.55
		5-shot	0.345	89.89	8.51	1.61	32.05	25.65	39.80	30.65
		10-shot	0.338	90.85	7.75	1.41	32.69	29.22	35.37	29.35
Mistral-Small-24B-Base-2501	**baseline**	**zero-shot**	0.432	62.20	18.72	19.08	29.81	21.75	39.12	20.65
		1-shot	0.381	73.75	19.38	6.88	32.69	25.65	32.65	24.84
		5-shot	0.274	85.80	9.86	4.35	25.32	20.13	21.77	29.35
		10-shot	0.219	90.58	6.88	2.54	23.40	17.86	9.86	30.00
Mistral-Small-24B-Instruct-2501	**baseline**	**zero-shot**	0.314	78.03	15.91	6.06	27.24	19.81	31.63	22.58
		1-shot	0.290	81.15	14.48	4.37	30.77	18.83	27.89	19.68
		5-shot	0.257	86.11	11.42	2.47	30.13	23.38	17.01	20.32
		10-shot	0.217	89.05	7.66	3.28	27.24	19.81	8.50	23.55

Table 33: Error rates, incorrect choice distributions, and local negation confusion rates for the **Mistral** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Mistral-7B-v0.3	**baseline**	**zero-shot**	0.521	74.12	20.24	5.63	26.60	32.14	55.44	45.81
		1-shot	0.530	78.29	17.81	3.89	29.81	32.47	61.22	48.39
		5-shot	0.432	82.94	15.23	1.83	23.08	24.35	52.38	48.71
		10-shot	0.398	83.07	14.54	2.39	21.47	24.35	47.28	43.87
	after SFT	**zero-shot**	0.208	87.02	11.45	1.53	9.62	12.34	25.51	27.42
Mistral-7B-Instruct-v0.3	**baseline**	**zero-shot**	0.338	68.78	30.52	0.70	17.31	18.18	23.47	36.77
		1-shot	0.337	69.88	28.94	1.18	15.06	19.16	31.63	31.61
		5-shot	0.296	73.19	26.54	0.27	13.14	16.56	28.57	31.29
		10-shot	0.282	73.03	26.12	0.84	13.14	13.64	25.85	32.58
	after SFT	**zero-shot**	0.205	83.40	15.06	1.54	8.01	9.42	23.81	29.68
Mistral-Nemo-Base-2407 (12B)	**baseline**	**zero-shot**	0.535	71.81	24.04	4.15	25.96	30.84	53.74	48.39
		1-shot	0.504	75.28	20.79	3.94	27.88	30.19	53.40	45.48
		5-shot	0.409	78.68	18.41	2.91	21.47	24.35	42.52	44.84
		10-shot	0.355	81.43	17.00	1.57	19.87	22.40	33.33	43.55
Mistral-Nemo-Instruct-2407 (12B)	**baseline**	**zero-shot**	0.489	71.47	25.28	3.24	24.36	29.55	48.30	42.58
		1-shot	0.450	75.35	22.36	2.29	21.79	26.95	45.58	46.13
		5-shot	0.362	78.73	19.96	1.32	18.91	20.45	36.05	42.26
		10-shot	0.326	79.56	19.46	0.97	16.99	19.16	30.27	40.65
Mistral-Small-24B-Base-2501	**baseline**	**zero-shot**	0.484	70.49	23.44	6.07	25.00	31.82	50.00	34.52
		1-shot	0.466	76.32	18.91	4.77	25.64	28.25	54.08	39.35
		5-shot	0.385	81.24	15.26	3.51	19.23	23.70	41.84	44.52
		10-shot	0.339	83.14	14.99	1.87	17.31	21.10	36.05	41.94
Mistral-Small-24B-Instruct-2501	**baseline**	**zero-shot**	0.420	76.94	20.60	2.46	25.32	32.79	44.22	31.29
		1-shot	0.360	77.97	20.04	1.98	21.79	24.03	38.78	31.61
		5-shot	0.328	82.57	16.95	0.48	17.95	19.48	35.03	39.35
		10-shot	0.297	82.62	16.31	1.07	14.74	17.21	30.27	39.03

Table 34: Error rates, incorrect choice distributions, and local negation confusion rates for the **Mistral** family under zero-shot, few-shot, and SFT conditions, evaluated in the **completion-based setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Incorrect Choice Distribution			Local Negation Confusion Rate			
				Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
Mistral-7B-v0.3	**baseline**	**zero-shot**	0.498	59.39	15.13	25.48	24.04	25.97	43.88	28.71
		1-shot	0.543	48.03	9.64	42.34	26.28	18.51	36.39	26.77
		5-shot	0.364	71.02	7.63	21.35	24.36	20.45	30.95	30.97
		10-shot	0.339	72.20	7.71	20.09	22.76	20.13	25.85	32.26
	after SFT	**zero-shot**	0.523	51.75	23.37	24.89	26.92	28.25	30.95	25.48
Mistral-7B-Instruct-v0.3	**baseline**	**zero-shot**	0.366	80.04	10.63	9.33	31.73	20.78	35.71	32.58
		1-shot	0.356	71.71	16.48	11.80	26.60	24.35	30.61	23.87
		5-shot	0.355	69.42	16.07	14.51	24.68	21.10	31.97	24.19
		10-shot	0.355	71.21	16.52	12.28	25.32	23.05	31.97	24.19
	after SFT	**zero-shot**	0.290	80.27	15.62	4.11	23.72	20.45	30.61	21.29
Mistral-Nemo-Base-2407 (12B)	**baseline**	**zero-shot**	0.578	66.67	14.54	18.79	37.18	40.91	47.62	33.55
		1-shot	0.497	82.30	11.16	6.54	38.78	35.06	55.10	40.32
		5-shot	0.443	88.35	5.91	5.73	38.46	30.52	49.32	43.23
		10-shot	0.400	89.09	6.75	4.17	33.97	28.25	40.48	44.19
Mistral-Nemo-Instruct-2407 (12B)	**baseline**	**zero-shot**	0.374	85.56	10.83	3.61	29.49	27.92	42.86	31.94
		1-shot	0.364	86.93	11.33	1.74	30.13	27.92	39.12	33.55
		5-shot	0.344	91.01	7.83	1.15	31.41	26.95	39.80	31.29
		10-shot	0.340	90.91	8.16	0.93	34.29	27.92	36.05	29.35
Mistral-Small-24B-Base-2501	**baseline**	**zero-shot**	0.425	61.57	17.91	20.52	31.41	25.00	29.25	22.26
		1-shot	0.357	70.00	21.78	8.22	29.81	20.13	29.59	23.55
		5-shot	0.265	83.23	10.78	5.99	24.04	20.13	16.33	30.00
		10-shot	0.215	89.67	6.64	3.69	22.12	17.53	8.84	30.32
Mistral-Small-24B-Instruct-2501	**baseline**	**zero-shot**	0.271	80.41	13.16	6.43	32.37	22.73	11.22	22.90
		1-shot	0.243	83.01	13.4	3.59	27.56	21.43	14.97	18.71
		5-shot	0.240	87.09	9.93	2.98	27.88	21.10	12.93	23.55
		10-shot	0.195	87.40	9.35	3.25	24.36	16.88	4.76	23.55

Table 35: Error rates, incorrect choice distributions, and local negation confusion rates for the **Mistral** family under zero-shot, few-shot, and SFT conditions, evaluated in the **option-selection setting** using **detailed instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Answer Format Wrong	Incorrect Choice Distribution			Local Negation Confusion Rate			
					Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
GPT-4o mini	**baseline**	**zero-shot**	0.287	0	68.23	31.77	0	19.87	11.36	31.97	18.06
		1-shot	0.251	0	68.99	30.38	0.63	20.51	10.71	26.87	13.55
		5-shot	0.231	0	54.64	45.02	0.34	18.27	5.52	17.69	10.65
		10-shot	0.205	0	52.51	47.10	0.39	16.67	7.14	11.90	8.71
GPT-4o	**baseline**	**zero-shot**	0.218	0	80.73	19.27	0	18.91	8.44	33.67	12.26
		1-shot	0.209	0	88.26	11.36	0.38	23.08	8.77	27.55	17.10
		5-shot	0.198	0	87.55	12.05	0.40	22.12	11.36	17.35	20.32
		10-shot	0.175	0	85.45	14.09	0.45	20.19	10.39	12.24	18.39
GPT-4.1 mini	**baseline**	**zero-shot**	0.244	0	90.88	9.12	0	14.42	5.19	51.36	21.61
		1-shot	0.204	0	88.33	11.67	0	18.59	7.14	30.27	18.71
		5-shot	0.159	0	84.00	15.50	0.50	15.71	5.52	15.31	18.39
		10-shot	0.151	0	77.89	21.58	0.53	13.78	5.84	8.84	19.68
GPT-4.1	**baseline**	**zero-shot**	0.119	0	78.67	20.00	1.33	10.90	6.49	15.65	5.81
		1-shot	0.136	0	81.87	18.13	0	17.31	9.09	10.88	8.39
		5-shot	0.128	0	81.99	18.01	0	18.27	9.42	5.10	10.00
		10-shot	0.109	0	79.71	19.57	0.72	15.06	8.44	2.04	10.00
Haiku 4.5	**baseline**	**zero-shot**	0.228	0	88.54	11.46	0	22.12	15.26	38.78	8.06
		1-shot	0.226	0	89.12	9.82	1.05	24.36	15.26	32.65	11.29
		5-shot	0.190	0	90.79	9.21	0	29.49	17.53	13.61	10.00
		10-shot	0.172	0	91.71	7.83	0.46	25.64	17.86	7.48	13.55
Sonnet 4.5	**baseline**	**zero-shot**	0.125	0	85.35	12.74	1.91	16.35	14.61	8.50	4.19
		1-shot	0.122	0	92.86	6.49	0.65	19.23	16.56	6.12	4.52
		5-shot	0.127	0	81.25	12.50	6.25	16.67	12.99	4.76	7.74
		10-shot	0.118	15	82.09	10.45	7.46	13.46	9.42	5.44	7.42

Table 36: Error rates, incorrect choice distributions, and local negation confusion rates for the **API** models under zero-shot and few-shot conditions, using **definition instruction.**

Model	Training Setting	N Shot	Error Rate (1-acc)	Answer Format Wrong	Incorrect Choice Distribution			Local Negation Confusion Rate			
					Local Negation (%)	Contra-diction (%)	Para-phrase (%)	Relative Clause (%)	Participle Clause (%)	Compound Sentence (%)	Adverbial Clause (%)
GPT-4o mini	**baseline**	**zero-shot**	0.244	0	82.14	17.21	0.65	14.42	12.66	38.10	18.39
		1-shot	0.226	0	81.05	17.89	1.05	19.55	8.12	33.33	15.16
		5-shot	0.224	0	63.60	35.69	0.71	19.55	4.55	19.73	15.16
		10-shot	0.210	0	60.38	39.25	0.38	16.99	7.14	15.65	12.58
GPT-4o	**baseline**	**zero-shot**	0.137	0	74.57	25.43	0	12.82	7.47	12.24	9.68
		1-shot	0.147	0	79.46	20.00	0.54	18.27	10.06	13.27	6.45
		5-shot	0.172	0	84.79	13.82	1.38	19.87	13.31	12.93	13.87
		10-shot	0.185	0	83.69	15.45	0.86	24.68	12.01	11.22	15.48
GPT-4.1 mini	**baseline**	**zero-shot**	0.098	0	90.32	9.68	0	3.53	3.57	16.67	13.23
		1-shot	0.133	0	84.52	15.48	0	14.42	5.84	18.37	8.06
		5-shot	0.124	1	83.23	15.48	1.29	12.50	4.87	11.56	13.23
		10-shot	0.130	0	75.61	23.78	0.61	12.50	4.22	11.56	12.26
GPT-4.1	**baseline**	**zero-shot**	0.064	0	75.31	24.69	0	6.73	3.90	6.12	3.23
		1-shot	0.082	0	91.26	8.74	0	13.14	8.12	4.42	4.84
		5-shot	0.098	0	81.30	18.70	0	13.14	9.09	3.74	6.45
		10-shot	0.098	0	84.68	15.32	0	13.78	8.44	1.70	10.00
Haiku 4.5	**baseline**	**zero-shot**	0.135	1	89.94	10.06	0	16.99	15.58	13.27	3.87
		1-shot	0.128	0	87.58	9.32	3.11	16.67	12.01	11.56	5.81
		5-shot	0.126	0	93.71	5.66	0.63	22.12	14.29	6.80	5.16
		10-shot	0.118	0	94.63	5.37	0	19.55	13.64	5.10	7.42
Sonnet 4.5	**baseline**	**zero-shot**	0.085	1	91.51	7.55	0.94	10.58	12.34	6.12	2.58
		1-shot	0.083	0	91.43	8.57	0	12.50	13.96	2.04	2.58
		5-shot	0.121	26	82.68	14.17	3.15	14.74	12.01	3.74	3.55
		10-shot	0.188	150	75.86	13.79	10.34	8.65	7.14	1.70	3.87

Table 37: Error rates, incorrect choice distributions, and local negation confusion rates for the **API** models under zero-shot and few-shot conditions, using **detailed instruction.**