Thunder-KoNUBench: A Corpus-Aligned Benchmark for Korean Negation Understanding

Sungmok Jung*[1], Yeonkyoung So*[1], Joonhak Lee*[1], Sangho Kim[1], Yelim Ahn[1], Jaejin Lee[1,2]

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

Abstract

Although negation is known to challenge large language models (LLMs), benchmarks for evaluating negation understanding—especially in Korean—are scarce. We conduct a corpus-based analysis of Korean negation and show that LLM performance degrades under negation. We then introduce *Thunder-KoNUBench*, a sentence-level negation understanding benchmark that reflects the empirical distribution of Korean negation phenomena. Evaluating 47 LLMs on Thunder-KoNUBench, we analyze the effects of model size and instruction tuning, and perform error analysis to better understand model behavior. We further show that fine-tuning on Thunder-KoNUBench improves negation understanding and broader contextual comprehension in Korean [1].

1 Introduction

Negation is a fundamental operation in natural language that reverses the meaning of an expression into its opposite. It enables a variety of linguistic functions, including contradiction, inability, and denial, while also supporting logical reasoning by specifying what is false or absent. Therefore, effectively processing negation is crucial for strong language comprehension.

However, many studies indicate that large language models (LLMs) often struggle with handling negation (Jumelet and Hupkes, 2018; Kassner and Schütze, 2020; Truong et al., 2023). Similar difficulties have been reported in vision-language models (VLMs) (Alhamoud et al., 2025; Park et al., 2025). In response to these challenges, several datasets (García-Ferrero et al., 2023; Hossain et al., 2022b; Hartmann et al., 2021; Ravichander et al., 2022) and training strategies (Hosseini et al., 2021; Singh et al., 2023; Han et al., 2025; Park et al., 2025) have been proposed to enhance or evaluate negation handling in both LLMs and VLMs. Unfortunately, these efforts have predominantly focused on English. Despite the development of recent benchmarks for negation phenomena in languages such as Czech, German, Ukrainian, Bulgarian, and French (Hartmann et al., 2021; Vrabcová et al., 2025), there is a notable lack of benchmarks and evaluation studies concerning negation in Korean. Consequently, little is known about whether similar limitations exist in Korean, and no benchmark currently exists that systematically evaluates sentence-level negation understanding in Korean.

To address this issue, we first analyze the characteristics and distribution of negation in Korean and examine whether existing LLMs suffer performance degradation in the presence of negation. We then introduce Thunder-KoNUBench, a multiple-choice benchmark that reflects the empirical distribution of Korean negation phenomena. Finally, we evaluate a wide range of Korean and non-Korean LLMs on Thunder-KoNUBench and investigate fine-tuning strategies for improving sentence-level negation understanding.

The main contributions of this paper are summarized as follows:

- We demonstrate that LLMs, both Korean and non-Korean, encounter difficulties when handling negation in Korean, even in simple sentence-level tasks. They experience significant performance degradation when required to reason with negation.
- We conduct a comprehensive corpus study of Korean negation, analyzing the statistical distribution of major negation types and the sentence structures in which they are used.
- We introduce *Thunder-KoNUBench*, a sentence-level multiple-choice benchmark that system-

*These authors contributed equally to this work.
[1]Our code and dataset are publicly available on Github and HuggingFace.
https://github.com/mcrl/Thunder-KoNUBench
https://huggingface.co/datasets/thunder-research-group/SNU_Thunder-KoNUBench

Findings of the Association for Computational Linguistics: ACL 2026, pages 6491–6518
July 2-7, 2026

atically reflects the empirical distribution of Korean negation phenomena.

- We evaluate 47 LLMs (18 Korean models and 29 non-Korean models) using Thunder-KoNUBench to analyze the effects of model size and instruction tuning on negation understanding.
- Our experiments indicate that supervised fine-tuning on Thunder-KoNUBench enhances the models' contextual understanding in Korean, and that cloze-style supervision is more effective than symbol-style supervision for learning sentence-level negation.

2 Related Work

2.1 Challenges of LLMs in Negation Handling

Numerous studies have demonstrated that negation presents considerable challenges for language models. Researches indicate that pretrained language models (PLMs) often struggle to differentiate between negated and non-negated questions (Kassner and Schütze, 2020; Ettinger, 2020). These findings have been questioned as to whether the observed performance degradation truly reflects an inability to handle negation, or rather stems from difficulties in contextual or factual reasoning (Gubelmann and Handschuh, 2022). However, subsequent work using more fine-grained evaluation settings shows that some PLMs still struggle to handle negation robustly under controlled conditions(Kletz et al., 2023). Furthermore, their inability to understand synonym-antonym relationships contributes to failures in reasoning under negation (Truong et al., 2023).

These shortcomings have been identified as key factors leading to degraded performance across various downstream NLP tasks, including machine translation (Hossain et al., 2020a; Tang et al., 2021), information extraction (Grivas et al., 2020), and sentiment analysis (Barnes et al., 2021). More recently, similar findings have been observed in multimodal language models, such as VLMs, where improved understanding of negation significantly impacts performance (Alhamoud et al., 2025). Enhancing awareness of negation has also been shown to yield measurable improvements (Park et al., 2025).

2.2 Datasets and Benchmarks for Negation

Previous studies have highlighted that both NLP corpora and downstream tasks contain very few instances of negation (Hossain et al., 2020b, 2022a). It has led to an increasing focus on benchmarks specifically designed to evaluate negation comprehension. NaN-NLI (Truong et al., 2022) and ScoNe (Ravichander et al., 2022) assess whether models can accurately determine premise–hypothesis relationships in the presence of negation. Additionally, CONDAQA (Ravichander et al., 2022) evaluates whether models understand the semantic implications of negated statements. Thunder-NUBench (So et al., 2026) tests if models can identify the correct standard negation of an original sentence, with a clearly defined scope of negation. However, these efforts are primarily focused on English and a few high-resource languages.

Although multilingual benchmarks have been proposed (Hartmann et al., 2021; Vrabcová et al., 2025), there are no comparable resources or systematic studies of negation in low-resource languages such as Korean. Among existing Korean benchmarks, KoBest (Jang et al., 2022) introduces SentiNeg, a sentiment analysis task involving negation, but its coverage of negation is limited, and the sentences are overly simplistic. KMMLU (Son et al., 2025) includes a subset of questions that contain negation, yet these items are mostly easy declarative questions, failing to capture the impact of negation on model performance meaningfully. To address this gap, we examine the impact of negation in Korean and introduce a benchmark specifically targeting negation understanding, carefully designed to reflect the linguistic characteristics of Korean negation.

3 Korean Negation

Negation in Korean is expressed through various morphological, syntactic, and semantic mechanisms. Despite its complexity, there has been limited empirical investigation into its statistical properties, and few analyses have explored its impact on the performance of LLMs. In this section, we define the negation in Korean, analyze their statistical distribution, and evaluate LLMs' robustness to Korean negation.

3.1 Definition of Negation in Korean

Logically, negation is an operation that reverses the meaning of an original sentence. If a sentence expresses the proposition P, its logical negation corresponds to $\neg P$. This complementary relation-

ship is distinct from contradiction, which simply refers to the incompatibility between two propositions. To implement logical negation, Thunder-NUBench (So et al., 2026) introduces standard negation as a logical operation that reverses the truth value of the entire sentence. This is different from local negation, which provides only a partial negation of the overall meaning of the sentence. In this work, while adopting the core concepts of Thunder-NUBench, we redefine standard negation and local negation to systematically reflect the linguistic properties of Korean.

Standard Negation. Standard negation is defined as a recursive operation applied to the logical structure among main clauses. It first negates the logical relationship (e.g., conjunction, disjunction, implication) connecting the main clauses and is then recursively applied to each main clause. This process continues until only atomic propositions remain, at which point standard negation is realized by negating the predicate of each atomic proposition. This concept is grounded in two linguistic assumptions: (1) a main clause can express a complete meaning on its own and thus functions as an atomic proposition, and (2) the predicate serves as the head of the clause (Miller and Miller, 2011).

When negating predicates, diverse negative markers in Korean may be used. Korean employs a variety of negative markers such as "안", "못", "-지 않-", "-지 못하-", and "-지 말-". Each of these markers conveys a distinct semantic nuance and is subject to specific syntactic and semantic constraints on its usage (see Appendix A for a detailed linguistic description of Korean negation). When negating the predicate of a clause, these markers may be used depending on the predicate type and sentence context.

Local Negation. Local negation refers to negation applied locally, producing a partial negation that does not reverse the meaning of original sentence completely. This includes cases where negation targets a dependent clause or only one of multiple main clauses. In Thunder-NUBench (So et al., 2026), the classification of local negation based on English sentence structure is divided into several categories: negation in relative clauses, participial clauses, adverbial clauses, and compound sentences with local negation.

In this paper, we redefine the typology of local negation in terms of Korean sentence structure. Clause combinations in Korean take diverse structural forms. Korean clauses may be linked via connective endings or conjunctive particles, or embedded within another clause as part of a larger syntactic structure (see Appendix B for a detailed discussion of Korean sentence structure). Our classification includes negation in noun clauses, adnominal clauses, adverbial clauses, quotation clauses, subordinate clauses, and coordinated sentences with local negation. Examples of each category are provided in Table 1.

Category	Example
Noun clauses Negation	농부들이 비가 오기를 기다린다. → 농부들이 비가 오지 않기를 기다린다. (The farmers are waiting for it to rain. → The farmers are waiting for it not to rain.)
Adnominal clauses Negation	내가 태어난 1950년에 전쟁이 발발했다. → 내가 태어나지 않은 1950년에 전쟁이 발발했다. (In 1950, the year I was born, a war broke out. → In 1950, the year I was not born, a war broke out.)
Quotation clauses Negation	나는 그가 나의 연설에 만족했다고 들었다. → 나는 그가 나의 연설에 만족하지 않았다고 들었다. (I heard that he was satisfied with my speech. → I heard that he was not satisfied with my speech.)
Subordinate clauses Negation	꽃이 잘 자라도록 나는 물을 주었다. → 꽃이 잘 자라지 못하도록 나는 물을 주었다. (I watered the flowers so that they would grow well. → I watered the flowers so that they would not grow well.)
Adverbial clauses Negation	우리는 그녀가 지나가도록 길을 비켜 주었다. → 우리는 그녀가 지나가지 못하도록 길을 비켜 주었다. (We stepped aside to let her pass. → We stepped aside so that she could not pass.)
Coordinated sentence with local negation	나는 밥을 먹었고, 친구는 빵을 먹었다. → 나는 밥을 안 먹었고, 친구는 빵을 먹었다. (I ate rice, and my friend ate bread. → I didn't eat rice, and my friend ate bread.)

Table 1: Examples of Local Negation Typology in Korean.

3.2 Statistics of Korean Negation

We analyze the statistical distribution of negation in Korean using a large-scale corpus from the OpenAI Dataset Project (AI-Hub, S.Korea). The dataset we utilize is titled "한국어 성능이 개선된 초거대 AI 언어모델 개발 및 데이터" (Dataset and Large Language Models with Improved Korean Performance). It includes both spoken and written-style texts and covers a wide range of topics, reflecting diverse real-world language use. Further details on the corpus are provided in Appendix C.

To begin the analysis, we segment all sentences in the corpus using `split_sentences` function in the `KSS` library. We then randomly sample 30,000

	안 계열(*an* type)		못 계열(*mot* type)		말다(*malda*)	Total
	Short-Form	Long-Form	Short-Form	Long-Form		
Instance (%)	795 (25.16%)	1,598 (50.57%)	174 (5.51%)	473 (14.97%)	120 (3.8%)	3,160 (100%)

	Adnominal Clause	Adverbial Clause	Noun Clause	Quotation Clause	Subordinate Clause	Main Clause	Total
Instance (%)	1,022 (32.34%)	624 (19.75%)	98 (3.1%)	132 (4.18%)	241 (7.63%)	1,043 (33.01%)	3,160 (100%)

Table 2: Distribution of syntactic negation in Korean. The upper part shows the distribution of negation types, while the lower part presents the distribution of clause types in which negation appears.

	KMMLU		BoolQ	
	Negative	Affirmative	Original	Negated
Korean Models	63.0	66.0	73.2	58.6
Non-Korean Models	62.7	63.6	62.8	50.2
All Models	62.8	64.6	67.2	53.7

Table 3: Average performance of models on KMMLU and BoolQ with and without negation.

sentences using a fixed random seed 2025. Upon inspection, we find that some sentences are excessively long to be treated as single sentences, so we remove sentences longer than 400 characters, resulting in a total of 29,476 sentences. Next, we apply a rule-based method to identify sentences containing negation. Then, the authors manually verify the candidates to ensure they accurately represent genuine instances of negation.

Through this verification process, we identify 3,160 negative sentences, indicating that approximately 10.7% of the Korean corpus consists of negative sentences. Furthermore, we examine the statistical distribution of these negations. Table 2 presents the distribution of each type of negation in the corpus, along with the types of sentences in which each negation expression appears. For a detailed linguistic description of Korean sentence structure, please see Appendix B.

3.3 LLMs on Korean negation

LLM performance on KMMLU. KMMLU is a multiple-choice dataset featuring 45 categories that reflect Korea's cultural and regional characteristics. While the benchmark includes questions involving negation, it does not sufficiently address whether language models genuinely struggle with negation in Korean. In fact, the authors pointed out that models tend to perform better on items with negation compared to items without negation in the KMMLU test set. Their analysis suggests that this result arises because the negative questions in KMMLU focus primarily on relatively simple declarative knowledge, rather than procedural knowledge (Son et al., 2025).

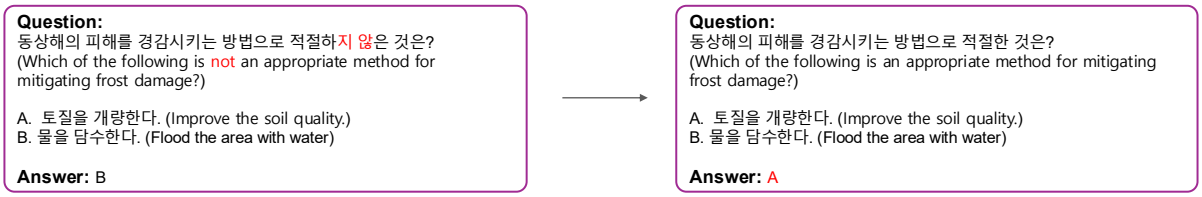
To investigate whether models face difficulties with negation even in simple declarative questions, we conducted an additional experiment. First, we extracted a total of 7,153 questions containing negation from KMMLU using a rule-based detector and converted them into binary-choice items for evaluation. Next, we transformed these items into affirmative sentences and re-evaluated the models. Figure 1a provides an example of this process. The results, shown in Table 3, indicate that both Korean and non-Korean models achieve higher performance on the affirmative versions on average. These findings suggest that models struggle to handle negation, even in simple declarative questions.

LLM performance on KoBest BoolQ. KoBest BoolQ is a binary-choice dataset featured in KoBest (Jang et al., 2022), in which models are required to respond with either *True* or *False* based on their understanding of the provided context. This task is more challenging than the straightforward declarative questions in KMMLU, as it requires context-sensitive reasoning. To investigate the impact of negation, we transformed all 1,404 questions into their negated forms and assessed the models' performance. An example of this process is shown in Figure 1b. The results indicate a significant decline in performance across all 43 models, reinforcing the previous finding that LLMs struggle to reason under negation (Truong et al., 2023), and this issue is also relevant in Korean.

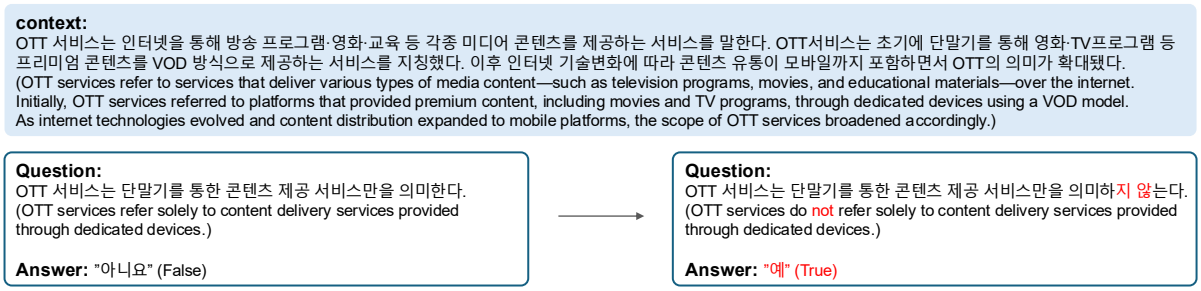
4 Thunder-KoNUBench

4.1 Task Overview

Thunder-KoNUBench is a multiple-choice dataset consisting of 4,784 instances, designed to evaluate sentence-level understanding of negation in Korean (see Figure 2 for an example). In terms of the scope and categorization of negation,



(a) An example of binary-format KMMLU questions containing negation (left) and their affirmative counterparts (right).



(b) An example of KoBest BoolQ questions in their original (left) and negated (right) forms.

Figure 1: Illustration of negation-induced performance evaluation on KMMLU and KoBest BoolQ.

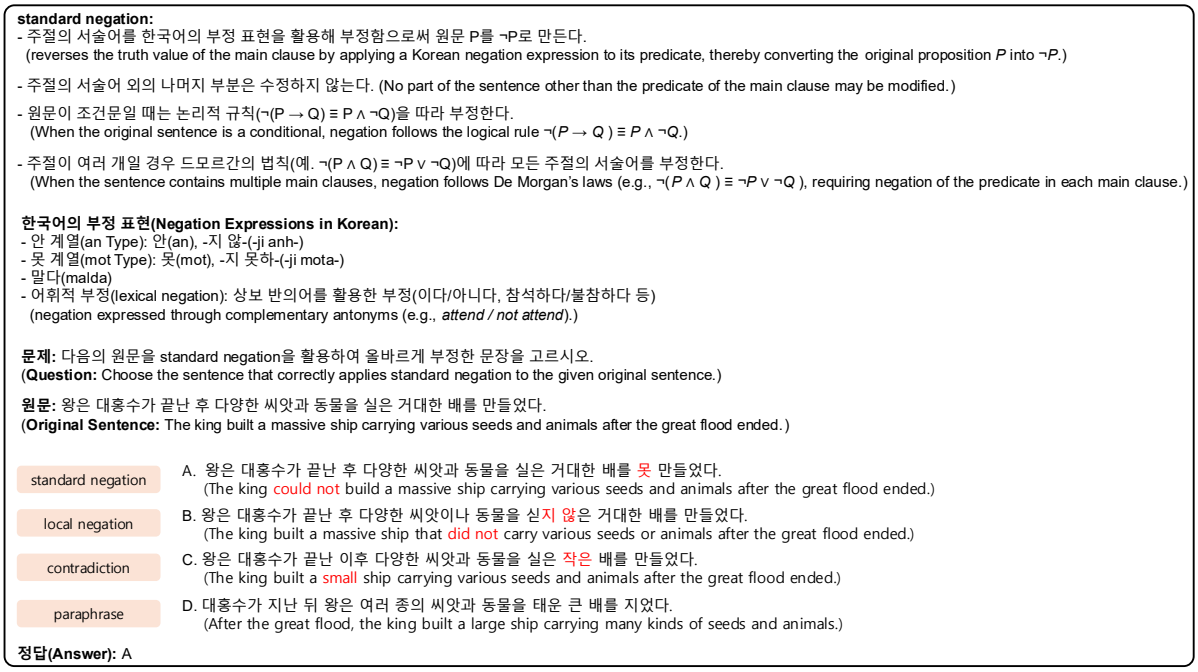


Figure 2: An example instance from Thunder-KoNUBench.

Thunder-KoNUBench closely follows the structure of Thunder-NUBench (So et al., 2026), while appropriately adapting it to reflect the linguistic properties of the Korean language. Specifically, models are required to select the correct standard negation of the original sentence from various distractors, which include local negation, contradiction, and paraphrase. For a detailed description of each category, see Table 4. To choose the correct answer, models must identify the main clause and its primary predicate and then apply the appropriate negation to it.

Distributions of negation. Table 5 shows the distribution of negation in Thunder-KoNUBench. We analyze the types of negation present in both standard and local contexts, as well as the clause types in which local negation occurs. Since the KL divergence from the distribution observed in the corpus (Table 2) is very small, we conclude that our benchmark closely aligns with the distributional

Category	Description
Standard Negation	Sentences that negate the main predicate of the main clause, thereby reversing the truth value of the original sentence. The main predicate is negated using Korean syntactic negation or complementary antonyms . This category constitutes the correct answer choice in Thunder-KoNUBench.
Local Negation	Sentences in which negation applies locally—either to a predicate in a dependent clause or to only one of multiple main clauses—thereby producing a partial negation that does not reverse the overall sentence meaning.
Contradiction	Sentences whose meaning is incompatible with the original sentence, but which do not use syntactic or lexical negation. The meaning is altered through gradable antonyms, different numerical values, changes of entity, or other semantic shifts.
Paraphrase	Sentences that preserve the meaning of the original sentence while changing its surface form. This may involve using synonyms or altering the syntactic structure, resulting in sentences that must be true whenever the original sentence is true.

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

	안 계열(*an* type)		못 계열(*mot* type)		말다(*malda*)	**Total**	$D_{\mathrm{KL}}(P\|Q)$
	Short-Form	**Long-Form**	**Short-Form**	**Long-Form**			
Instance(%)	1,991 (20.92%)	5,377 (56.49%)	671 (7.05%)	1,210 (12.71%)	269 (2.83%)	9,518 (100%)	0.007430

	Adnominal Clause	**Adverbial Clause**	**Noun Clause**	**Quotation Clause**	**Subordinate Clause**	**Coordinated Sentence**	**Total**	$D_{\mathrm{KL}}(P\|Q)$
Instance(%)	1,433 (33.73%)	891 (20.97%)	134 (3.15%)	152 (3.58%)	243 (5.72%)	1,395 (32.84%)	4,248 (100%)	0.413696

Table 5: Distribution of negation in Thunder-KoNUBench. The upper part shows the distribution of negation types, while the lower part presents the distribution of clause types in which negation appears under local negation. The KL divergence is computed as $D_{\mathrm{KL}}(P\|Q)$, where P denotes the distribution observed in the Korean corpus in Section 3.2 and Q denotes the distribution in Thunder-KoNUBench.

Split	**Train**	**Validation**	**Test**	**Total**
Count	2,500	1,000	1,284	4,784

Table 6: Thunder-KoNUBench statistics.

Max	**Min**	**Median**	**Average**
100.0	88.0	98.0	97.6

Table 7: Human evaluation results on Thunder-KoNUBench.

properties of negation in Korean.

4.2 Construction of Thunder-KoNUBench

The Thunder-KoNUBench dataset is developed through three main stages: (1) pre-processing of original sentences, (2) generating each choice, and (3) review. Detailed prompt examples used in the construction process are provided in Appendix D.

Pre-processing of original sentences. We crawl the Korean Wikipedia (ko.wikipedia, 2025) and split the text into pairs of sentences using a rule-based method. Then, we employ the OpenAI API (OpenAI, 2025) to merge each pair into a single, well-formed sentence. This approach of splitting the text into two-sentence units before merging them is designed to avoid overly simplistic original sentences. The authors conduct a manual verification process to correct any grammatical errors and unnatural expressions, thereby finalizing the set of original sentences.

Generation of choices. Each original sentence is paired with four options: standard negation, local negation, contradiction, and paraphrase. Descriptions of these options are provided in Table 4. These four options serve as candidates for the negated form of the original sentence, and the model must select the standard negation by following the instructions and demonstrating an understanding of sentence structure and Korean negation expressions. Standard negation and local negation are created manually by the authors, without the use of language models. When language models are employed, the generated sentences often fail to negate the intended part correctly or introduce unintended modifications elsewhere. Since Thunder-KoNUBench is designed to assess whether a model can accurately identify the main clause's predicate and negate it appropriately, the authors manually construct these two categories. In contrast, the contradiction and paraphrase options are initially generated using the OpenAI API (OpenAI, 2025) and are subsequently refined through a thorough review process conducted by the authors.

Review process. The construction of the original sentences and all response options undergoes thorough cross-checking among the authors. For each item, a different author, independent of the creator, verifies its correctness. Any disagreements are addressed collectively during regular meetings (see Appendix E for details on the review process and inter-annotator agreement). This process ensures that all authors reach a consensus on every item, which upholds the quality of the final dataset. Table 6 displays the statistics of the final dataset.

4.3 Human Evaluation

After creating the dataset, we conduct a human evaluation to verify its reliability and establish a baseline for human performance. We recruit ten participants who are not part of our research group and ask them to solve 50 questions randomly selected from the test set. To ensure the evaluation's validity, we restrict all participants' internet access to prevent the use of external resources.

The evaluation is conducted by asking participants to select the option they consider correct for each question. Performance is measured using standard accuracy; since the test consists of 50 questions, each correct answer is assigned 2 points (and 0 points for incorrect answers), resulting in a total score out of 100. The human performance results on Thunder-KoNUBench are presented in Table 7. The consistently high human performance suggests that our benchmark is internally consistent and linguistically sound, serving as a reliable testbed for model evaluation.

5 Experiments

5.1 Experimental Setup

Models. We conduct experiments on 47 large language models (LLMs), including several variants of Qwen 3 (Yang et al., 2025), Mistral (Jiang et al., 2023), Llama 3.1 and Llama 3.2 (Grattafiori et al., 2024), GPT-4.1 (Achiam et al., 2023), and Claude-Opus 4.5 (ANTHROPIC, 2025). Our evaluation also includes nearly all major Korean LLMs released to date, such as Kanana 1.5 (Bak et al., 2025), mi:dm 2.0 (KT, 2025), hyperclobaX (Team-HyperCLOVA, 2025), EXAONE-4.0 (LG-Research et al., 2025a), EXAONE-Deep (LG-Research et al., 2025b), and A.X 4.0 (SKT, 2025), along with many widely used non-Korean models. More details can be found in Appendix F. This comprehensive selection ensures that our results accurately reflect the performance of both domestic and international state-of-the-art LLMs.

Evaluation method. We evaluate models using the LM Evaluation Harness (Sutawika et al., 2025) in two standard settings for Multiple-Choice Question Answering (MCQA): *cloze* and *symbol*. In the cloze setting, when presented with a question, the model determines which option has the highest log-likelihood. We report performance using length-normalized accuracy (*acc_norm*), calculated by dividing the raw log-likelihood of each option by the number of characters. In the symbol setting, the question and all answer options are provided together in a multiple-choice format. The model selects the answer by choosing the symbol (e.g., A, B, C, and D) with the highest log-likelihood. Since all symbols have the same length, we evaluate performance using the standard accuracy (*acc*).

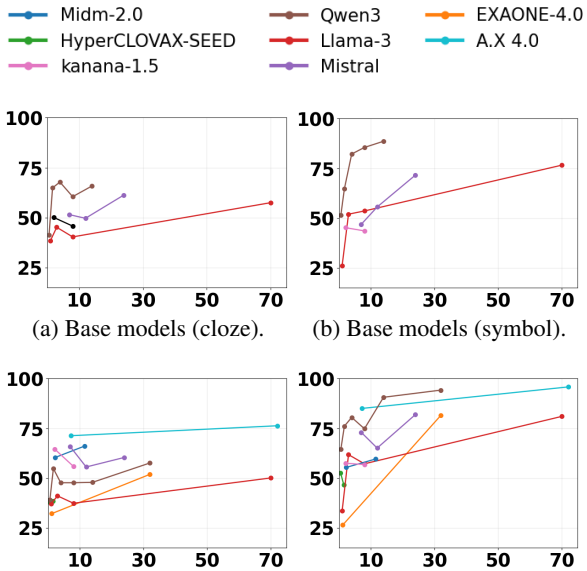
We assess models in both zero-shot and few-shot settings (1, 2, 5, and 10 shots). For the few-shot evaluations, we sample demonstration examples with three random seeds (1234, 308, 1028) and report the average performance across them.

Supervised fine-tuning. Using the training set from Thunder-KoNUBench, we perform supervised fine-tuning (SFT) on 35 models with fewer than 20 billion parameters. To prevent catastrophic forgetting (Kirkpatrick et al., 2017), where a model loses previously learned knowledge while acquiring new information, and to enable parameter-efficient training, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2022). Specific configurations can be found in Appendix G.

5.2 Results

We analyze the results by examining trends in model size and by comparing base models with instruction-tuned models. Additionally, we investigate the performance improvements achieved through SFT. Detailed evaluation results can be found in Appendix H.

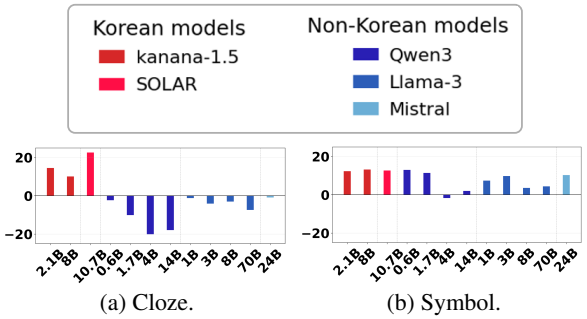
Model size. Our observations reveal a consistent trend across both Korean and non-Korean models: within each model family, larger models tend to achieve better performance. This pattern is evident in both evaluation formats—the cloze and symbol settings—indicating that increases in model size are generally associated with improved handling of negation (see Figure 3). This finding contrasts



(a) Base models (cloze). (b) Base models (symbol).

(c) Instruction-tuned models (cloze). (d) Instruction-tuned models (symbol).

Figure 3: Model performance across different model sizes on zero-shot setting. The horizontal axis represents model size (in billions of parameters), and the vertical axis indicates performance (*acc* or *acc_norm*).



(a) Cloze. (b) Symbol.

Figure 4: Performance change of instruction-tuned models relative to their base counterparts. The vertical axis represents the difference between the performance of instruction-tuned models and that of the corresponding base models.

with earlier research (Truong et al., 2023), which suggested that larger models are less sensitive to negation. In contrast, our results indicate that larger models are better equipped to move beyond superficial cues, making them more robust when addressing sentence-level negation.

It is important to note that this improvement is not strictly monotonic. We observe a noticeable slowdown or even a temporary decline in performance, especially among models with 8 to 12 billion parameters. This non-monotonic trend suggests that we are in a transitional phase of model scaling, where increasing capacity does not immediately lead to better handling of Korean negation. We hypothesize that this pattern indicates a mismatch between the emerging representational complexity and the level of linguistic supervision needed to master the nuanced aspects of negation.

Instruction-tuned models. In Figure 4, it is observed that instruction tuning improves general performance in the symbol setting. However, instruction tuning of non-Korean models often degrades performance in the cloze setting, suggesting that instruction tuning overemphasizes symbol-style MCQA formats. This bias improves format proficiency but reduces robustness to negation when evaluated on low-resource languages such as Korean.

These findings suggest that instruction tuning may exacerbate the *curse of multilinguality* (Conneau et al., 2020; Wang et al., 2020). Fine-tuning on high-resource languages can negatively affect performance on low-resource languages, and this effect is especially pronounced in understanding negation.

Error Analysis. Table 8 presents the distribution of incorrect choices selected by representative Korean and non-Korean models on Thunder-KoNUBench. The full error analysis results for all models are presented in Appendix I. In the cloze setting, more than 90% of the errors are concentrated on local negation options. This pattern consistently holds across model families, instruction-tuning settings, model sizes, and overall performance levels, as well as across both Korean and non-Korean models.

In the symbol-based setting, the concentration on local negation is somewhat reduced compared to the cloze setting, but it still remains the dominant error type. These results suggest that models do not fully capture the semantic effect of negation in Korean; rather, they tend to focus on the surface-level application of negation markers.

Effect of SFT. SFT is conducted in two settings: cloze and symbol. We find that both approaches yield performance comparable to or even better than the original 10-shot results, without compromising performance on other tasks such as ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), and Winogrande (Sakaguchi et al., 2021), as shown in Table 9. Additionally, our experiments reveal an apparent asymmetry in transferability between the two formats. Fine-tuning on the cloze for-

	model name	evaluation method	performance	incorrect choice distribution		
				local negation(%)	contradiction(%)	paraphrase(%)
Non-Korean	Qwen3-8B-Base	cloze	56.31	94.83	3.74	1.43
		symbol	85.44	85.56	8.02	6.42
	Llama-3.1-8B	cloze	39.56	93.17	4.51	2.32
		symbol	53.66	53.28	12.27	34.45
	Mistral-7B-Instruct-v0.3	cloze	63.16	93.66	4.65	1.69
		symbol	72.90	75.86	6.03	18.10
	gpt-4.1	symbol	92.13	97.03	1.98	0.99
	claude-sonnet-4-5-20250929	symbol	98.05	72.00	28.00	0.00
Korean	Midm-2.0-Base-Instruct (11.5B)	cloze	61.21	94.38	4.42	1.20
		symbol	59.50	50.77	30.77	18.46
	kanana-1.5-8b-instruct-2505	cloze	53.50	94.64	4.19	1.17
		symbol	56.70	51.44	13.67	34.89
	SOLAR-10.7B-v1.0	cloze	40.50	93.98	4.71	1.31
		symbol	41.98	44.83	14.23	40.94
	EXAONE-4.0-32B	cloze	50.16	96.41	3.12	0.47
		symbol	81.46	90.76	5.88	3.36
	A.X-4.0 (72B)	cloze	73.83	91.67	5.65	2.68
		symbol	95.72	87.27	7.27	5.45

Table 8: Incorrect choice distributions of Korean and non-Korean models under zero-shot evaluation.

	Thunder-KoNUBench		KMMLU		BoolQ		ARC		Hellaswag	Winogrande
	Cloze	Symbol	Negative	Affirmative	Original	Negated	Easy	Challenge		
Cloze-style fine-tuning	85.1 (+34.2)	69.6 (+10.5)	61.0 (+0.4)	62.3 (-0.1)	66.7 (+3.0)	53.4 (+0.9)	71.5 (+0.5)	49.2 (+0.2)	70.7 (+0.1)	67.4 (+0.1)
Symbol-style fine-tuning	57.3 (+6.4)	89.8 (+30.7)	61.2 (+0.7)	62.5 (+0.1)	65.5 (+1.8)	52.8 (+0.3)	71.4 (+0.4)	49.2 (+0.2)	70.7 (+0.1)	67.3 (+0.0)

Table 9: Average model performance after SFT on Thunder-KoNUBench and other tasks. Values in parentheses indicate the performance gain relative to the baseline.

mat leads to an average improvement of 10.5% in symbol-format evaluation, whereas fine-tuning on the symbol format only results in a 6.4% enhancement in cloze-format evaluation. This asymmetry suggests that training models to generate negated sentences (required in the cloze format) provides a richer and more generalizable supervision signal than merely selecting from pre-generated options (e.g., A, B, C, and D).

Cloze-style fine-tuning requires models to generate the correct standard negation of a sentence, which involves identifying the negation scope and applying appropriate Korean negation to the main predicate. In contrast, symbol-based formulations reduce the negation task to a label selection problem, providing a limited training signal for learning how negation is constructed.

Fine-tuning on Thunder-KoNUBench also improves broader contextual understanding, as evidenced by substantial gains on KoBest BoolQ, which we previously examined in Section 3.3. Consistent with our negation results, cloze-style fine-tuning yields larger improvements than symbol-based fine-tuning, highlighting the importance of generation-based supervision for learning negation.

6 Conclusion

In this paper, we demonstrate that LLMs struggle with negation in Korean and introduce Thunder-KoNUBench, a benchmark for evaluating sentence-level negation understanding. Inspired by Thunder-NUBench, Thunder-KoNUBench covers a broad range of Korean-specific negation types and structures and closely matches their empirical distribution. Evaluating 47 LLMs, we find that larger models are generally more robust, while multilingual instruction tuning can degrade negation performance in low-resource languages like Korean. Error analysis indicates that most errors concentrate on local negation, suggesting reliance on surface-level negation cues rather than true semantic understanding. Fine-tuning experiments further show that a cloze-based format provides more effective supervision than symbol-based alternatives for improving negation understanding. Thunder-KoNUBench is publicly available at https://champ.snu.ac.kr/.

Limitations

Lexical negation is inherently relational, as forms such as "있다" (exist) and "없다" (not exist) constitute negation only with respect to each other, rather than in isolation. As a result, in our corpus-level statistical analysis, we primarily focused on syntactic negation markers and did not identify or count lexical negation. However, Thunder-KoNUBench itself includes a number of instances involving lexical negation, as complementary antonyms are permitted when constructing standard negation. Both syntactic and lexical negation are therefore represented in Thunder-KoNUBench.

Furthermore, Thunder-KoNUBench does not simply require models to interpret naturally occurring negated sentences; instead, it asks them to identify the option that correctly corresponds to the negation of a given original sentence. This design is motivated by our goal of capturing a fundamental aspect of negation—the operation that maps a proposition P to its counterpart $\neg P$—in Thunder-KoNUBench. As a result, our benchmark does not evaluate negation understanding in more naturalistic settings, where negation appears in natural contexts. In particular, it does not assess how models comprehend and reason over negation in context. Developing more naturalistic evaluation settings that examine how models understand negation in context remains an important direction for future work.

Ethical Considerations

This work does not rely on crowdsourcing. Instead, all data included in Thunder-KoNUBench were manually reviewed by the authors to ensure high quality, relevance, and compliance with ethical standards. All datasets and tools used for training and evaluation are publicly available and were utilized in accordance with their respective licenses.

When using OpenAI's text generation models, we exercise additional caution to prevent the inclusion of harmful, biased, or privacy-violating content. All generated examples undergo manual inspection to ensure they meet ethical and safety requirements. In particular, we verify that the final dataset contains no personally identifiable information or offensive material.

The human evaluation was approved by the Institutional Review Board (IRB No. 2512/004-012). All participants received sufficient information about the study and were given adequate rest time, ensuring that the evaluation was conducted in an ethically responsible manner.

The Thunder-KoNUBench dataset is made available under the CC BY-NC-SA 4.0 license to support transparent and reproducible research and to facilitate responsible reuse. We believe that our work provides a meaningful step toward building more trustworthy and interpretable language models.

Acknowlegments

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*.

AI-Hub, S.Korea. 한국어 성능이 개선된 초거대ai 언어모델 개발 및 데이터 (dataset and large language models with improved korean performance). Access date: 2025-10.

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.

ANTHROPIC. 2025. System card: Claude opus 4.5.

Yunju Bak, Hojin Lee, Minho Ryu, Jiyeon Ham, Seungjae Jung, Daniel Wontae Nam, Taegyeong Eo, Donghun Lee, Doohae Jung, Boseop Kim, and 1 others. 2025. Kanana: Compute-efficient bilingual language models. *arXiv e-prints*, pages arXiv–2502.

Jeremy Barnes, Erik Velldal, and Lilja Øvrelid. 2021. Improving sentiment analysis with multi-task learning of negation. *Natural Language Engineering*, 27(2):249–269.

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*.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

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.

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. *arXiv preprint arXiv:2310.15941*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Andreas Grivas, Beatrice Alex, Claire Grover, Richard Tobin, and William Whiteley. 2020. Not a cute stroke: analysis of rule-and neural network-based information extraction systems for brain radiology reports. In *The 11th International Workshop on Health Text Mining and Information Analysis at EMNLP 2020*, pages 24–37. Association for Computational Linguistics.

Reto Gubelmann and Siegfried Handschuh. 2022. Context matters: A pragmatic study of plms' negation understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4602–4621.

Haochen Han, Alex Jinpeng Wang, Fangming Liu, and Jun Zhu. 2025. Negation-aware test-time adaptation for vision-language models. *arXiv preprint arXiv:2507.19064*.

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.

Md Mosharaf Hossain, Antonios Anastasopoulos, Eduardo Blanco, and Alexis Palmer. 2020a. It's not a non-issue: Negation as a source of error in machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3869–3885.

Md Mosharaf Hossain, Dhivya Chinnappa, and Eduardo Blanco. 2022a. An analysis of negation in natural language understanding corpora. *arXiv preprint arXiv:2203.08929*.

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. 2020b. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. *arXiv preprint arXiv:2105.03519*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Edward J Hu, 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. In *International Conference on Learning Representations*.

Myeongjun Jang, Dohyung Kim, Deuk Sin Kwon, and Eric Davis. 2022. KoBEST: Korean balanced evaluation of significant tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3697–3708, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of lstms to understand negative polarity items. *arXiv preprint arXiv:1808.10627*.

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, Online. Association for Computational Linguistics.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521–3526.

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.

ko.wikipedia. 2025. Korean Wikipedia(위키백과). Access date: 2025-06.

Tech. Innovation Group KT. 2025. Mi:dm 2.0: Korea-centric bilingual language models.

LG-Research, Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Kyubeen Han, Seokhee Hong, Junwon Hwang, Taewan Hwang, and 1 others. 2025a. Exaone 4.0: Unified large language models integrating non-reasoning and reasoning modes. *arXiv preprint arXiv:2507.11407*.

LG-Research, Kyunghoon Bae, Eunbi Choi, Kibong Choi, Stanley Jungkyu Choi, Yemuk Choi, Seokhee Hong, Junwon Hwang, Hyojin Jeon, Kijeong Jeon, and 1 others. 2025b. Exaone deep: Reasoning enhanced language models. *arXiv preprint arXiv:2503.12524*.

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

OpenAI. 2025. Text generation and prompting.

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*.

Abhilasha Ravichander, Matt Gardner, and Ana Marasović. 2022. Condaqa: A contrastive reading comprehension dataset for reasoning about negation. *arXiv preprint arXiv:2211.00295*.

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):99–106.

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.

Telecom SKT. 2025. A.X 4.0: Foundation model specialized in korean, optimized for enterprise applications.

Yeonkyoung So, Gyuseong Lee, Sungmok Jung, Joonhak Lee, JiA Kang, Sangho Kim, and Jaejin Lee. 2026. Thunder-nubench: A benchmark for llms' sentence-level negation understanding. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 4749–4793.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. Kmmlu: Measuring massive multitask language understanding in korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104.

Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, Charles Lovering, Jason Phang, Anish Thite, Thomas Wang, and 1 others. 2025. Eleutherai/lm-evaluation-harness: v0. 4.9. *Zenodo*.

Gongbo Tang, Philipp Rönchen, Rico Sennrich, and Joakim Nivre. 2021. Revisiting negation in neural machine translation. *Transactions of the Association for Computational Linguistics*, 9:740–755.

Team-HyperCLOVA. 2025. HyperCLOVA X THINK Technical Report. *Preprint*, arXiv:2506.22403.

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.

Tereza Vrabcová, Marek Kadlčík, Petr Sojka, Michal Štefánik, and Michal Spiegel. 2025. Negation: A pink elephant in the large language models' room? *arXiv preprint arXiv:2503.22395*.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

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*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

A Negative Expressions in Korean

A.1 Syntactic Negation

There are three main types of syntactic negation in Korean, each characterized by distinct meanings and syntactic constraints(Table 10 shows details and examples). In constructing Thunder-KoNUBench, we carefully adhered to these constraints and incorporated all three types.

안 계열 (*An* type). The *An* type is used to either simply reverse the truth value or to negate the subject's volition. It cannot be used in sentences whose predicates involve actions that presuppose the subject's ability (e.g., "*알다*"(know), "*견디다*"(endure)). The *An* type is divided into short-form negation, where a negative adverb "*안*" precedes the predicate, and long-form negation, where a negative auxiliary verb "*-지 않-*" follows the predicate.

못 계열 (*Mot* Type). *Mot* type expresses the subject's inability to perform an action (corresponding to *cannot* in English). It cannot be used with predicates that are adjectival inflections(e.g., "*착하다*"(kind), "*아름답다*"(beautiful)), as the notion of inability cannot be semantically ascribed to stative predicates. Like the *An* type, it has both a short form, where "못" precedes the predicate, and a long form, where the negative auxiliary verb "*-지 못하-*" follows the predicate.

말다 (*Malda*). In imperative or hortative sentence, prohibitive negation is realized by attaching the auxiliary verb "*-지 말-*" after the predicate.

Constraints of short-form negation. Short-form negation realized by negative adverbs cannot generally be applied to predicates that are compounds or derived forms. For example, expressions such as "안 아름답다" and "못 공부했다" are considered awkward, whereas their long-form counterparts "아름답지 않다" and "공부하지 못했다" are natural. However, this restriction is not absolute. When compound predicates are formed through auxiliary connective endings such as -아/어-, short-form negation is permitted (e.g., "안 들어가다"). In addition, short-form negation is also allowed when the predicate is a derived form created by passive suffixes (-이-, -히-, -리-, -기-) or causative suffixes (-이-, -히-, -리-, -기-, -우-, -구-, -추-).

A.2 Lexical Negation

Lexical negation in Korean is realized either by attaching a negative prefix or by using a complementary antonym. Negative prefixes (e.g., "비-", "불-", "미-", "무-", "몰-") are attached to nominals. For example, the noun "공평" (fairness) combines with the negative prefix "불-" to form "불공평" (unfairness). In addition, certain derivational suffixes (e.g., "-하다") attach to nouns, ideophones, or mimetic words to form predicates. For instance, "공평" (fairness) becomes "공평하다" (is fair), which functions as a predicate in a sentence. The same process also applies to "불공평" (unfairness), yielding "불공평하다" (is unfair), which likewise serves as a predicate.

Since Thunder-KoNUBench focuses on negation at the predicate level, we allow prefix-based negation only when a predicate is formed as a "nominal + 하다" (e.g., "공평하다"(is fair) → "불공평하다"(is unfair)) construction.

Antonyms can be broadly categorized into three types:

- complementary antonyms: mutually exclusive pairs with no intermediate states (e.g., 살다 (alive) / 죽다(dead))

Negation Type		Affirmative Sentence	Negative Sentence
An Type (안 계열)	**Short-Form**	나는 저녁으로 빵을 먹었다. (I ate bread for dinner.)	나는 저녁으로 빵을 안 먹었다. (I didn't eat bread for dinner.)
	Long-Form	나는 저녁으로 빵을 먹었다. (I ate bread for dinner.)	나는 저녁으로 빵을 먹지 않았다. (I didn't eat bread for dinner.)
Mot Type (못 계열)	**Short-Form**	나는 저녁으로 빵을 먹었다. (I ate bread for dinner.)	나는 저녁으로 빵을 못 먹었다. (I could not eat bread for dinner.)
	Long-Form	나는 저녁으로 빵을 먹었다. (I ate bread for dinner.)	나는 저녁으로 빵을 먹지 못했다. (I could not eat bread for dinner.)
Malda(말다)		저녁을 먹어라. (Eat dinner.)	저녁을 먹지 말아라. (Do not eat dinner.)

Table 10: Syntactic negation type and example in Korean

- gradable antonyms: pairs of words that denote opposite ends of a continuous scale, allowing for intermediate degrees between the two extremes (e.g., 덥다(hot) / 춥다(cold)).

- relational antonyms: pairs of words that describe a reciprocal relationship where one implies the existence of the other (e.g., 사다(buy) / 팔다(sell)).

However, from the perspective of negation that completely reverses the meaning of the original sentence, only complementary antonyms constitute true negation. Accordingly, Thunder-KoNUBench permits only complementary antonyms when applying lexical negation to predicates.

B Sentence Structure in Korean

In both corpus analysis and the construction of Thunder-KoNUBench, we carefully adhere to the characteristics of Korean sentence structure. In this section, we describe the sentence structure of Korean.

Korean sentence structure can be broadly divided into 홑문장(simple sentences) and 겹문장(complex sentences). A simple sentence contains a single main(i.e., independent) clause, whereas a complex sentence consists of two or more clauses, featuring multiple subject–predicate structures. Complex sentences are further classified into 이어진 문장(connected sentences) and 안긴 문장(sentences with embedded clause), depending on how the clauses are combined. Table 11 presents the detailed descriptions and examples.

B.1 Connected Sentences

A connected sentence is formed by combining two or more clauses through connective endings(e.g., -고, -거나, -(어/아)서). Coordinated sentences can be further divided into:

- 대등하게 이어진 문장(Coordinated sentences), where two or more main clauses are linked in an equal relationship,

- 종속적으로 이어진 문장(sentences with subordinate clauses), where a subordinate clause is combined with a main clause in a dependent relationship.

B.2 Sentences with Embedded Clause

In Korean, a clause can be embedded within another matrix clause, functioning as a single grammatical constituent. In this paper, we treat the matrix clause as the main clause and the embedded clause as the dependent clause. This distinction is motivated by the design principle of Thunder-KoNUBench: clauses are categorized as main clauses if they can function as a complete sentence on their own, and as dependent clauses otherwise.

Embedded sentences are classified according to the grammatical role of the embedded clause, including:

- 명사절을 안은 문장(sentences with embedded noun clauses)

- 관형절을 안은 문장(sentences with embedded adnominal clauses)

Structure		Description	Example
Simple Sentence		A sentence that contains a single main(i.e., independent) clause.	나는 집에 갔다. (I went home.)
Connected Sentences	**Coordinated sentences**	Two or more main clauses are linked in an equal relationship.	예술은 길고 인생은 짧다. (Art is long, but life is short.)
	With subordinate clause	A subordinate clause is combined with a main clause in a dependent relationship.	비가 와서 땅이 젖었다. (Because it rained, the ground became wet.)
Sentences with Embedded clause	**Noun clause**	An embedded clause that functions as a noun (e.g., formed with endings such as -(으)ㅁ or -기).	비가 오기를 기다린다. (I am waiting for it to rain.)
	Adnominal clause	An embedded clause that modifies a nominal argument (e.g., formed with -(으)ㄴ, -는, -(으)ㄹ, -던).	내가 읽은 책은 재미있다. (The book that I read is interesting.)
	Quotation clause	An embedded clause that reports/quotes utterance or thought (e.g., formed with -라고, -고).	그가 그녀가 온다고 말했다. (He said that she would come.)
	Adverbial clause	An embedded clause that functions as an adverbial modifier (e.g., formed with -게, -도록), providing additional information about the main clause.	그는 그 날 날씨가 나빠서 집에 있었다. (He stayed home because the weather is bad.)
	Predicative clause	An embedded clause that serves as the predicate of the sentence.	코끼리는 코가 길다. (An elephant has a long trunk.)

Table 11: Korean sentence structure.

- 인용절을 안은 문장(sentences with embedded quotation clauses)
- 부사절을 안은 문장(sentences with embedded adverbial clauses)
- 서술절을 안은 문장(sentences with embedded predicative clauses).

C Details of the corpus

This section describes the corpus used for analyzing the statistical distribution of negation in Korean, as introduced in Section 3.2. We use the dataset titled "한국어 성능이 개선된 초거대 AI 언어모델 개발 및 데이터" (Dataset and Large Language Models with Improved Korean Performance). It consists of approximately 2.28 billion whitespace-delimited Korean word units(어절; Eojel) and is publicly released by The Open AI Dataset Project(AI-Hub, S.Korea).

The corpus contains both spoken and written-style texts and spans a wide range of topics and domains. Table 12 presents the distribution of sentence types in the corpus, and Table 13 shows the topic-wise distribution of sentences. The diversity of the corpus ensures that the observed statistical patterns of negation reflect real-world usage of Korean negation phenomena.

Category	Number of Sentences	Number of whitespace-delimited Korean word units (어절; Eojeol)
Spoken-style	683,277	1,025,519,624
Written-style	2,678,129	1,260,946,221
Total	3,361,406	2,286,465,845

Table 12: Distribution of sentence types in the corpus.

Category	Number of Sentences	Proportion (%)
Engineering	61,963	1.84
Miscellaneous	374,513	11.14
Daily-life Expressions	3,217	0.10
Healthcare	186,247	5.54
Society	1,218,049	36.24
Industry	600,154	17.85
Arts & Entertainment	178,491	5.31
Humanities	489,046	14.55
Natural Science	149,095	4.44
Religion	100,629	2.99
Total	3,361,404	100.00

Table 13: Distribution of sentences across topic categories in the corpus.

D Details of Constructing Thunder-KoNUBench

This section describes the data construction procedure for Thunder-KoNUBench, including sentence-pair extraction, sentence merging, and the genera-

tion of contradiction and paraphrase options. The prompts used in this process were originally written in Korean, and English translations are provided in the corresponding figures.

Splitting original text into sentence pairs. We first crawled the entire Korean Wikipedia and segmented the raw text into individual sentences using the `split_sentences` function from the `KSS` package, a rule-based Korean sentence splitter. From the crawled corpus, we extracted approximately 4 million consecutive sentence pairs. Since our target benchmark size was about 5,000 instances, we did not use all extracted pairs. Instead, we randomly sampled 5,000 sentence pairs from this pool.

Merging sentence pairs into a single sentence. Each two-sentence unit was converted into a single, natural, and well-formed sentence using the GPT-4.1 mini model with OpenAI API. This step was designed to avoid overly short or trivial source sentences and to obtain contexts rich enough for constructing benchmark instances. During generation, we instructed the model to preserve the original meaning while producing a grammatically coherent sentence and avoiding unnecessary discourse markers or unnatural referential expressions. The Korean and English versions of the prompt used for this step are shown in Figure 5. The generated outputs were later manually checked and corrected when needed.

Generating contradiction and paraphrase options. We generated contradiction and paraphrase candidates from the merged original sentences using the GPT-5 model with OpenAI API. For contradiction, we instructed the model to produce a sentence incompatible with the original one without using explicit negation markers, while preserving the overall structure as much as possible. The corresponding Korean and English prompts are shown in Figure 6. For paraphrase, we instructed the model to preserve the original meaning while changing the surface form without adding new information. The corresponding prompts are shown in Figure 7. All generated candidates were manually reviewed and refined by the authors.

E Details of the Review Process

Thunder-KoNUBench is constructed through a thorough review process among the authors. In this section, we provide the protocol in reviewing and how we deal with inter-annotators' disagreements.

E.1 Human Review Protocol

Thunder-KoNUBench is constructed through a thorough review process conducted by the authors. In this section, we describe the review protocol and how inter-annotator disagreements are identified and resolved.

Independent task allocation and review. Each author is assigned a distinct subset of the dataset. For each instance, the author first finalizes the original sentence and then generates its corresponding standard negation and local negation. In addition, contradiction and paraphrase options initially generated by OpenAI API(OpenAI, 2025) are manually reviewed and revised to ensure linguistic correctness and semantic validity.

After an author completes their assigned subset, all instances undergo independent cross-checking by another author. The reviewer verifies whether each option is appropriate with respect to the original sentence and records all cases of disagreement.

Consensus-building and revision. All disputed instances are discussed in weekly regular meetings attended by all authors. During these meetings, we jointly assess the validity of each disagreement and determine appropriate revisions when necessary. Through this consensus-based revision process, all issues are resolved and the final versions of the data are confirmed.

E.2 Inter-annotator Agreement

	1st regular meeting	2nd regular meeting
disagreement rate	12.8%	3.2%

Table 14: Inter-annotation disagreement rates across review stages.

We conduct two rounds of cross-checking for all items, each performed by an independent reviewer. Table 14 reports the proportion of items for which at least one label (original sentence, standard negation, local negation, contradiction, or paraphrase) was marked as a disagreement during each review stage. All disagreement cases are resolved through consensus-building discussions in regular meetings, yielding a fully consistent final dataset.

```
def merge(text):
    prompt = f"""
        -   대등적 연결어미('-고', '-지만', '-(으)나', '-(으)며 등)는 앞 절과
            뒤 절이 독립적일 때만 사용할 것.
        -   대등적 연결어미가 쓰였을 때 지시대명사('이', '그', '저', '이것', '
            저것', '그것' 등), 지시관형사('이', '그', '저' 등)앞 절의 내용을
            지칭하는 말이 들어가면 안 됨
        -   앞 절과 뒤 절의 내용이 종속적일 경우 종속적 연결어미('-아/어', '-
            아서/어서', '-기 때문에' 등)를 사용할 것
        -   보조사 '도', '만' 등은 사용하지 말 것
        -   위의 지침들을 지키기 위해 원문의 세부 정보를 삭제하여 문장을 만들
            어도 됨
        원문을 줄 테니, 다른 말은 하지 말고 한 문장만 생성해.
        원문: {text}
        생성된 문장:
        """

    completion = client.chat.completions.create(
        model="gpt-4.1-mini",
        messages=[
            {"role": "system", "content": "너는 자연스럽고 논리적인 한 문장
            을 생성하는 전문가야. 아래 지침을 따라 여러 문장으로 이루어진 원
            문의 내용을 종합해서, 매끄럽고 자연스러운 한 문장을 생성해줘."},
            {"role": "user", "content": prompt}
        ]
    )

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

```
def merge_english_version(text):
    prompt = f"""
        - Use coordinate connective endings (e.g., '-go', '-jiman', '-(eu)na',
          '-(eu)myeo') only when the preceding and following clauses are
          independent.
        - When a coordinate connective ending is used, do not include
          demonstrative pronouns (e.g., 'this', 'that', 'these') or
          demonstrative determiners that refer back to the content of the
          preceding clause.
        - If the relationship between the clauses is subordinate, use
          subordinate connective endings (e.g., '-a/eo', '-aseo/eoseo', '-gi
          ttaemune').
        - Do not use auxiliary particles such as 'do' or 'man'.
        - To satisfy the guidelines above, some details from the original text
          may be omitted.
        Given the original text, generate only one sentence and nothing else.
        Original text: {text}
        Generated sentence:
        """

    completion = client.chat.completions.create(
        model="gpt-4.1-mini",
        messages=[
            {"role": "system", "content": "You are an expert in generating a
            natural and logically coherent single sentence. Following the
            guidelines below, synthesize the content of the original multi-
            sentence text into one smooth and natural sentence."},
            {"role": "user", "content": prompt}
        ]
    )

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

Figure 5: Korean prompt used to merge a pair of sentences into a single natural and well-formed sentence, with an English translation.

```
def contradiction_generate(text):
    prompt = f"""
        넌 부정 표현을 사용하지 않고, 원문과 양립할 수 없는 문장을 만드는 전문
        가야. 양립 불가능한 문장이란, 원문이 참일 때 성립이 불가능한 문장을 말
        해. 원문을 줄 테니, 부정 표현을 사용하지 않고 원문과 양립할 수 없는
        contradiction 문장을 생성해야 해.
        단, contradiction 문장은 원문을 과도하게 축약하거나 핵심 요소(인물, 사
        건, 날짜, 장소 등)를 생략하지 말고, 원문의 전체 구조와 길이를 최대한 유
        지하면서 일부 표현만 바꾸어 만들어야 한다.
        〈부정 표현〉
        안/-지 않다/못/-지 못하다
        〈부정 표현 예시〉
        원문: 민주주의 지지자들이 비상사태 법이 공정한 재판과 투표 권리를 침해
        하는 행위라고 비판했으며, 인권 단체가 1990년대부터 2010년까지 수천 명이
        기소나 재판 없이 장기 수감되었다고 밝혔다.
        부정: 민주주의 지지자들이 비상사태 법이 공정한 재판과 투표 권리를 침해
        하는 행위라고 비판하지 않았거나, 인권 단체가 1990년대부터 2010년까지 수
        천 명이 기소나 재판 없이 장기 수감되었다고 밝히지 않았다.
        〈contradiction 예시〉
        원문: 1940년 8월 2일 비시 정부가 드골에게 사형을 선고했으며, 영국 정부
        와 드골이 관계가 원만하지 못해 충돌하였다.
        contradiction: 1940년 8월 2일 비시 정부가 드골에게 무죄를 선고했으며,
        영국 정부와 드골의 관계가 매우 우호적이었다.
        이제 원문을 줄 테니, contradiction에 해당하는 문장만 생성해. 다른 말은
        하지 마.
        원문: {text}
        contradiction:
        """

    response = client.responses.create(
        model="gpt-5", input=prompt, reasoning={"effort":"low"},
        text={"verbosity":"low"}
    )

    return response.output_text
```

```
def contradiction_generate_english_version(text):
    prompt = f"""
        You are an expert in generating a sentence that is incompatible with
        the original sentence without using explicit negation expressions. An
        incompatible sentence is one that cannot be true if the original
        sentence is true. Given the original sentence, generate a contradiction
        sentence that is incompatible with it without using explicit negation.
        However, do not excessively shorten the contradiction sentence or omit
        core elements such as people, events, dates, or locations. Preserve the
        overall structure and length of the original sentence as much as
        possible, changing only some expressions.
        〈Explicit negation expressions〉
        an / -ji anhda / mot / -ji mothada
        〈Example of explicit negation〉
        Original: Supporters of democracy criticized the emergency law as
        violating the right to a fair trial and voting rights, and a human
        rights group stated that thousands of people had been detained for long
        periods without indictment or trial from the 1990s to 2010.
        Negation: Supporters of democracy did not criticize the emergency law
        as violating the right to a fair trial and voting rights, or a human
        rights group did not state that thousands of people had been detained
        for long periods without indictment or trial from the 1990s to 2010.
        〈Example of contradiction〉
        Original: On August 2, 1940, the Vichy government sentenced de Gaulle
        to death, and the British government and de Gaulle clashed because
        their relationship was not smooth.
        Contradiction: On August 2, 1940, the Vichy government declared de
        Gaulle not guilty, and the relationship between the British government
        and de Gaulle was very friendly.
        Now, given the original sentence, generate only a contradiction
        sentence and nothing else.
        Original: {text}
        Contradiction:
        """

    response = client.responses.create(
        model="gpt-5", input=prompt, reasoning={"effort":"low"},
        text={"verbosity":"low"}
    )

    return response.output_text
```

Figure 6: Korean prompt used to generate a contradiction sentence that is incompatible with the original sentence without explicit negation, with an English translation.

```python
def paraphrase_generate(text):
    prompt = f"""
        너는 원문과 일치하지 않으면서도 원문의 의미를 유지하는 문장을 생성하는
        paraphrase 전문가야. 원문을 줄 테니, 원문의 의미를 유지하면서 문장 구조
        나 단어가 수정된 문장을 생성해줘. 원문과 동일한 문장을 생성해서는 안 돼.
        수정된 문장은 원문이 참이라면 반드시 참이어야 해. 즉, 원문에서 세부 요
        소가 삭제되는 것은 괜찮지만 세부 요소가 추가되어서는 안 돼.
        〈paraphrase 예시〉
        원문: 우루과이는 2023년 코파 아메리카 8강전에서 페루와의 경기에서 승부
        차기로 패해 대회에서 탈락했다.
        paraphrase: 2023년 코파 아메리카의 8강전에서 우루과이가 페루에 패했다.
        원문: 이여로는 경기도 이천시와 여주시를 연결하는 14.9km 길이의 도로이다.
        paraphrase: 이여로는 경기도에 있는 도로이다.
        원문: 마이클 신은 1969년 2월 5일에 태어난 웨일스 출신의 영국 배우이다.
        paraphrase: 마이클 신의 출생일과 출생지는 각각 1969년 2월 5일과 웨일스
        이고, 영국 배우이다.
        이제 원문을 줄 테니, paraphrase 문장을 생성해. 다른 말은 하지 마.
        원문: {text}
        paraphrase:
        """

    response = client.responses.create(
        model="gpt-5", input=prompt, reasoning={"effort":"low"},
        text={"verbosity":"low"}
    )

    return response.output_text
```

```python
def paraphrase_generate_english_version(text):
    prompt = f"""
        You are an expert in generating paraphrases that preserve the meaning
        of the original sentence while differing from it in form. Given the
        original sentence, generate a sentence whose structure or wording is
        modified while preserving the original meaning. Do not generate a
        sentence identical to the original.
        The generated sentence must be true whenever the original sentence is
        true. In other words, it is acceptable to omit some details from the
        original sentence, but it must not introduce any new information.
        〈Examples of paraphrase〉
        Original: Uruguay was eliminated from the 2023 Copa America after
        losing to Peru in a penalty shootout in the quarterfinals.
        Paraphrase: Uruguay lost to Peru in the quarterfinals of the 2023 Copa
        America.
        Original: Iyeo-ro is a 14.9 km road connecting Icheon and Yeoju in
        Gyeonggi Province.
        Paraphrase: Iyeo-ro is a road in Gyeonggi Province.
        Original: Michael Sheen is a British actor from Wales, born on February
        5, 1969.
        Paraphrase: Michael Sheen is a British actor whose birthplace is Wales
        and whose birth date is February 5, 1969.
        Now, given the original sentence, generate only a paraphrase sentence
        and nothing else.
        Original: {text}
        Paraphrase:
        """

    response = client.responses.create(
        model="gpt-5", input=prompt, reasoning={"effort":"low"},
        text={"verbosity":"low"}
    )

    return response.output_text
```

Figure 7: Korean prompt used to generate a paraphrase that preserves the original meaning while changing the surface form, with an English translation.

F Details of the Models

This section details the models evaluated in this paper and their implementation methods. For all evaluation, we use a 16-bit (bf16) quantized model.

Korean Models

- Midm-2.0
  - Midm-2.0-Mini-Instruct (2.3B)
  - Midm-2.0-Base-Instruct (11.5B)
- EXAONE
  - EXAONE-4.0-1.2B
  - EXAONE-4.0-32B
  - EXAONE-Deep-2.4B
  - EXAONE-Deep-7.8B
  - EXAONE-Deep-32B
- kanana-1.5
  - kanana-1.5-2.1b-base
  - kanana-1.5-2.1b-instruct-2505
  - kanana-1.5-8b-base
  - kanana-1.5-8b-instruct-2505
- HyperCLOVAX-SEED
  - HyperCLOVAX-SEED-Text-Instruct-0.5B
  - HyperCLOVAX-SEED-Text-Instruct-1.5B
  - HyperCLOVAX-SEED-Think-14B
- A.X-4.0
  - A.X-4.0-Light (7.2B)
  - A.X-4.0 (72B)
- SOLAR
  - SOLAR-10.7B-v1.0
  - SOLAR-10.7B-Instruct-v1.0

Non-Korean Models

- Qwen3
  - 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 3.1 and Llama 3.2
  - Llama-3.1-8B
  - Llama-3.1-8B-instruct
  - Llama-3.1-70B
  - Llama-3.1-70B-instruct
  - Llama-3.2-1B
  - Llama-3.2-1B-instruct
  - Llama-3.2-3B
  - Llama-3.2-3B-instruct
- Mistral
  - Mistral-7B-v0.3
  - Mistral-7B-Instruct-v0.3
  - Mistral-Nemo-Base-2407 (12B)
  - Mistral-Nemo-Instruct-2407 (12B)
  - Mistral-Small-24B-Base-2501
  - Mistral-Small-24B-Instruct-2501

G Configurations for Fine-Tuning

We apply LoRA(Hu et al., 2021) with a rank of 8, targeting all linear layers. The scaling factor alpha is set to 32, and the dropout rate to 0.05. We use a batch size of 128 with gradient accumulation,

cosine learning-rate scheduling, and bfloat16 precision. All training is conducted on 8 AMD MI250 GPUs, each with 64 GB of memory.

H Evaluation Results

This section presents our comprehensive evaluation results. We report accuracy(*acc*) for KMMLU(Son et al., 2025) under the symbol-based evaluation, *acc* for BoolQ(Jang et al., 2022) and Winogrande(Sakaguchi et al., 2021) under the cloze-based evaluation, and length-normalized accuracy (*acc_norm*) for ARC(Clark et al., 2018) and HellaSwag(Zellers et al., 2019) under the cloze-based evaluation, following the default settings of the LM Evaluation Harness(Sutawika et al., 2025). For API-based models such as GPT and Claude, we report only symbol-based performance measured by `exact_match`, since these models provide only final textual outputs and do not expose token-level likelihoods. All evaluations is conducted on 8 AMD MI250 GPUs, each with 64 GB of memory.

Table 15 presents the evaluation results on KMMLU using the methodology introduced in Section 3.3, comparing questions containing negation with their affirmative counterparts. In addition, it reports performance on KoBest BoolQ and the corresponding results obtained when the questions are transformed into their negated forms. Table 16 reports the zero-shot and few-shot performance on Thunder-KoNUBench. Tables 17 and 18 present model performance after fine-tuning with a focus on the cloze format. Tables 19 and 20 report model performance across diverse tasks after fine-tuning targeting the symbol format.

I Error Analysis

This section presents detailed error analysis results. Table 21 reports the results for the Qwen3 (Yang et al., 2025) and Llama (Grattafiori et al., 2024) model families. Table 22 presents the results for the Mistral (Jiang et al., 2023), GPT-4.1 (Achiam et al., 2023), and Claude-Opus 4.5 (ANTHROPIC, 2025) families. Table 23 provides the error analysis results for all Korean models.

Model	KMMLU		BoolQ	
	Negative	Affirmative	Original	Negated
Midm-2.0-Mini-Instruct (2.3B)	63.1	**63.5**	**87.3**	68.2
Midm-2.0-Base-Instruct (11.5B)	61.7	**73.8**	**92.1**	77.3
EXAONE-4.0-1.2B	**51.0**	50.1	**51.4**	48.4
EXAONE-4.0-32B	**75.4**	74.2	**81.1**	52.8
EXAONE-Deep-2.4B	47.8	**53.6**	**51.0**	48.2
EXAONE-Deep-7.8B	48.6	**56.3**	**69.6**	48.9
EXAONE-Deep-32B	56.6	**65.9**	**85.4**	68.2
kanana-1.5-2.1b-base	57.4	**64.0**	**54.8**	48.5
kanana-1.5-2.1b-instruct-2505	64.5	**66.8**	**66.4**	51.5
kanana-1.5-8b-base	66.3	**66.5**	**63.7**	51.2
kanana-1.5-8b-instruct-2505	66.5	**69.9**	**80.3**	56.9
HyperCLOVAX-SEED-Text-Instruct-0.5B	59.9	**60.8**	**53.5**	48.1
HyperCLOVAX-SEED-Text-Instruct-1.5B	61.9	**63.5**	**69.3**	51.0
HyperCLOVAX-SEED-Think-14B	**75.7**	74.8	**87.0**	66.6
A.X-4.0-Light (7.2B)	76.2	**76.8**	**93.0**	74.9
A.X-4.0 (72B)	**87.7**	86.5	**94.7**	77.6
SOLAR-10.7B-v1.0	54.0	**58.2**	**50.6**	48.1
SOLAR-10.7B-Instruct-v1.0	60.4	**62.7**	**86.0**	67.7
Qwen3-0.6B-Base	**56.9**	54.2	**52.1**	49.2
Qwen3-0.6B	50.0	**52.2**	**50.4**	48.1
Qwen3-1.7B-Base	54.9	**62.0**	**61.7**	48.7
Qwen3-1.7B	57.2	**60.8**	**51.9**	48.6
Qwen3-4B-Base	**70.2**	69.1	**63.7**	48.9
Qwen3-4B	59.5	**65.9**	**53.6**	48.1
Qwen3-8B-Base	**73.6**	72.7	**57.2**	48.0
Qwen3-8B	68.8	**70.8**	**61.2**	48.9
Qwen3-14B-Base	**76.3**	69.0	**67.7**	50.8
Qwen3-14B	**75.5**	70.2	**64.0**	54.1
Qwen3-32B	**79.5**	77.7	**58.8**	47.6
Llama-3.2-1B	**51.2**	49.7	49.4	**49.8**
Llama-3.2-1B-Instruct	50.6	**52.4**	**50.4**	48.1
Llama-3.2-3B	**56.7**	54.2	**54.2**	47.4
Llama-3.2-3B-Instruct	**59.5**	57.7	**51.5**	47.9
Llama-3.1-8B	60.3	**60.9**	**53.0**	48.2
Llama-3.1-8B-Instruct	56.3	**57.7**	**54.1**	48.1
Llama-3.1-70B	**70.8**	70.4	**67.7**	49.6
Llama-3.1-70B-Instruct	**75.0**	74.7	**89.7**	63.5
Mistral-7B-v0.3	52.4	**56.3**	**58.0**	48.1
Mistral-7B-Instruct-v0.3	58.3	**59.1**	**77.1**	56.0
Mistral-Nemo-Base-2407 (12B)	54.6	**65.3**	**66.5**	46.9
Mistral-Nemo-Instruct-2407 (12B)	62.2	**63.1**	**76.1**	47.6
Mistral-Small-24B-Base-2501	66.8	**71.9**	**87.7**	53.8
Mistral-Small-24B-Instruct-2501	69.9	**72.3**	**92.7**	58.1

Table 15: Model Performance on KMMLU and BoolQ with and without negation. Bold values indicate the higher performance between the negated and affirmative (or original) versions for each model.

Model	0shot		1shot		2shot		5shot		10shot	
	cloze	symbol	cloze	symbol	cloze	symbol	cloze	symbol	cloze	symbol
Midm-2.0-Mini-Instruct (2.3B)	60.4	55.4	70.0	65.0	77.2	69.2	84.0	68.7	87.2	70.9
Midm-2.0-Base-Instruct (11.5B)	66.1	59.6	75.9	72.2	81.6	74.8	86.7	80.7	89.6	83.0
EXAONE-4.0-1.2B	32.2	26.5	44.1	29.2	49.1	41.6	60.2	43.7	65.8	44.3
EXAONE-4.0-32B	51.9	81.5	65.6	86.9	72.0	88.0	80.4	91.0	83.1	92.1
EXAONE-Deep-2.4B	43.2	40.3	57.6	42.6	64.4	43.4	73.4	46.8	79.8	46.7
EXAONE-Deep-7.8B	59.3	43.7	64.9	57.3	69.6	68.2	76.4	75.4	80.1	77.3
EXAONE-Deep-32B	60.1	71.7	69.5	75.5	72.7	76.6	78.7	81.1	82.4	84.2
kanana-1.5-2.1b-base	50.1	45.2	65.0	44.0	72.1	47.0	80.3	46.3	85.4	49.5
kanana-1.5-2.1b-instruct-2505	64.6	57.5	78.3	60.8	82.1	61.9	87.7	58.2	90.3	57.3
kanana-1.5-8b-base	45.8	43.6	60.0	62.0	70.3	62.1	78.8	67.7	82.7	71.8
kanana-1.5-8b-instruct-2505	55.9	56.8	70.6	67.5	77.5	74.4	87.0	81.1	89.4	82.4
HyperCLOVAX-SEED-Text-Instruct-0.5B	38.0	52.7	46.6	37.0	52.0	41.9	61.1	25.6	54.7	45.8
HyperCLOVAX-SEED-Text-Instruct-1.5B	38.5	46.7	53.8	46.4	60.2	52.6	68.7	53.9	74.1	54.0
HyperCLOVAX-SEED-Think-14B	47.5	78.1	70.1	88.1	77.1	89.9	83.3	92.0	86.4	92.1
A.X-4.0-Light (7.2B)	71.3	85.0	72.1	86.1	77.5	88.3	84.9	88.4	86.8	93.3
A.X-4.0 (72B)	76.3	95.8	84.4	95.1	87.9	95.1	90.6	91.1	90.3	95.9
SOLAR-10.7B-v1.0	42.2	42.1	58.8	56.7	65.8	58.0	74.6	62.1	80.3	62.6
SOLAR-10.7B-Instruct-v1.0	64.8	54.8	77.4	61.2	81.0	63.6	85.0	67.0	86.4	68.1
Qwen3-0.6B-Base	41.4	51.6	60.7	49.6	65.8	53.0	73.2	53.8	77.6	58.4
Qwen3-0.6B	39.1	64.6	48.8	49.7	51.1	47.6	56.4	54.1	61.7	56.2
Qwen3-1.7B-Base	64.9	64.7	73.7	72.8	78.9	79.7	86.5	81.9	88.8	84.1
Qwen3-1.7B	54.8	76.1	61.3	78.7	67.3	82.7	77.3	86.2	82.7	84.5
Qwen3-4B-Base	67.8	82.2	76.3	85.5	80.4	87.1	87.2	90.0	89.5	91.2
Qwen3-4B	47.7	80.5	63.6	85.7	69.7	87.4	78.5	89.3	82.7	89.8
Qwen3-8B-Base	60.5	85.5	72.8	88.8	79.7	90.3	87.5	92.5	90.5	92.9
Qwen3-8B	47.7	74.8	64.1	83.0	70.6	86.0	81.6	89.3	85.5	92.0
Qwen3-14B-Base	65.8	88.6	76.9	90.9	80.3	92.2	86.3	93.4	89.5	94.1
Qwen3-14B	47.9	90.6	66.3	92.5	73.6	92.5	83.1	93.0	86.9	94.0
Qwen3-32B	57.6	94.2	75.2	94.5	82.5	94.8	88.8	95.8	89.8	96.9
Llama-3.2-1B	38.4	26.2	47.7	24.0	52.5	24.9	61.7	25.3	69.2	26.2
Llama-3.2-1B-Instruct	37.2	33.6	46.6	29.3	51.7	34.8	61.2	34.9	67.5	33.4
Llama-3.2-3B	45.2	52.0	58.7	50.6	64.0	58.3	74.0	62.7	78.3	64.5
Llama-3.2-3B-Instruct	41.1	61.8	46.5	60.8	51.2	64.4	64.7	69.0	73.5	69.7
Llama-3.1-8B	40.4	53.6	54.7	64.3	63.2	70.7	74.9	74.7	81.3	77.6
Llama-3.1-8B-Instruct	37.4	57.2	52.8	66.8	61.9	68.0	74.1	71.3	80.5	72.4
Llama-3.1-70B	57.5	76.6	70.8	85.2	75.8	88.3	83.0	91.0	87.8	92.4
Llama-3.1-70B-Instruct	50.1	81.0	69.5	86.9	76.4	88.9	85.5	92.3	89.6	93.0
Mistral-7B-v0.3	51.5	46.9	65.2	55.6	72.1	67.3	79.3	77.6	82.9	77.5
Mistral-7B-Instruct-v0.3	65.8	73.0	76.5	71.9	80.0	69.9	84.7	67.7	86.3	65.3
Mistral-Nemo-Base-2407 (12B)	49.7	51.2	69.9	69.9	79.5	76.2	87.0	76.6	90.7	74.8
Mistral-Nemo-Instruct-2407 (12B)	55.6	65.1	73.2	75.1	80.0	78.3	87.1	82.4	90.7	81.6
Mistral-Small-24B-Base-2501	61.3	71.6	75.1	83.9	82.3	89.3	87.7	94.8	90.3	95.2
Mistral-Small-24B-Instruct-2501	60.4	81.9	77.2	87.7	84.1	91.8	90.1	95.8	91.5	96.1
claude-haiku-4-5-20251001	-	92.8	-	94.0	-	95.4	-	96.2	-	96.6
claude-sonnet-4-5-20250929	-	98.1	-	98.6	-	98.4	-	97.6	-	97.0
gpt-4.1-mini	-	83.6	-	88.1	-	88.1	-	90.8	-	92.4
gpt-4.1	-	92.2	-	92.8	-	93.8	-	94.5	-	95.0

Table 16: Model Performance on zero-shot, few-shot setting on Thunder-KoNUBench. Few-shot results are averaged over 3 random seeds (1234, 308, 1028). Red text indicates the model with highest performance in each setting.

Model	Thunder-KoNUBench		KMMLU		BoolQ	
	cloze	symbol	negative	affirmative	original	negated
Midm-2.0-Base-Instruct	84.7 (+18.6)	66.7 (+7.1)	61.7 (+0.0)	73.5 (-0.3)	92.0 (-0.1)	77.6 (+0.3)
Midm-2.0-Mini-Instruct	91.0 (+30.6)	66.9 (+11.5)	64.3 (+1.2)	64.5 (+1.0)	85.3 (-2.0)	59.0 (-9.2)
EXAONE-4.0-1.2B	67.4 (+35.2)	31.7 (+5.2)	50.9 (-0.1)	49.9 (-0.2)	56.8 (+5.4)	49.2 (+0.8)
EXAONE-Deep-2.4B	67.8 (+24.6)	49.8 (+9.5)	48.2 (+0.4)	53.4 (-0.2)	50.9 (-0.1)	48.1 (-0.1)
EXAONE-Deep-7.8B	81.7 (+22.4)	60.2 (+16.5)	49.3 (+0.7)	56.2 (-0.1)	71.7 (+2.1)	50.9 (+2.0)
kanana-1.5-2.1b-base	92.7 (+42.6)	62.2 (+17.0)	56.8 (-0.6)	62.7 (-1.3)	63.0 (+8.2)	49.8 (+1.3)
kanana-1.5-2.1b-instruct-2505	92.1 (+27.5)	67.5 (+10.0)	65.0 (+0.5)	66.9 (+0.1)	71.2 (+4.8)	54.6 (+3.1)
kanana-1.5-8b-base	94.1 (+48.3)	60.3 (+16.7)	62.0 (-4.3)	62.0 (-4.5)	69.2 (+5.5)	51.7 (+0.5)
kanana-1.5-8b-instruct-2505	91.4 (+35.5)	68.5 (+11.7)	67.2 (+0.7)	70.2 (+0.3)	83.6 (+3.3)	61.2 (+4.3)
HyperCLOVAX-SEED-Text-Instruct-0.5B	69.2 (+31.2)	60.3 (+7.6)	61.0 (+1.1)	61.2 (+0.4)	56.3 (+2.8)	48.3 (+0.2)
HyperCLOVAX-SEED-Text-Instruct-1.5B	70.5 (+32.0)	49.6 (+2.9)	62.0 (+0.1)	63.4 (-0.1)	70.6 (+1.3)	50.9 (-0.1)
HyperCLOVAX-SEED-Think-14B	86.4 (+38.9)	86.4 (+8.3)	75.9 (+0.2)	75.0 (+0.2)	87.6 (+0.6)	67.4 (+0.8)
A.X-4.0-Light (7.2B)	85.1 (+13.8)	88.9 (+3.9)	76.3 (+0.1)	76.6 (-0.2)	92.9 (-0.1)	74.9 (+0.0)
SOLAR-10.7B-v1.0	97.6 (+55.4)	79.3 (+37.2)	54.9 (+0.9)	58.4 (+0.2)	58.3 (+7.7)	50.6 (+2.5)
SOLAR-10.7B-Instruct-v1.0	97.2 (+32.4)	74.5 (+19.7)	62.0 (+1.6)	62.5 (-0.2)	88.1 (+2.1)	75.0 (+7.3)
Qwen3-0.6B-Base	78.0 (+36.6)	59.0 (+7.4)	57.5 (+0.6)	54.0 (-0.2)	52.6 (+0.5)	51.3 (+2.1)
Qwen3-0.6B	73.7 (+34.6)	65.5 (+0.9)	50.1 (+0.1)	52.0 (-0.2)	50.4 (+0.0)	48.0 (-0.1)
Qwen3-1.7B-Base	86.0 (+21.1)	66.0 (+1.3)	55.8 (+0.9)	62.2 (+0.2)	63.9 (+2.2)	48.8 (+0.1)
Qwen3-1.7B	74.8 (+20.0)	78.2 (+2.1)	57.8 (+0.6)	60.7 (-0.1)	51.9 (+0.0)	48.4 (-0.2)
Qwen3-4B-Base	89.3 (+21.5)	87.2 (+5.0)	70.5 (+0.3)	68.9 (-0.2)	68.8 (+5.1)	50.8 (+1.9)
Qwen3-4B	79.4 (+31.7)	86.9 (+6.4)	61.2 (+1.7)	66.1 (+0.2)	52.8 (-0.8)	47.9 (-0.2)
Qwen3-8B-Base	89.9 (+29.4)	91.2 (+5.7)	74.0 (+0.4)	72.6 (-0.1)	58.5 (+1.3)	47.8 (-0.2)
Qwen3-8B	87.3 (+39.6)	87.1 (+12.3)	69.5 (+0.7)	70.8 (+0.0)	62.7 (+1.5)	48.9 (+0.0)
Qwen3-14B-Base	89.9 (+24.1)	93.1 (+4.5)	76.2 (-0.1)	69.4 (+0.4)	70.9 (+3.2)	51.4 (+0.6)
Qwen3-14B	86.2 (+38.3)	92.9 (+2.3)	75.5 (+0.0)	70.1 (-0.1)	72.0 (+8.0)	56.6 (+2.5)
Llama-3.2-1B	71.7 (+33.3)	26.1 (-0.1)	51.0 (-0.2)	50.0 (+0.3)	51.1 (+1.7)	48.8 (-1.0)
Llama-3.2-1B-Instruct	68.7 (+31.5)	36.8 (+3.2)	51.0 (+0.4)	52.2 (-0.2)	50.4 (+0.0)	48.0 (-0.1)
Llama-3.2-3B	85.7 (+40.5)	58.5 (+6.5)	57.0 (+0.3)	54.1 (-0.1)	58.6 (+4.4)	48.0 (+0.6)
Llama-3.2-3B-Instruct	81.9 (+40.8)	67.0 (+5.2)	59.4 (-0.1)	58.1 (+0.4)	54.0 (+2.5)	48.3 (+0.4)
Llama-3.1-8B	91.8 (+51.4)	70.4 (+16.8)	60.8 (+0.5)	61.1 (+0.2)	59.3 (+6.3)	48.8 (+0.6)
Llama-3.1-8B-Instruct	90.1 (+52.7)	73.7 (+16.5)	58.4 (+2.1)	59.4 (+1.7)	65.3 (+11.2)	47.9 (-0.2)
Mistral-7B-v0.3	97.4 (+45.9)	80.1 (+33.2)	52.5 (+0.1)	55.5 (-0.8)	65.5 (+7.5)	52.3 (+4.2)
Mistral-7B-Instruct-v0.3	97.0 (+31.2)	91.4 (+18.4)	58.2 (-0.1)	57.9 (-1.2)	79.5 (+2.4)	65.5 (+9.5)
Mistral-Nemo-Base-2407 (12B)	95.4 (+45.7)	69.7 (+18.5)	56.7 (+2.1)	65.9 (+0.6)	71.3 (+4.8)	46.4 (-0.5)
Mistral-Nemo-Instruct-2407 (12B)	93.8 (+38.2)	81.2 (+16.1)	63.0 (+0.8)	63.7 (+0.6)	79.3 (+3.2)	46.7 (-0.9)
Mean Δ over Baseline	34.2	10.5	0.4	-0.1	3.0	0.9

Table 17: Model performance on Thunder-KoNUBench, KMMLU, and KoBest BoolQ after fine-tuning targeting the cloze format. Values in parentheses indicate the performance gain relative to the baseline results.

Model	ARC		Hellaswag	Winogrande
	easy	challenge		
Midm-2.0-Mini-Instruct (2.3B)	79.3 (+0.8)	50.2 (-0.2)	70.3 (+0.1)	66.7 (-0.1)
Midm-2.0-Base-Instruct (11.5B)	84.6 (+0.1)	62.6 (+1.1)	80.5 (+0.1)	73.6 (+0.2)
EXAONE-4.0-1.2B	33.2 (+0.2)	23.7 (-1.0)	32.2 (+1.1)	49.3 (-0.8)
EXAONE-Deep-2.4B	56.9 (+2.6)	38.2 (-0.2)	55.8 (+0.6)	54.0 (-0.7)
EXAONE-Deep-7.8B	66.9 (+1.9)	45.2 (+1.4)	64.3 (+0.7)	57.5 (-0.3)
kanana-1.5-2.1b-base	76.9 (-1.4)	50.0 (-0.6)	66.1 (+0.1)	65.6 (+1.7)
kanana-1.5-2.1b-instruct-2505	76.2 (+2.3)	51.0 (+0.9)	67.1 (+0.2)	62.7 (-0.3)
kanana-1.5-8b-base	82.3 (+1.0)	55.1 (+1.0)	77.5 (+0.1)	72.6 (-0.4)
kanana-1.5-8b-instruct-2505	34.7 (+0.6)	29.9 (+0.0)	78.8 (+0.1)	72.5 (+0.7)
HyperCLOVAX-SEED-Text-Instruct-0.5B	65.2 (-0.2)	37.5 (+0.2)	52.1 (-0.1)	54.9 (+0.3)
HyperCLOVAX-SEED-Text-Instruct-1.5B	66.2 (+0.1)	44.3 (+0.4)	60.9 (+0.0)	57.1 (-0.3)
HyperCLOVAX-SEED-Think-14B	75.4 (+0.5)	54.7 (+0.0)	79.4 (+0.0)	71.2 (+0.0)
A.X-4.0-Light (7.2B)	75.3 (+0.8)	56.2 (+0.3)	77.2 (+0.1)	71.3 (+0.2)
SOLAR-10.7B-v1.0	77.6 (-0.4)	54.2 (-1.3)	83.3 (+0.2)	74.8 (+0.1)
SOLAR-10.7B-Instruct-v1.0	81.4 (+0.3)	61.7 (-0.5)	86.5 (-0.1)	77.0 (+1.9)
Qwen3-0.6B-Base	59.9 (+2.0)	37.9 (-0.1)	53.6 (+0.0)	58.8 (-0.3)
Qwen3-0.6B	31.9 (+0.5)	27.8 (-0.4)	47.1 (-0.1)	55.6 (-0.1)
Qwen3-1.7B-Base	70.2 (+1.9)	45.6 (+0.7)	66.5 (+0.0)	64.8 (+0.4)
Qwen3-1.7B	69.4 (-0.3)	41.6 (-1.2)	60.6 (+0.2)	61.2 (-0.5)
Qwen3-4B-Base	77.2 (+1.4)	52.4 (+0.9)	73.6 (-0.1)	70.9 (+0.4)
Qwen3-4B	78.9 (+0.4)	54.0 (+0.3)	68.6 (+0.2)	66.5 (+0.6)
Qwen3-8B-Base	80.4 (+0.1)	57.3 (+0.6)	78.7 (+0.1)	72.4 (-0.3)
Qwen3-8B	80.9 (+0.0)	56.0 (-0.3)	75.1 (+0.2)	68.0 (+0.4)
Qwen3-14B-Base	82.0 (+0.0)	59.1 (+0.1)	81.4 (+0.0)	73.8 (-0.2)
Qwen3-14B	83.2 (+0.4)	60.9 (+0.7)	78.9 (+0.0)	73.2 (+0.2)
Llama-3.2-1B	61.5 (-0.3)	37.4 (+0.3)	64.3 (+0.2)	60.7 (+0.0)
Llama-3.2-1B-Instruct	63.7 (-0.1)	38.1 (+0.4)	61.6 (+0.0)	62.0 (+0.7)
Llama-3.2-3B	72.2 (+0.4)	46.5 (+0.3)	74.1 (+0.0)	69.6 (-0.1)
Llama-3.2-3B-Instruct	71.5 (+0.4)	46.4 (+0.2)	71.8 (+0.2)	69.2 (+0.4)
Llama-3.1-8B	82.4 (-0.2)	55.5 (+0.7)	79.3 (+0.0)	73.3 (-1.1)
Llama-3.1-8B-Instruct	78.0 (+0.8)	54.8 (+0.6)	79.6 (-0.1)	74.2 (+0.1)
Mistral-7B-v0.3	80.0 (-0.1)	54.3 (-0.1)	80.7 (+0.1)	73.8 (-0.2)
Mistral-7B-Instruct-v0.3	81.2 (-0.6)	60.1 (-0.1)	83.2 (-0.1)	76.1 (+1.3)
Mistral-Nemo-Base-2407 (12B)	83.8 (+0.3)	60.2 (+0.6)	82.9 (-0.1)	76.8 (-0.1)
Mistral-Nemo-Instruct-2407 (12B)	80.9 (+0.3)	59.4 (-0.2)	82.4 (+0.0)	76.9 (+0.0)
Mean Δ over Baseline	0.5	0.2	0.1	0.1

Table 18: Model performance on ARC, Hellaswag, and Winogrande after fine-tuning targeting the cloze format. Values in parentheses indicate the performance gain relative to the baseline results.

Model	Thunder-KoNUBench		KMMLU		BoolQ	
	cloze	symbol	negative	affirmative	original	negated
Midm-2.0-Mini-Instruct (2.3B)	70.0 (+9.6)	92.6 (+37.2)	66.0 (+2.9)	65.2 (+1.7)	87.7 (+0.4)	67.2 (-1.0)
Midm-2.0-Base-Instruct (11.5B)	67.1 (+1.0)	79.6 (+20.0)	61.7 (+0.0)	73.7 (-0.1)	92.2 (+0.1)	77.3 (+0.0)
EXAONE-4.0-1.2B	34.0 (+1.8)	85.0 (+58.5)	51.5 (+0.5)	50.3 (+0.2)	51.4 (+0.0)	48.1 (-0.3)
EXAONE-Deep-2.4B	44.5 (+1.3)	60.2 (+19.9)	48.6 (+0.8)	53.6 (+0.0)	50.6 (-0.4)	48.1 (-0.1)
EXAONE-Deep-7.8B	64.4 (+5.1)	93.2 (+49.5)	48.8 (+0.2)	56.4 (+0.1)	71.9 (+2.3)	49.9 (+1.0)
kanana-1.5-2.1b-base	64.3 (+14.2)	92.6 (+47.4)	59.8 (+2.4)	62.9 (-1.1)	59.9 (+5.1)	49.3 (+0.8)
kanana-1.5-2.1b-instruct-2505	77.2 (+12.6)	97.8 (+40.3)	65.5 (+1.0)	66.7 (-0.1)	67.7 (+1.3)	52.7 (+1.2)
kanana-1.5-8b-base	69.0 (+23.2)	97.4 (+53.8)	68.6 (+2.3)	68.5 (+2.0)	66.1 (+2.4)	51.6 (+0.4)
kanana-1.5-8b-instruct-2505	74.1 (+18.2)	97.4 (+40.6)	68.1 (+1.6)	69.3 (-0.6)	79.8 (-0.5)	56.3 (-0.6)
HyperCLOVAX-SEED-Text-Instruct-0.5B	38.3 (+0.3)	89.5 (+36.8)	62.6 (+2.7)	61.2 (+0.4)	54.6 (+1.1)	47.8 (-0.3)
HyperCLOVAX-SEED-Text-Instruct-1.5B	40.5 (+2.0)	87.0 (+40.3)	62.8 (+0.9)	63.6 (+0.1)	70.3 (+1.0)	51.0 (+0.0)
HyperCLOVAX-SEED-Think-14B	54.5 (+7.0)	98.0 (+19.9)	75.7 (+0.0)	74.6 (-0.2)	87.0 (+0.0)	66.7 (+0.1)
A.X-4.0-Light (7.2B)	73.7 (+2.4)	95.3 (+10.3)	76.6 (+0.4)	76.5 (-0.3)	93.1 (+0.1)	74.4 (-0.5)
SOLAR-10.7B-v1.0	52.5 (+10.3)	99.5 (+57.4)	53.2 (-0.8)	55.2 (-3.0)	53.7 (+3.1)	48.6 (+0.5)
SOLAR-10.7B-Instruct-v1.0	77.5 (+12.7)	98.8 (+44.0)	61.1 (+0.7)	61.5 (-1.2)	80.7 (-5.3)	63.1 (-4.6)
Qwen3-0.6B-Base	45.8 (+4.4)	85.4 (+33.8)	57.4 (+0.5)	54.4 (+0.2)	52.6 (+0.5)	49.2 (+0.0)
Qwen3-0.6B	39.6 (+0.5)	90.3 (+25.7)	50.1 (+0.1)	52.5 (+0.3)	50.2 (-0.2)	48.1 (+0.0)
Qwen3-1.7B-Base	69.0 (+4.1)	87.5 (+22.8)	54.2 (-0.7)	61.8 (-0.2)	64.7 (+3.0)	48.6 (-0.1)
Qwen3-1.7B	55.7 (+0.9)	89.3 (+13.2)	57.1 (-0.1)	59.9 (-0.9)	51.0 (-0.9)	48.1 (-0.5)
Qwen3-4B-Base	72.9 (+5.1)	96.2 (+14.0)	70.2 (+0.0)	68.9 (-0.2)	70.0 (+6.3)	51.0 (+2.1)
Qwen3-4B	47.7 (+0.0)	93.5 (+13.0)	57.2 (-2.3)	64.6 (-1.3)	53.1 (-0.5)	47.9 (-0.2)
Qwen3-8B-Base	68.1 (+7.6)	95.6 (+10.1)	73.6 (+0.0)	72.3 (-0.4)	62.3 (+5.1)	48.6 (+0.6)
Qwen3-8B	52.7 (+5.0)	92.8 (+18.0)	67.1 (-1.7)	70.1 (-0.7)	60.5 (-0.7)	48.6 (-0.3)
Qwen3-14B-Base	77.1 (+11.3)	95.1 (+6.5)	76.1 (-0.2)	70.4 (+1.4)	73.4 (+5.7)	51.6 (+0.8)
Qwen3-14B	51.6 (+3.7)	94.2 (+3.6)	75.6 (+0.1)	70.8 (+0.6)	64.2 (+0.2)	54.0 (-0.1)
Llama-3.2-1B	38.6 (+0.2)	28.4 (+2.2)	50.2 (-1.0)	50.2 (+0.5)	49.2 (-0.2)	48.6 (-1.2)
Llama-3.2-1B-Instruct	37.3 (+0.1)	80.5 (+46.9)	50.2 (-0.4)	51.9 (-0.5)	50.4 (+0.0)	48.0 (-0.1)
Llama-3.2-3B	45.4 (+0.2)	68.1 (+16.1)	57.3 (+0.6)	55.2 (+1.0)	55.3 (+1.1)	47.6 (+0.2)
Llama-3.2-3B-Instruct	49.4 (+8.3)	93.8 (+32.0)	59.3 (-0.2)	57.8 (+0.1)	51.6 (+0.1)	48.2 (+0.3)
Llama-3.1-8B	46.5 (+6.1)	95.2 (+41.6)	61.2 (+0.9)	61.5 (+0.6)	54.4 (+1.4)	48.4 (+0.2)
Llama-3.1-8B-Instruct	50.5 (+13.1)	97.1 (+39.9)	61.4 (+5.1)	60.5 (+2.8)	64.2 (+10.1)	48.8 (+0.7)
Mistral-7B-v0.3	57.9 (+6.4)	99.4 (+52.5)	54.1 (+1.7)	55.6 (-0.7)	73.3 (+15.3)	53.4 (+5.3)
Mistral-7B-Instruct-v0.3	69.2 (+3.4)	99.7 (+26.7)	58.7 (+0.4)	59.1 (+0.0)	79.6 (+2.5)	62.0 (+6.0)
Mistral-Nemo-Base-2407 (12B)	60.0 (+10.3)	97.4 (+46.2)	57.2 (+2.6)	65.4 (+0.1)	71.2 (+4.7)	46.4 (-0.5)
Mistral-Nemo-Instruct-2407 (12B)	67.2 (+11.6)	98.6 (+33.5)	64.2 (+2.0)	65.4 (+2.3)	75.1 (-1.0)	47.8 (+0.2)
Mean Δ over Baseline	6.4	30.7	0.7	0.1	1.8	0.3

Table 19: Model performance on Thunder-KoNUBench, KMMLU, and KoBest BoolQ after fine-tuning targeting the symbol format. Values in parentheses indicate the performance gain relative to the baseline results.

Model	ARC		Hellaswag	Winogrande
	easy	challenge		
Midm-2.0-Mini-Instruct (2.3B)	79.4 (+0.9)	50.8 (+0.4)	70.3 (+0.1)	67.1 (+0.3)
Midm-2.0-Base-Instruct (11.5B)	84.9 (+0.4)	61.8 (+0.3)	80.4 (+0.0)	73.8 (+0.4)
EXAONE-4.0-1.2B	33.2 (+0.2)	24.8 (+0.1)	32.1 (+1.0)	49.3 (-0.8)
EXAONE-Deep-2.4B	56.3 (+2.0)	37.5 (-0.9)	55.7 (+0.5)	54.5 (-0.2)
EXAONE-Deep-7.8B	65.2 (+0.2)	44.8 (+1.0)	63.6 (+0.0)	57.3 (-0.5)
kanana-1.5-2.1b-base	78.5 (+0.2)	51.0 (+0.4)	66.0 (+0.0)	64.6 (+0.7)
kanana-1.5-2.1b-instruct-2505	74.7 (+0.8)	49.9 (-0.2)	66.7 (-0.2)	62.5 (-0.5)
kanana-1.5-8b-base	82.6 (+1.3)	56.6 (+2.5)	77.3 (-0.1)	72.6 (-0.4)
kanana-1.5-8b-instruct-2505	34.2 (+0.1)	29.9 (+0.0)	78.8 (+0.1)	71.6 (-0.2)
HyperCLOVAX-SEED-Text-Instruct-0.5B	65.2 (-0.2)	38.1 (+0.8)	52.3 (+0.1)	55.2 (+0.6)
HyperCLOVAX-SEED-Text-Instruct-1.5B	66.1 (+0.0)	44.1 (+0.2)	60.7 (-0.2)	56.9 (-0.5)
HyperCLOVAX-SEED-Think-14B	75.2 (+0.3)	54.2 (-0.5)	79.4 (+0.0)	71.5 (+0.3)
A.X-4.0-Light (7.2B)	75.0 (+0.5)	56.2 (+0.3)	77.2 (+0.1)	71.6 (+0.5)
SOLAR-10.7B-v1.0	79.0 (+1.0)	55.9 (+0.4)	83.2 (+0.1)	75.5 (+0.8)
SOLAR-10.7B-Instruct-v1.0	80.5 (-0.6)	61.7 (-0.5)	86.8 (+0.2)	76.2 (+1.1)
Qwen3-0.6B-Base	61.0 (+3.1)	38.5 (+0.5)	53.7 (+0.1)	58.1 (-1.0)
Qwen3-0.6B	31.6 (+0.2)	28.0 (-0.2)	47.3 (+0.1)	56.0 (+0.3)
Qwen3-1.7B-Base	70.2 (+1.9)	45.5 (+0.6)	66.6 (+0.1)	64.8 (+0.4)
Qwen3-1.7B	69.3 (-0.4)	43.0 (+0.2)	60.2 (-0.2)	61.6 (-0.1)
Qwen3-4B-Base	77.7 (+1.9)	53.1 (+1.6)	73.7 (+0.0)	70.6 (+0.1)
Qwen3-4B	76.9 (-1.6)	52.3 (-1.4)	68.4 (+0.0)	66.0 (+0.1)
Qwen3-8B-Base	81.1 (+0.8)	57.8 (+1.1)	78.7 (+0.1)	72.4 (-0.3)
Qwen3-8B	80.3 (-0.6)	56.1 (-0.2)	74.9 (+0.0)	68.1 (+0.5)
Qwen3-14B-Base	82.6 (+0.6)	59.6 (+0.6)	81.4 (+0.0)	74.1 (+0.1)
Qwen3-14B	83.1 (+0.3)	60.9 (+0.7)	78.8 (-0.1)	73.3 (+0.3)
Llama-3.2-1B	61.9 (+0.1)	36.8 (-0.3)	64.2 (+0.1)	60.8 (+0.1)
Llama-3.2-1B-Instruct	63.2 (-0.6)	37.5 (-0.2)	61.3 (-0.3)	61.4 (+0.1)
Llama-3.2-3B	72.0 (+0.2)	46.3 (+0.1)	74.2 (+0.1)	69.2 (-0.5)
Llama-3.2-3B-Instruct	71.3 (+0.2)	46.2 (+0.0)	71.6 (+0.0)	68.6 (-0.2)
Llama-3.1-8B	82.3 (-0.3)	55.3 (+0.5)	79.3 (+0.0)	74.4 (+0.0)
Llama-3.1-8B-Instruct	77.3 (+0.1)	54.1 (-0.1)	79.5 (-0.2)	74.3 (+0.2)
Mistral-7B-v0.3	80.5 (+0.4)	54.9 (+0.5)	80.7 (+0.1)	74.4 (+0.4)
Mistral-7B-Instruct-v0.3	81.1 (-0.7)	59.8 (-0.4)	83.4 (+0.1)	74.8 (+0.0)
Mistral-Nemo-Base-2407 (12B)	83.6 (+0.1)	60.0 (+0.4)	83.0 (+0.0)	76.5 (-0.4)
Mistral-Nemo-Instruct-2407 (12B)	80.6 (+0.0)	59.6 (+0.0)	82.5 (+0.1)	76.6 (-0.3)
Mean Δ over Baseline	0.4	0.2	0.0	0.0

Table 20: Model performance on ARC, Hellaswag, and Winogrande after fine-tuning targeting the symbol format. Values in parentheses indicate the performance gain relative to the baseline results.

model name	evaluation method	performance	incorrect choice distribution		
			local negation(%)	contradiction(%)	paraphrase(%)
Qwen3-0.6B-Base	**cloze**	38.94	95.15	3.32	1.53
	symbol	51.56	61.09	12.38	26.53
Qwen3-0.6B	**cloze**	36.92	94.94	3.58	1.48
	symbol	64.56	75.16	7.25	17.58
Qwen3-1.7B-Base	**cloze**	60.28	94.31	4.51	1.18
	symbol	64.64	66.30	7.71	25.99
Qwen3-1.7B	**cloze**	51.79	93.05	5.17	1.78
	symbol	76.01	71.75	7.14	21.10
Qwen3-4B-Base	**cloze**	64.88	93.35	4.66	2.00
	symbol	82.17	82.53	12.66	4.80
Qwen3-4B	**cloze**	45.56	95.14	3.58	1.29
	symbol	80.45	86.85	7.57	5.58
Qwen3-8B-Base	**cloze**	56.31	94.83	3.74	1.43
	symbol	85.44	85.56	8.02	6.42
Qwen3-8B	**cloze**	46.11	93.64	4.48	1.88
	symbol	74.77	86.73	7.72	5.56
Qwen3-14B-Base	**cloze**	62.31	94.21	4.13	1.65
	symbol	88.55	74.83	18.37	6.80
Qwen3-14B	**cloze**	46.18	94.21	4.34	1.45
	symbol	90.50	77.05	18.03	4.92
Qwen3-32B	**cloze**	54.21	94.56	3.91	1.53
	symbol	94.16	89.33	6.67	4.00
Llama-3.2-1B	**cloze**	37.38	93.91	4.35	1.74
	symbol	26.17	35.34	31.01	33.65
Llama-3.2-1B-Instruct	**cloze**	35.59	91.29	6.65	2.06
	symbol	33.49	38.88	18.97	42.15
Llama-3.2-3B	**cloze**	43.46	94.63	3.31	2.07
	symbol	51.95	74.39	11.02	14.59
Llama-3.2-3B-Instruct	**cloze**	38.40	91.66	6.07	2.28
	symbol	61.68	77.64	9.55	12.80
Llama-3.1-8B	**cloze**	39.56	93.17	4.51	2.32
	symbol	53.66	53.28	12.27	34.45
Llama-3.1-8B-Instruct	**cloze**	35.98	91.61	6.08	2.31
	symbol	57.09	66.24	10.34	23.41
Llama-3.1-70B	**cloze**	53.97	95.43	3.55	1.02
	symbol	76.56	77.41	8.97	13.62
Llama-3.1-70B-Instruct	**cloze**	48.05	94.90	4.20	0.90
	symbol	80.92	87.76	6.12	6.12

Table 21: Incorrect choice distributions for the **Qwen3** and **Llama** family under zero-shot conditions.

model name	evaluation method	performance	incorrect choice distribution		
			local negation(%)	contradiction(%)	paraphrase(%)
Mistral-7B-v0.3	**cloze**	49.14	94.18	4.44	1.38
	symbol	46.81	36.75	14.35	48.90
Mistral-7B-Instruct-v0.3	**cloze**	63.16	93.66	4.65	1.69
	symbol	72.90	75.86	6.03	18.10
Mistral-Nemo-Base-2407 (12B)	**cloze**	47.35	93.34	5.18	1.48
	symbol	51.17	63.32	10.85	25.84
Mistral-Nemo-Instruct-2407 (12B)	**cloze**	52.57	92.12	5.91	1.97
	symbol	65.03	92.65	4.90	2.45
Mistral-Small-24B-Base-2501	**cloze**	57.40	94.15	4.57	1.28
	symbol	71.50	63.93	23.50	12.57
Mistral-Small-24B-Instruct-2501	**cloze**	56.93	96.38	3.07	0.54
	symbol	81.78	81.62	14.53	3.85
gpt-4.1-mini	**symbol**	83.49	91.98	6.60	1.42
gpt-4.1	**symbol**	92.13	97.03	1.98	0.99
claude-haiku-4-5-20251001	**symbol**	92.76	84.95	11.83	3.23
claude-sonnet-4-5-20250929	**symbol**	98.05	72.00	28.00	0.00

Table 22: Incorrect choice distributions for the **Mistral**, **GPT-4.1**, and **Claude-Opus 4.5** family under zero-shot conditions.

model name	evaluation method	performance	incorrect choice distribution		
			local negation(%)	contradiction(%)	paraphrase(%)
Midm-2.0-Mini-Instruct (2.3B)	**cloze**	56.00	88.32	9.03	2.65
	symbol	55.30	58.71	12.89	28.40
Midm-2.0-Base-Instruct (11.5B)	**cloze**	61.21	94.38	4.42	1.20
	symbol	59.50	50.77	30.77	18.46
EXAONE-Deep-2.4B	**cloze**	35.75	89.45	7.64	2.91
	symbol	40.19	50.65	13.28	36.07
EXAONE-Deep-7.8B	**cloze**	56.00	93.45	5.13	1.42
	symbol	43.61	42.40	17.40	40.19
EXAONE-Deep-32B	**cloze**	58.33	93.27	4.86	1.87
	symbol	71.65	76.92	9.34	13.74
EXAONE-4.0-1.2B	**cloze**	28.82	86.21	10.28	3.50
	symbol	26.48	36.65	31.04	32.31
EXAONE-4.0-32B	**cloze**	50.16	96.41	3.12	0.47
	symbol	81.46	90.76	5.88	3.36
kanana-1.5-2.1b-base	**cloze**	47.98	94.31	3.59	2.10
	symbol	45.09	42.98	12.06	44.96
kanana-1.5-2.1b-instruct-2505	**cloze**	61.99	91.80	6.35	1.84
	symbol	57.40	63.62	10.97	25.41
kanana-1.5-8b-base	**cloze**	43.54	96.55	2.07	1.38
	symbol	43.54	31.31	17.52	51.17
kanana-1.5-8b-instruct-2505	**cloze**	53.50	94.64	4.19	1.17
	symbol	56.70	51.44	13.67	34.89
HyperCLOVAX-SEED-Text-Instruct-0.5B	**cloze**	35.12	93.88	3.84	2.28
	symbol	52.65	62.17	10.86	26.97
HyperCLOVAX-SEED-Text-Instruct-1.5B	**cloze**	36.14	93.05	5.24	1.71
	symbol	46.57	47.23	23.47	29.30
HyperCLOVAX-SEED-Think-14B	**cloze**	45.56	98.00	1.43	0.57
	symbol	78.04	70.57	14.89	14.54
A.X-4.0-Light (7.2B)	**cloze**	69.08	88.41	8.31	3.27
	symbol	84.89	84.54	5.15	10.31
A.X-4.0 (72B)	**cloze**	73.83	91.67	5.65	2.68
	symbol	95.72	87.27	7.27	5.45
SOLAR-10.7B-v1.0	**cloze**	40.50	93.98	4.71	1.31
	symbol	41.98	44.83	14.23	40.94
SOLAR-10.7B-Instruct-v1.0	**cloze**	61.29	92.96	5.03	2.01
	symbol	54.67	77.84	11.68	10.48

Table 23: Incorrect choice distributions for the Korean Models (**mi:dm 2.0**, **EXAONE**, **Kanana 1.5**, **HyperCLOVAX**, **A.X 4.0**, and **SOLAR** model family) under zero-shot conditions.