Lost the Negation or Lost in Negation

Vennela Bairi, Parameswari Krishnamurthy
Language Technology Research Centre, IIIT Hyderabad
vennela.bairi@research.iiit.ac.in, param.krishna@iiit.ac.in

Abstract

Despite negation being one of the core element in any language, it remains a challenging phenomenon for modern Large Language Models(LLMs). Recently, there have been growing efforts to evaluate how models handle negation. However, the existing probing datasets are mostly English-centric. To facilitate evaluation for Indian Languages especially Telugu, which has complex morphological features, we present **NEGTEG** benchmark. This benchmark is a test suite that contains 5 tasks: Negation Detection, Negation Translation, Paraphrase Detection, Sentiment Analysis and Polarity Flipping. The test suite is designed based on strong linguistic analysis and includes annotations of different negation types. This helps us evaluate how models perform across various forms of negation. We use the benchmark to probe the negation handling capabilities of multilingual language models at different levels and our evaluation reveals that most of the models struggle significantly with Telugu negation across all tasks.

Keywords: Negation, Benchmark, Evaluation

1. Introduction

Negation is a fundamental component of natural language and is used to express non-existence, denial, or contradiction. It can change the truth value and can completely flip the meaning of a sentence. Despite its significance, there is ample evidence showing that state-of-the-art NLP models struggle with processing negation effectively. In recent years, there has been a growing effort to evaluate the LLMs' capabilities to handle negation. According to Truong et al. (2022), Khandelwal and Sawant (2020), models are insensitive to the presence of negation. Hossain et al. (2020), Tang et al. (2021), Dobrovolska et al. (2024), Alshargabi et al. (2022), Abuhammad and Ahmed (2024), Gupta and Joshi (2021), reports that the presence of negation significantly impacts downstream task quality like Translation and Sentiment Analysis. Truong et al. (2023), Bhattarai and Erk (2024), She et al. (2023) evaluate how models struggle when asked a logical composition including negation for NLI task. Though there are many existing Benchmarks to evaluate, most of them are inclined towards English. Our work primarily focuses on the systematic evaluation of LLMs' performance for morphologically rich language, Telugu, where negation is not only expressed through words like *no, not, never* as in English, but is also morphologically encoded. This linguistic complexity motivated us to manually curate NEGTEG Benchmark: a test suite of 5 tasks that helps evaluate models' capabilities at different levels of linguistic and logical complexity.

1. **Negation Detection** checks whether models can identify negation cues, evaluating basic syntactic awareness of negation.

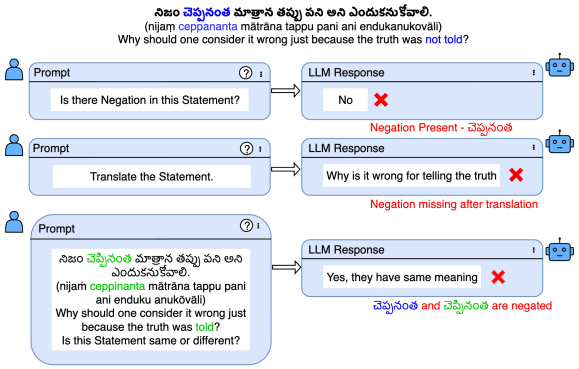

Figure 1: Failures of LLMs' in detecting, preserving, and misinterpretation of semantically opposite forms over negation in Telugu sentences.

2. **Translation** evaluates whether negation is accurately preserved when translating from the source to the target language.
3. **Paraphrase Detection** evaluates whether models can distinguish between semantic equivalence and meaning reversal caused by negation within sentence pairs.
4. **Sentiment Classification** evaluates whether models can correctly determine the sentiment of a sentence in the presence of negation, probing the model's understanding of how negation shifts the sentiment of a sentence.
5. **Polarity Flipping** requires models to rewrite a sentence by flipping the polarity of a specified word or phrase, evaluating whether they can manipulate the negation and not just understand it.

In this paper, we address the lack of resources and evaluation frameworks for negation in Telugu.

Proceedings of the Second workshop on Challenges in Processing South Asian Languages (CHiPSAL2026) @ LREC, pages 18–31

Task	Input / Context	Expected vs Model's Response
Negation Detection	మీ ఆనందంలో పాలు పంచుకొనేందుకు నేను ఉండనే అని దిగులుగా ఉంది. (mī ānaṃdaṃlō pālu pañcukōnēṃduku nēnu uṃḍanē ani digulugā uṃdi) I am sad that I will not be there to share in your happiness.	**Actual:** NEGATED **Model:** NOT_NEGATED
Translation	నిజం చెప్పనంత మాత్రాన తప్పు పని అని ఎందుకనుకోవాలి. (nijaṃ ceppananta mātrāna tappu pani ani endukanukovāli) Why should one consider it wrong just because the truth was not told?	**Expected Output:** Why should one consider it wrong just because the truth was not told? **Model:** Why should one consider it a wrong act just for telling the truth?
Paraphrase Detection	**S1:** నాకు నచ్చని పని చేసినందుకు నా అహం దెబ్బ తింది. (nāku naccani pani cēsinanduku nā ahaṃ debba tiṃdi.) My ego was hurt because I did work that I did not like. **S2 (Paraphrase):** నాకు ఇష్టంలేని పని చేయడంవల్ల నా అహంకారానికి భంగం కలిగింది. (nāku iṣṭaṃlēni pani cēyaḍaṃvalla nā ahaṃkārāniki bhaṃgaṃ kaligindi.) My pride was wounded by doing work I disliked. **S2 (Negated):** నాకు నచ్చని పని చేసినందుకు నా అహం దెబ్బ తినలేదు. (nāku naccani pani cēsinanduku nā ahaṃ debba tinalēdu.) My ego was not hurt by doing work I disliked.	*For S2 Paraphrase:* **Actual:** PARAPHRASE **Model:** PARAPHRASE *For S2 Negated:* **Actual:** NEGATED **Model:** PARAPHRASE
Sentiment Classification	స్మోక్ ఫ్రీ నినాదానికి ఫైవ్ స్టార్ హోటళ్లకు కూడా మినహాయింపు లేదు. (smōk free ninādāniki fayiv sṭār hōṭaḷlaku kūḍā mināhāyiṃpu lēdu) Even five-star hotels are not exempt from the smoke-free initiative.	**Actual:** POSITIVE **Model:** NEGATIVE
Polarity Flipping	నేను ఆమెని చూడనట్టు నటిస్తూ తప్పించుకున్నాను. (nēnu āmēni cūḍanattu naṭistū tappiñcukunnānu) I pretended not to see her and escaped. **Verb to flip:** తప్పించుకున్నాను (tappiñcukunnānu) escaped	**Expected Output:** నేను ఆమెని చూడనట్టు నటిస్తూ తప్పించుకోలేదు. (tappiñcukōlēdu) didn't escape **Model:** నేను ఆమెని చూడనట్టు నటిస్తూ తప్పించుకున్నాను. (tappiñcukunnānu) escaped

Table 1: Example instances from each task in the benchmark, with Expected and Model's Response

We release the benchmark publicly[1]. Table 1 illustrates a sample for each task. To best of our knowledge, this is the first work on evaluating LLMs on downstream tasks for negation in Telugu. To this end, our major contributions are:

1. We introduce a manually extracted and verified comprehensive list of Telugu negative morphemes, covering the full range of negation patterns possible in Telugu.
2. We present **NEGSEED**, a manually curated set of example sentences for each of the above morphemes, serving as a foundational resource for Telugu negation.
3. We present **NEGTEG**, a test suite of 5 tasks designed to systematically probe LLMs' capabilities to handle negation across varying levels of linguistic and logical complexity.
4. We evaluate a diverse set of LLMs on this benchmark, providing a thorough analysis of how effectively the models process negation in Telugu.

[1] https://github.com/VennelaBairi/NEGTEG

2. Related Work

Several works have investigated the negation handling capabilities of models by creating task-specific datasets and analyzing model outputs.

Negation Detection: Studies have exposed how state-of-the-art NLP models fail to find negation cues. Khandelwal and Sawant (2020) reviewed previous literature including rule-based systems, ML classifiers, CRF models, CNNs, BiLSTMs. Okpala et al. (2022) demonstrated methods for detecting negations in a sentence by evalu-

ating the lexical structure of the text via word-sense disambiguation. Martinis et al. (2024) reviewed how negation detection plays an important role in understanding clinical documentation. Other works highlighted the importance of detecting the scope of negation for downstream tasks such as sentiment analysis (Makkar et al., 2024) and also in Information Extraction (Weller et al., 2024).Shah and Pareek (2024) conducted a survey on negation detection in Hindi, exploring both BERT-based and hybrid rule-based approaches, finding that the complex syntactic structure of Hindi differs greatly from English, making the direct application of English models inadequate.

Translation: Fancellu and Webber (2015) conducted an error analysis while translating Chinese into English, which involves negation. They found that identifying the negation cue right is easier than translating the event and scope correctly. Hossain et al. (2020) reported that negation is often mistranslated and has the potential to cause loss of information. Tang et al. (2021) similarly reports that that negations are under-translated and according to a qualitative analysis by Alshargabi et al. (2022), students faced difficulty translating sentences containing double negation

Paraphrase Detection: Work by Ettinger (2020) on Pretrained Language models like BERT and ELMo and extended work by Truong et al. (2023) on autoregressive models state that models are insensitive to the contextual impacts of negation. Anschütz et al. (2023) designed the CANNOT Dataset for paraphrased or negated classification and fine-tuned a sentence transformer to improve their negation sensitivity. Vahtola et al. (2022) finds that LLMs underestimate the impact of negations on how much they change the meaning of a sentence. Rezaei and Blanco (2024) finds that paraphrasing in affirmative terms improves the performance of negation understanding. The benchmarks constructed by Hartmann et al. (2021) and So et al. (2025) evaluate how the models understand and reason in the NLI task.

Sentiment analysis: Despite of having positive or negative keywords, the sentiment can be easily flipped by inserting negations, making sentiment analysis challenging. The study by Mamatha Mylarappa et al. (2023), Ilmawan et al. (2024), Mukherjee et al. (2021) highlight the importance of considering negation in the text to improve the performance of sentiment analysis. Cruz et al. (2016)'s work on negation and speculation information in review texts says that the correct identification of cues and scopes is vital for the task of sentiment analysis and opinion mining systems.

Polarity Flipping: Experiments conducted by Chen et al. (2023) and Ghasiya and Sasahara (2026) reveal that LLMs frequently fail to generate valid sentences grounded in negative common sense knowledge. They further find that though guided prompts improve the quality of generated negatives, they often generate ambiguous statements, or statements with negative keywords but a positive meaning.

Negation	Example	Example Sentence
Explicit	కాదు (kādu) is not	అతను విద్యార్థి కాదు. (atanu vidyārthi kādu.) He is not a student.
Morphological	వెళ్ళకు (veḷḷaku) do not go	అక్కడికి వెళ్ళకు. (akkadiki veḷḷaku.) Do not go there.
Lexical	అసాధ్యం (asādhyam) impossible	ఇది అసాధ్యం. (idi asādhyam.) This is impossible.

Table 2: Types of negation in Telugu with illustrative examples across four categories.

Pretrained Language Models like BERT, have been shown to fail in distinguishing between negated and non-negated sentences (Hosseini et al., 2021), and automatic evaluation metrics such as BERTScore are equally insensitive to negation (Hanna and Bojar, 2021).

These observations, along with the lack of resources for Indian languages, collectively motivate the creation of a benchmark for evaluating downstream tasks in Telugu.

3. Benchmark Creation

3.1. Base Corpora Collection

For creating the benchmark we used Telugu Books dataset available on Kaggle[2] which contains 25,000 books. This extensive collection of original Telugu books in various genres ensures that the benchmark captures the authentic nuances and natural flow of the language. The Indic Sentence Tokenizer[3] was applied on the books dataset to extract sentences.

3.2. Negation Types and Coverage

Languages express negation in different forms and types. When separate negative words are used, it is called Explicit Negation. When specific morphemes are added to the words, it is called Morphological Negation. Other ways include Lexical Negation, where certain words carry negative meaning, such as impossible or unlikely. Table 2 illustrates types of negation with examples.

Telugu, being a morphologically rich language, expresses negation not only through Explicit, Lex-

[2] https://www.kaggle.com/datasets/sudalairajkumar/telugu-nlp

[3] https://indic-nlp-library.readthedocs.io/en/latest/indicnlp.tokenize.html

ical forms, but predominantly by attaching negative morphemes directly to words, mostly verbs. For example, for the verb చెప్పు (cheppu -to say), adding negative morphemes such as _అకు (_aku) and _అను (_anu) results in forms like చెప్పకు (ceppaku) (do not say) and చెప్పను (cheppanu) ("I will not say").

Morpheme	Example
అని_అట్లు (ani_aṭlu)	చెప్పనట్లు (ceppanaṭlu) as if not said'
అని (ani)	చెప్పని (ceppani) that which is not said
అ_లేదు (a_lēdu)	చెప్పలేదు (ceppalēdu) did not say
అడం_లేదు (aḍaṃ_lēdu)	చెప్పడంలేదు (ceppaḍaṃlēdu) is not saying
అని_అప్పుడు (ani_appuḍu)	చెప్పనప్పుడు (ceppanappuḍu) when not saying
అని_అంత (ani_anta)	చెప్పనంత (ceppananta) to the extent of not saying
అ_కూడదు (a_kūḍadu)	చెప్పకూడదు (ceppakūḍadu) must not say

Table 3: Telugu Negation Morphemes illustrated with the root verb చెప్పు (cheppu) 'to say'.

In Telugu there are approximately 800 distinct morpheme combinations (Dasari et al., 2023). We examined this complete set to identify morphemes that carry a negative meaning, as shown in Table 3. Through manual extraction and cross-verification, we identified 104 negative morphemes. Of these, we used 94 in this study, excluding 10 traditional forms that rarely appear in contemporary Telugu.

Building a dataset with this extensive negative morpheme coverage required a hybrid collection strategy. As a first step, we applied a Morphological Analyzer Dasari et al. (2023) to the tokenized sentences. This tool decomposes each word form into its constituent morphemes, providing seven fields of information: Root, Lexical category (lcat), Gender (gen), Number (num), and Person (per), Case, Suffix(when lcat is noun)/Tense-Aspect-Mood(TAM) (when lcat is verb). Table 4 illustrates this analysis for two example words.

Word	Root	lcat	Gen	Num	Per	Case	TAM
చెప్పలేదు	చెప్పు	V	any	any	any	–	అ_లేదు
చెప్పవు	చెప్పు	V	n	pl	3	–	అ

Table 4: Morphological analysis of example negative verb forms (V- Verb and pl- plural)

Using the morphological decomposition produced by this analyzer, we matched the TAM field of each tokenized word against our list of 94 negative morphemes. This automatic extraction identified approximately 3,000 sentences that covered 54 of the 94 negative morphemes. From these, we selected 270 sentences covering 5 sentences per morpheme, ensuring diversity in sentence structure and coverage across Negative Polarity Items (NPIs) DILIP and KUMAR (2019) (Table 10). All selected sentences were reviewed by native Telugu speakers to verify the grammatical correctness and naturalness.

The remaining 40 morphemes were underrepresented in the extracted data, either because they are relatively uncommon or because their morphological variants made them difficult to capture through pattern matching. For these, we employed human annotators to generate sentences. Annotators were provided with 800 sentences from the Telugu Books dataset as reference and were asked to modify them to incorporate the missing negative morphemes. They were also given detailed specifications for each morpheme, including its morphological structure and example words. This manual process yielded 200 sentences covering all 40 remaining morphemes.

Through this combined process of automatic extraction (270 sentences), human generation (200 sentences), followed by validation by native speakers, we compiled a final set of 470 negated sentences, which we refer to as the **NEGSEED** set. This set serves as the foundation for the datasets used in each test suite of our benchmark. NEGSEED is designed to comprehensively evaluate whether LLMs can handle the full morphological and syntactic range of Telugu negation. Since sentences are sourced from authentic Telugu literature, NEGSEED reflects natural linguistic diversity rather than artificially constructed examples. It serves as a reusable reference that can be extended to other domains beyond book-based text, including conversational, web, and task-specific corpora.

3.3. Test Suite for Benchmark

To evaluate LLMs' capabilities in understanding and handling negation, we designed 5 tasks at increasing levels of complexity, starting from whether models can detect the presence of negation, how negation affects downstream tasks, whether they could differentiate between the negated and non-negated pairs, how well they reason and understand when negation is involved, how well models can flip the polarity of a sentence. The following subsections describe how the data for each task was prepared and Table 1 illustrates a sample of each task.

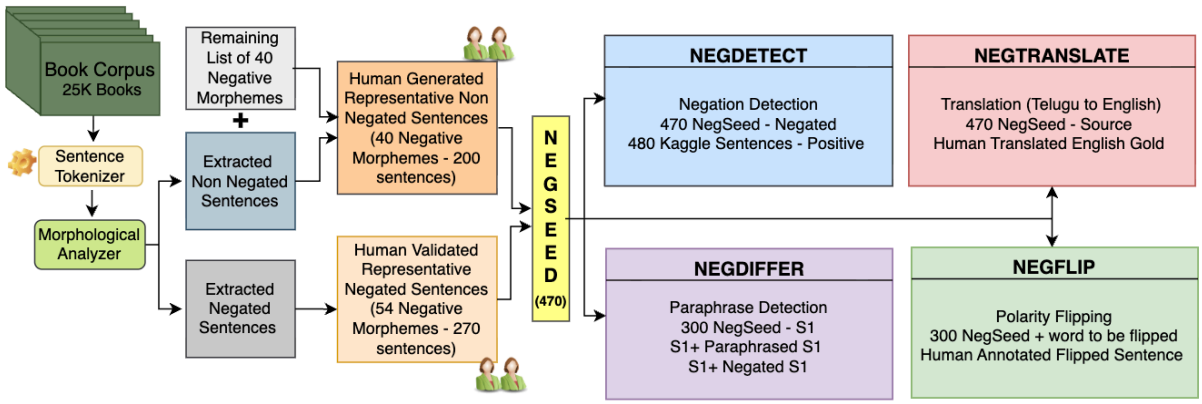

Figure 2: Pipeline for building NEGTEG, a negation-focused evaluation benchmark. NegSeed is first constructed through corpus filtering and human validation, and subsequently expanded into four test suites: NEGDETECT, NEGTRANSLATE, NEGDIFFER, and NEGFLIP.

3.3.1. NEGDETECT: Negation Detection

Our primary motive to include this task in the test suite is to check whether the LLMs can at least detect the presence of negation. This capability is a prerequisite for more complex tasks. The test set has 950 sentences in total with the labels NEGATED and NON NEGATED. It consists of 470 negated sentences from the NEGSEED set and 480 non-negated sentences drawn from the same Telugu Books corpus to ensure class balance. All non-negated sentences were validated by annotators to confirm the absence of negative morphemes and explicit negation words. The two classes are defined as follows:

- **NEGATED:** Sentences that contain explicit negative words and Negative morphemes.
- **NON-NEGATED:** Sentences that have Lexical negatives and other non-negated sentences.

3.3.2. Translation

Translation is a very common task, but this test set evaluates how well LLMs and translation models preserve negation during translation. We used the NEGSEED set as input. Each sentence was translated into English by human translators. They were given explicit instructions to preserve the negation in the target language rather than omitting or paraphrasing it, ensuring that the negation is explicitly reflected in the English translation and not merely implied by the overall meaning.

3.3.3. Paraphrase Detection

This task evaluates whether the model can understand how negation alters the meaning of a sentence and the logical relationship between two sentences. For this test set, a subset of 300 sentences from NEGSEED served as Sentence 1. For each Sentence 1, two versions of Sentence 2 were manually generated as follows:

- **NEGATED S2:** A sentence whose meaning is reversed by flipping the polarity of the whole sentence or a specific part of Sentence 1.
- **PARAPHRASED S2:** A sentence that retains the same meaning as Sentence 1 but uses different words or structures.

Given Sentence 1 and Sentence 2, the model should predict the correct label NEGATED or PARAPHRASED. This task requires the model to understand how negation alters the meaning and logical relationship between statements.

3.3.4. Sentiment Task

To construct the benchmark, we utilized two publicly available Telugu sentiment datasets: ACTSA and SentiRama[4]. We utilized the complete ACTSA dataset (Mukku and Mamidi, 2017). From SentiRama, we consider only the Books Reviews and Product Reviews subsets.

To keep the analysis straightforward, we considered only two classes: POSITIVE and NEGATIVE. Samples labeled as NEUTRAL were discarded, as binary classification provides a clearer understanding for evaluating how negation shifts sentiment polarity. This task directly tests whether models understand the effects of negation on sentiment. A model may correctly predict positive sentiment for a word like *'good'*, but must also recognize that the addition of negation, as in *'not good'*, reverses the overall sentiment. A model that fails to account for negation will likely misclassify such sentences.

[4]SentiRama:https://ltrc.iiit.ac.in/showfile.php?filename=downloads/sentiraama/

This serves as a strong indicator of a model's understanding of negation in a downstream task.

3.3.5. Polarity Flipping

This task evaluates the model's ability to generate semantically opposite sentences through controlled polarity flipping. Since a sentence can produce multiple valid negations depending on which constituent is negated, we explicitly specify the word or phrase whose polarity is to be flipped (Table 5). Each sample in the dataset consists of an input sentence, the target word or phrase to be flipped, and a human-annotated reference sentence with reversed polarity. Two annotators independently annotated 150 samples each, followed by cross-validation to ensure consistency in target selection and reference quality.

Sentence: నాకు నచ్చని పని చేసినందుకు నా అహం దెబ్బ తింది. (nāku naccani pani cēsinaṃduku nā ahaṃ debba tindi) My ego was hurt because I did work that I did not like.	
Word to be Flipped	**Polarity Flipped Sentence**
చేసినందుకు (cēsinaṃduku) *'because I did'*	నాకు నచ్చని పని చేయనందుకు నా అహం దెబ్బ తింది. (nāku naccani pani cēyanaṃduku nā ahaṃ debba tindi) My ego was hurt because I did not do the work I disliked.
నచ్చని (naccani) *'did not like'*	నాకు నచ్చిన పని చేసినందుకు నా అహం దెబ్బ తింది. (nāku naccina pani cēsinaṃduku nā ahaṃ debba tindi) My ego was hurt because I did work that I liked.

Table 5: Multiple valid negations of the original sentence, each targeting a different constituent.

4. Evaluation and Results

4.1. Evaluating Negation Detection

This experiment evaluates whether the models can identify the presence of explicit negation and negative morphemes in a sentence. This is a preliminary, foundational check to assess whether the models are sensitive to negation cues or tend to overlook them. Models are explicitly prompted to classify a sentence as NEGATIVE only if it contains explicit negative words or negative morphemes. Sentences conveying negation through lexical as well as all affirmative sentences, must be classified as NON_NEGATED(Appendix B.1).

Table 6 reports Accuracy, Precision and Recall for both NEGATED and NON_NEGATED classes, and Macro F1. Larger models such as GPT-OSS 120b, GPT-OSS 20b, and Llama-3.3-70b-versatile tend to achieve higher NonNegRecall but lower NegRecall. This indicates that out of all negatives, fewer negated sentences are correctly identified compared to non-negated ones. However, smaller models with competitive accuracy, such as Meta-Llama-3-8B, Nemotron-Cascade-8B, and Qwen3-8B, exhibit higher NegRecall than NonNegRecall, indicating a tendency to predict negation more liberally. Models with relatively lower accuracy show even more heavily skewed predictions, collapsing almost entirely towards one class. Overall, most models struggle to correctly identify both negated and non-negated sentences, highlighting the difficulty of recognizing negation in Telugu.

4.2. Evaluating Translation

We gave the models a simple prompt: "*Translate the following Telugu sentence to English.*" This prompt, without any additional guidance regarding negation, is used to evaluate how negation is handled in a downstream task without giving any explicit instructions about the negation. The idea behind this test set is to examine whether the negation is semantically understood in a downstream task and translated from source to target language.

Along with the LLMs used for all other tasks, we also evaluated dedicated translation models including Google Translate, Rotary-Indictrans2, Bhashaverse (Mujadia and Sharma, 2024) and Sarvam Translate. Although standard translation metrics, such as BLEU, chrF++, METEOR, and COMET (Table 7), measure the overall quality of the translation, they do not specifically consider whether the negation present in the source sentence is preserved after translation. Therefore, we employed human evaluators to validate if negation is preserved after translating. The guidelines given to evaluators are in Appendix A.1.

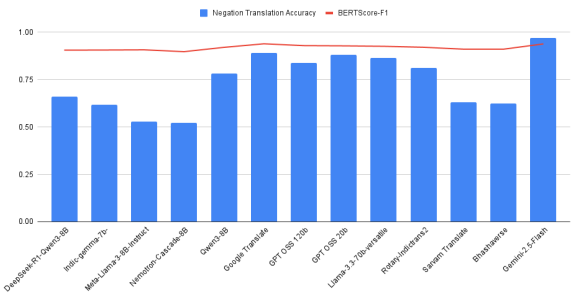

Figure 3: Human-Evaluated Negation Preservation Percentage Across Translation Models

Model	Accuracy	NegPrecision	NonNegPrecision	NegRecall	NonNegRecall	MacroF1
DeepSeek-R1-Qwen3-8B	57.53	53.93	83.49	95.94	20.04	50.69
Indic-gemma-7b-Navarasa	50.63	66.67	50.58	0.43	99.79	33.99
Meta-Llama-3-8B-Instruct	83.47	83.09	83.86	83.62	83.33	83.47
Nemotron-Cascade-8B	81.58	79.32	84.12	84.89	78.33	81.57
Qwen3-8B	79.26	74.86	85.29	87.45	71.25	79.15
Gemma-7b-it	59.16	80.60	55.64	22.98	94.58	52.91
GPT-OSS 120b	94.42	98.60	90.98	90.00	98.75	94.40
GPT-OSS 20b	91.54	97.85	85.09	87.02	97.48	91.49
Llama-3.3-70b-versatile	78.35	97.01	61.94	69.15	95.93	78.01
Sarvam-1	64.74	89.47	59.31	32.55	96.25	60.56
Gemini-2.5-flash	99.37	99.36	99.36	99.38	99.38	99.37

Table 6: Performance of LLMs on NEGDETECT, reporting Accuracy, Precision, Recall, and Macro F1 for NEGATED and NON_NEGATED classes.

Model	**BLEU**	**chrF++**	**METEOR**	**COMET**
DeepSeek-R1-Qwen3-8B	12.48	34.72	0.39	0.68
Indic-gemma-7b-Navarasa	13.10	33.80	0.39	0.69
Meta-Llama-3-8B-Instruct	13.12	35.58	0.40	0.69
Nemotron-Cascade-8B	10.48	31.67	0.36	0.65
Qwen3-8B	19.05	41.07	0.47	0.74
Google Translate	33.47	53.27	0.60	0.80
GPT OSS 120b	24.60	47.19	0.55	0.76
GPT OSS 20b	23.41	46.06	0.54	0.76
Llama-3.3-70b-versatile	22.95	45.26	0.51	0.76
Rotary-Indictrans2	18.43	40.08	0.46	0.74
Sarvam Translate	11.65	33.13	0.38	0.71
Bhashaverse	15.61	35.88	0.41	0.68
Gemini-2.5-Flash	28.57	51.33	0.59	0.80

Table 7: Performance of LLMs and Translation models on the Translation task, evaluated using BLEU, chrF++, METEOR, and COMET metrics.

From Figure 3, it is evident that models are unable to accurately translate negation. Notably, while the difference in BERTScore across models is not substantial, the negation preservation accuracy varies significantly, suggesting that standard metrics alone may not capture how well negation is preserved (Hosseini et al., 2021). Our manual qualitative analysis reveals some common error patterns: the negation is completely omitted in the translation, it is mistranslated, or the negation is dropped from the intended word or phrase and incorrectly applied to another (Appendix Table 11).

4.3. Evaluating Paraphrase Detection

In the Negation detection task, the objective was relatively straightforward: given a single sentence, determine whether it contains negation. In contrast, the paraphrase detection task is more challenging. Here, the model is provided with sentence pairs, either a sentence paired with its negated version or a sentence paired with its paraphrased version. The models are prompted to determine whether a given sentence pair conveys the same meaning or opposite meaning (at least in part) which helps distinguish between semantic equivalence and contradiction introduced by negation.

Although overall accuracy scores are low for several small models, the performance drop is primarily driven by errors in negated pairs. Recall analysis (Table 8) shows that paraphrased instances are identified with relatively high recall, whereas negated pairs are detected much less reliably, indicating difficulty in recognizing meaning reversal caused by negation. Even when a negated sentence appears alongside its original, models often fail to capture the semantic contradiction. However, Larger models such as GPT-OSS 120B and 20B, and Llama-3.3-70B exhibit higher NegRecall than ParaRecall, while smaller models show the opposite trend. This suggests that smaller models tend to treat most sentence pairs as semantically equal though they are not.

4.4. Evaluating Sentiment Classification

We first categorize the sentiment test set into sentences containing negation and sentences without negation, using a morphological analyzer (Dasari et al., 2023) to identify the presence of explicit negative words or negative morphemes. This categorization allows us to directly compare model perfor-

Model	Accuracy	ParaPrecision	NegPrecision	ParaRecall	NegRecall
DeepSeek-R1-Qwen3-8B	68.85	69.58	68.09	69.06	68.63
Indic-gemma-7b-Navarasa	53.89	52.24	61.86	86.88	21.13
Meta-Llama-3-8B-Instruct	51.77	50.82	86.67	99.29	4.58
Nemotron-Cascade-8B	62.19	57.00	93.75	98.23	26.41
Qwen3-8B	83.22	77.10	92.76	94.33	72.18
Gemma-7b-it	49.82	49.82	0.00	100.00	0.00
GPT-OSS 120b	88.16	90.26	86.29	85.46	90.85
GPT-OSS 20b	89.17	91.67	86.96	86.12	92.20
Llama-3.3-70b-versatile	87.10	96.86	80.76	76.60	97.54
Telugu-Llama2-7B-Instruct	49.82	49.82	0.00	100.00	0.00
Gemini-2.5-Flash	97.17	98.19	96.21	96.10	98.24

Table 8: Performance of LLMs on NEGDIFFER, reporting Accuracy, Precision, and Recall for PARAPHRASED and NEGATED classes.

	Flip Quality Metrics				Similarity Metrics		
Model	Correct (%)	(5,4) (%)	3 (%)	(2,1) (%)	BLEU	BERTScore	CharEdit$_{norm}$
DeepSeek-R1-Qwen3-8B	40.15	22.78	21.62	55.60	0.6133	0.9007	0.3044
Indic-gemma-7b-Navarasa	20.45	15.24	10.41	74.35	0.5562	0.8966	0.2969
Meta-Llama-3-8B-Instruct	18.42	10.90	7.89	80.45	0.6280	0.8975	0.2794
Qwen3-8B	58.58	42.54	22.76	34.70	0.6640	0.9216	0.2169
Gemma-7b-it	1.14	0.76	1.14	98.11	0.6209	0.9019	0.2836
GPT-oss-120b	58.21	47.76	20.15	32.09	0.7056	0.9381	0.1641
GPT-oss-20b	57.98	47.47	18.29	34.24	0.6896	0.9325	0.1812
Llama-3.3-70b-versatile	70.00	55.93	23.33	20.74	0.6898	0.9282	0.1860
Gemini-2.5-Flash	86.54	78.08	13.08	8.85	0.7393	0.9492	0.1367

Table 9: Performance of LLMs on NEGFLIP, reporting Flip Quality metrics (percentage of correctly flipped instances and distribution of Meaning scores ((5,4), 3, and (2,1))) and similarity metrics (BLEU, BERTScore, and normalized character edit distance).

mance across the sentences that contain explicit and morphological negations and others. From Figure 4, it is observed that accuracy is higher for sentences without negation. This suggests that negation introduces additional complexity.

From Appendix Table 12, it is evident that, recall for the positive sentiment class under negated conditions is consistently low across most models, indicating that models struggle to correctly identify positive sentiment when negation is present. This is likely because negation words pull the model toward predicting negative sentiment directly, rather than understanding the compositional meaning of the negated sentence. Conversely, recall for the negative sentiment class under negated conditions remains high, suggesting that models associate negation cues directly with negative sentiment rather than reasoning about the actual sentiment of the sentence.

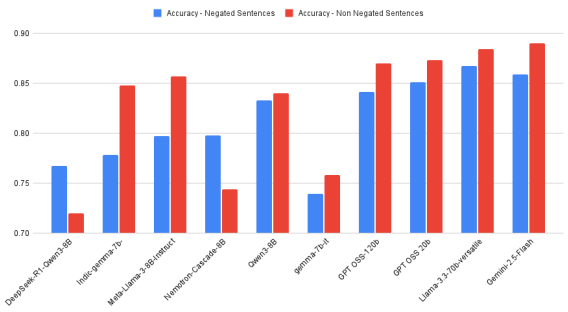

Figure 4: Comparison of overall sentiment classification accuracy on sentences containing negation and sentences without negation.

4.5. Evaluating Polarity Flipping

This experiment assesses whether models can generate a negated variant of a given sentence by flipping the meaning of a specified target word or phrase while preserving the rest of the sentence's structure and semantics. The human evaluation guidelines for this task are detailed in the Appendix A.2. The Results in Table 9 show that Polarity Flipping is the hardest task in this benchmark. Even large models like GPT-OSS 120b and GPT-OSS 20b achieve only 58% correctness, with only Gemini-2.5-Flash, Llama-3.3-70b-versatile performing reasonably well. However, models that fail to correctly flip the polarity still achieve high BERTScores, meaning the generated sentences look similar to the input but do not carry the intended opposite meaning highlighting the inadequacy of automated metrics.

5. Conclusion

In this work, we present **NEGTEG**, a comprehensive benchmark to evaluate negation handling in Telugu on five tasks: Negation Detection, Negation Translation, Paraphrase Detection, Sentiment classification, and Polarity flipping. Our evaluation of models reveals that negation in Telugu remains a persistent challenge even for state-of-the-art models. Models tend to be biased toward non-negative predictions, struggle to preserve negation in translation, and fail to distinguish semantically inverted sentence pairs in paraphrase detection. Furthermore, models struggle with downstream tasks in the presence of negation, particularly in sentiment classification and polarity flipping. A recurring observation is that automated metrics remain high even when models fail semantically, underscoring the need for human evaluation. Overall, larger models consistently outperform smaller ones, yet even the strongest models struggle with deep compositional understanding of negation in Telugu.

Limitations and Future Work

The benchmark covers only Telugu and does not cover other Indian languages, limiting its generalizability. Human evaluation, while necessary for translation and polarity flipping tasks, is inherently subjective and may introduce annotator bias despite the provided guidelines. Finally, the benchmark evaluates models in a zero-shot setting, and performance may vary with few-shot prompting or fine-tuning on Telugu-specific data. Although our evaluation reveals that models are insensitive to negation in Telugu, the lack of sufficient training data prevented us from fine-tuning the models to address this limitation.

As future work, we plan to extend NEGTEG to other Indian languages, especially Dravidian languages such as Tamil, Kannada, and Malayalam. Since these languages are agglutinative and morphologically rich like Telugu, this will help us evaluate whether the trends observed in Telugu hold across similar languages.

Acknowledgements

We sincerely thank Deepika Ketha (M.Sc. Computer Science) and T V N Prasanthi (B.Tech), both native Telugu speakers, for their valuable contributions to this work. They have extensive experience in creating linguistic benchmark datasets and working as translators, which was invaluable to the annotation and evaluation process. Their efforts in building the benchmark dataset, cross-verifying annotations, and evaluating model outputs were essential to ensuring the quality and reliability of NEGTEG. We also thank Nagaraju Vuppala for his support with the morphological analysis tool.

Ethical Considerations

We release NEGTEG publicly to support further research on negation in Telugu. The benchmark was built using a books dataset sourced from Kaggle, and we do not intend to republish the original data. It is used solely for research purposes.

References

Ahmed Suliman Abuhammad and Mahmoud Ali Ahmed. 2024. Negation detection techniques in sentiment analysis: A survey. *Iraqi Journal of Science*, pages 1060–1069.

Sahar Abdulsalam Alshargabi, Dina Fahmi Kamil, and Ali Hussein Hazem. 2022. A linguistic study of english double negation and its realization in arabic. *Studies in English Language and Education*, 9(3):1148–1169.

Miriam Anschütz, Diego Miguel Lozano, and Georg Groh. 2023. This is not correct! negation-aware evaluation of language generation systems. *arXiv preprint arXiv:2307.13989*.

Gunjan Bhattarai and Katrin Erk. 2024. To learn or not to learn: Replaced token detection for learning the meaning of negation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16237–16250.

Jiangjie Chen, Wei Shi, Ziquan Fu, Sijie Cheng, Lei Li, and Yanghua Xiao. 2023. Say what you mean! large language models speak too positively about negative commonsense knowledge. *arXiv preprint arXiv:2305.05976*.

Noa P Cruz, Maite Taboada, and Ruslan Mitkov. 2016. A machine-learning approach to negation and speculation detection for sentiment analysis. *Journal of the Association for Information Science and Technology*, 67(9):2118–2136.

Priyanka Dasari, Abhijith Chelpuri, Nagaraju Vuppala, Mounika Marreddy, Parameshwari Krishnamurthy, and Radhika Mamidi. 2023. Transformer-based context aware morphological analyzer for telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 25–32.

Mayuri J DILIP and Rajesh KUMAR. 2019. Negative polarity items in telugu. *Acta Linguistica Asiatica*, 9(1):9–28.

Svitlana Romanivna Dobrovolska, Mariana Bohdanivna Opyr, and Svitlana Bohdanivna Panchyshyn. 2024. The category of negation in scientific and technical discourse (translation aspect).

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

Federico Fancellu and Bonnie Webber. 2015. Translating negation: A manual error analysis. In *Proceedings of the Second Workshop on Extra-Propositional Aspects of Meaning in Computational Semantics (ExProM 2015)*, pages 2–11. Association for Computational Linguistics.

Piyush Ghasiya and Kazutoshi Sasahara. 2026. From generation to detection: Leveraging empirically derived linguistic hints for llm-based fake news detection.

Itisha Gupta and Nisheeth Joshi. 2021. A review on negation role in twitter sentiment analysis. *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, 16(4):1–19.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517.

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. 2020. It's not a non-issue: negation as a source of error in machine translation. *arXiv preprint arXiv:2010.05432*.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1301–1312.

Lutfi Budi Ilmawan, Didik Dwi Prasetya, et al. 2024. Negation handling for sentiment analysis task: approaches and performance analysis. *International Journal of Electrical & Computer Engineering (2088-8708)*, 14(3).

Aditya Khandelwal and Suraj Sawant. 2020. Negbert: a transfer learning approach for negation detection and scope resolution. In *Proceedings of the twelfth language resources and evaluation conference*, pages 5739–5748.

Kartika Makkar, Pardeep Kumar, Monika Poriye, and Shalini Aggarwal. 2024. Improving sentiment analysis using negation scope detection and negation handling. *International Journal of Computing and Digital Systems*, 16(1):239–247.

Shiva Kumar BN Mamatha Mylarappa, Thriveni J Gowda, and Venugopal K Rajuk. 2023. Enhanced negation handling for sentiment analysis on twitter using deep neural networks. *Indonesian Journal of Electrical Engineering and Computer Science*, 32(3):1736–1745.

Maria Chiara Martinis, Chiara Zucco, and Mario Cannataro. 2024. Negation detection in medical texts. In *International Conference on Computational Science*, pages 75–87. Springer.

Vandan Mujadia and Dipti Misra Sharma. 2024. Bhashaverse: translation ecosystem for indian subcontinent languages. *arXiv preprint arXiv:2412.04351*.

Partha Mukherjee, Youakim Badr, Shreyesh Doppalapudi, Satish M Srinivasan, Raghvinder S Sangwan, and Rahul Sharma. 2021. Effect of negation in sentences on sentiment analysis and polarity detection. *Procedia Computer Science*, 185:370–379.

Sandeep Sricharan Mukku and Radhika Mamidi. 2017. Actsa: Annotated corpus for telugu sentiment analysis. In *Proceedings of the first workshop on building linguistically generalizable NLP systems*, pages 54–58.

Izunna Okpala, Guillermo Romera Rodriguez, Andrea Tapia, Shane Halse, and Jess Kropczynski. 2022. A semantic approach to negation detection and word disambiguation with natural language processing. In *Proceedings of the 2022 6th International Conference on Natural Language Processing and Information Retrieval*, pages 36–43.

MohammadHossein Rezaei and Eduardo Blanco. 2024. Paraphrasing in affirmative terms improves negation understanding. *arXiv preprint arXiv:2406.07492*.

Nirja Shah and Jyoti Pareek. 2024. Optimized hindi negation detection using a hybrid rule-based and bert model. In *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, pages 544–550. IEEE.

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

Yeonkyoung So, Gyuseong Lee, Sungmok Jung, Joonhak Lee, JiA Kang, Sangho Kim, and Jaejin Lee. 2025. Thunder-nubench: A benchmark for llms' sentence-level negation understanding. *arXiv preprint arXiv:2506.14397*.

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.

Thinh Truong, Timothy Baldwin, Trevor Cohn, and Karin Verspoor. 2022. Improving negation detection with negation-focused pre-training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4188–4193.

Thinh Hung Truong, Timothy Baldwin, Karin Verspoor, and Trevor Cohn. 2023. Language models are not naysayers: an analysis of language models on negation benchmarks. *arXiv preprint arXiv:2306.08189*.

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

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

Appendix

A. Human Evaluation Guidelines

This appendix documents the guidelines provided to human annotators for evaluation tasks. To ensure consistency and reliability, annotators were briefed on these guidelines before beginning the annotation process.

A.1. Guidelines for Negation Preservation in Translation

Human Evaluation Guidelines for Negation Preservation after Translation
Preserved:
- The negation present in the source sentence is correctly translated.
- The scope of the negated verb or phrase is accurately preserved.

Not Preserved:
- The negation is completely omitted in the translation.
- The negation incorrectly modifies a different word or scope.
- The negation is mistranslated due to incorrect morphological agreement in gender or tense.

A.2. Guidelines for Polarity Flipping

Human Evaluation Guidelines for Polarity Flipping
Polarity Check:
- Is the polarity of the specified target correctly flipped? (Yes / No)

Meaning Preservation Rating:
- 5 – Perfect polarity reversal; meaning matches the reference.
- 4 – Meaning is mostly preserved with minor variation.
- 3 – Minor meaning change compared to the reference.
- 2 – Major change in meaning; polarity not correctly flipped.
- 1 – Completely irrelevant or incorrect generation.

B. Prompts Used for Model Inference

B.1. Negation Detection Prompt

You are a Telugu linguistics expert.
Your task is to analyze the following Telugu sentences and determine if they contain negation.
In Telugu, negation can be explicit (using words like కాదు, లేదు, వద్దు) or morphological negation where morphemes attached to verbs or clauses

(using suffixes like _లేదు, _అకపోతే, _అక, _అకు, _అన్_ఏ_లేదు, _అ_లెండి, and many more) or lexical (అసాధ్యం, అసంభవం, అన్యాయం, నిరాశ)
Rules:
- NEGATED (if explicit negative or negated morpheme is identified)
- NOT_NEGATED (if no explicit grammatical negation is present, regardless of whether the sentence contains lexical negativity.)
Sentence: "{sentence}"
Return ONLY JSON:
{"label":"NEGATED"} or {"label":"NOT_NEGATED"}

B.2. Paraphrase Detection Prompt

Your task is to determine if the sentence1 is a paraphrased or negated version of the Sentence2 in Telugu.
Classify the relationship:
- PARAPHRASED: Sentence1 conveys same meaning as Sentence2, by different words or different sentence structure.
- NEGATED: The Sentence1 contradicts or reverses the meaning of the Sentence2.
- If **any part** of the sentence is negated, polarity-flipped, contradicted, or reversed (even partially), you MUST classify it as **NEGATED**.
Given:
Sentence1: "{sentence1}"
Sentence2: "{sentence2}"
Return ONLY a valid JSON object in this exact format:
{"label":"PARAPHRASED"} or {"label":"NEGATED"}
DO NOT include any explanation, text, or formatting outside the JSON.

B.3. Sentiment Classification Prompt

Analyze the sentiment of the following text and classify it as either POSITIVE or NEGATIVE.
Text: "{text}"
Classification Guidelines:
- POSITIVE: Expresses favorable opinions, satisfaction, happiness, praise, benefit, success, approval, progress, desirable situation, or optimistic views
- NEGATIVE: Expresses unfavorable opinions, dissatisfaction, sadness, criticism, harm, failure, danger, loss, undesirable situation, or pessimistic views
- If both positive and negative aspects appear, choose the dominant one.
Return ONLY a valid JSON object in this exact format:
{"sentiment": "POSITIVE"} or {"sentiment": "NEGATIVE"}
DO NOT include any explanation, text, or formatting outside the JSON.

B.4. Polarity Flipping Prompt

In Telugu, polarity is realized on the verb; flipping a verb's polarity can flip the sentence meaning.
If multiple verbs are present, flipping different verbs yields different interpretations, so the target verb or verbs are explicitly provided.
You are given a Telugu sentence where the specified verb(s) contain negative polarity markers.
Rewrite the sentence by flipping the polarity of ONLY the specified verb(s).
Constraints:
- Flip the polarity of the specified verb(s), making only the minimal inflectional adjustments (tense, aspect, mood, person, number, gender) required for grammatical well-formedness
- If the original sentence contains NPIs (e.g., ఏమీ, మరేమీ, ఎవరూ, ఎక్కడికీ, ఏమాత్రం, ఎప్పటికీ), remove or adjust any that are no longer required after the polarity flip.
Input:
Sentence: "{sentence}"
Target Verb(s): "{verb_to_negate}"
Output:
Return ONLY a JSON object in the format:
{"flipped_sentence":"<sentence>"}

C. Negative Polarity Items

Negative Polarity Items (NPIs) are expressions that are grammatically licensed only in the presence of negation. Table 10 presents examples of NPI sentences from the dataset. Mishandling NPIs in downstream tasks can lead to significant errors. In the polarity flipping task, flipping a sentence containing an NPI requires the model to ensure that the NPI remains valid in the resulting sentence, or omit it if it is no longer licensed. In the translation task, sentences with and without NPIs should be translated while preserving their negative meaning.

D. Translation Error Analysis

To better understand how models fail at preserving negation during translation, we conducted a qualitative error analysis on the model outputs. Table 11 presents representative examples of translation errors, categorized by the type of negation failure observed.

E. Additional Results

Table 12 reports detailed accuracy, precision and recall for the sentiment classification task under negated and non-negated instances.

Sentences with Negative Polarity Items (NPI)
నాకు ఏమీ తెలియదని చెప్పింది. *nāku ēmī teliyādani ceppindi* She said that she does not know anything.
గదిలో పాత సామాను తప్ప మరేమీ లేవు. *gadilō pāta sāmānu tappa marēmī lēvu* There is nothing else in the room except old furniture.
శివరావుకి అభిముఖంగా కూర్చున్న సత్యానంద్ ఎప్పుడూ లేనంత ఎక్కువగా తాగుతున్నాడు. *śivarāvuki abhimukhaṃgā kūrcunna satyānaṃd eppuḍū lēnanta ekkuvagā tāgutunnāḍu* Satyanand is drinking more than ever before.
ఆ రొట్టె సరిగా కాలకట్టుంది. అందుకే ఆ రొట్టెను ఎవరూ తినలేదు. *ā roṭṭe sarigā kālakaṭṭuṃdi. aṃdukē ā roṭṭenu evarū tinalēdu* That bread was not properly baked. That is why nobody ate that bread.
నువ్వు ఏం చెప్పొద్దు. *nuvvu ēṃ ceppoddu* You should not say anything.
నేను ఈ విషయం ఎవరికీ చెప్పనుండు. *nēnu ī viṣayaṃ evarikī ceppanuṃḍu* I will not tell this matter to anyone.
ఆకరికి ఎలాంటి సంబోధనా లేకుండానే మొదలెట్టింది. *ākariki elāṃṭi saṃbodhanā lēkuṃḍānē modalleṭṭindi* In the end, she started without any form of address at all.
ఇంతవరకూ ఎవరూ ఎవరికీ ఇవ్వనటువంటి పార్టీ! *iṃtavarakū evarū evarikī ivvanaṭuvaṃṭi pārṭī* A party the likes of which nobody has ever given to anyone!
ఇద్దరూ అలాంటి వాళ్లే అయితే ఇక చెప్పనే అక్కర్లేదు. *iddaru alāṃṭi vāḷlē ayite ika ceppanē akkarlēdu* If both are like that, there is no need to say anything at all.
ఇంత ప్రొద్దున్నే లేవకనా, ఇంటిలో పనంతా చక్కబెట్టింది? *iṃta proddunnē lēvakanā, iṃṭilō panaṃtā cakkapeṭṭindi* Did she finish all the housework without even waking up this early?
అతని మాట ఇంకా పూర్తి కానేలేదు. కన్నారావు కెవ్వుమని అరిచాడు. *atani māṭa iṃkā pūrti kānēlēdu. kannārāvu kevvumani aricāḍu* His words were not even complete. Kannarao screamed.
దేనిమీదా మనస్సు లగ్నం చేయలేక చాలా ఇబ్బంది పడుతుంది. *dēnimīdā manassu lagnaṃ cēyalēka cālā ibbandi paḍutundi* Unable to focus the mind on anything, one is in great difficulty.
తనో చిల్లికానీ పెట్టకుండా ఈ రంగంలోనికి రావాలి. *tanō cillikānī peṭṭakuṃḍā ī raṃgaṃlōniki rāvāli* One must enter this field without spending a single penny.
అడ్రసు కూడా ఇవ్వనంత రహస్యమేముంది దీంట్లో? *aḍrasu kūḍā ivvananta rahasyamēmuṃdi dīṃṭlō* What is so secret that you cannot even give the address?
ఇహ ఇక్కడ ఒక్కక్షణమైనా ఉండకూడదు. *iha ikkaḍa okkakṣaṇamainā uṃḍakūḍadu* One should not stay here even for a single moment.
మళ్ళీ ఓసారి ఒక్కపైసా కూడా ఇవ్వద్దు భరణీ! *maḷḷī ōsāri okkapaisā kūḍā ivvaddu bharaṇī* Do not give even a single penny again, Bharani!

Table 10: NPI sentences in the dataset with NPI words highlighted in blue

Setence	Model Tranlsations	Error
అర్థం కానట్టు అయోమయంగా మళ్ళీ చదివింది. ardhaṃ kānaṭṭu ayōmayaṃgā maḷḷī cadividhi. Confused, she read again as if she didn't understand.	**Bhashaverse:** You haven't read it again. **Indic-gemma-7b:** She read it again to make sure it made sense.	Negation shifted; omitted
బాగా చూస్తేతప్ప అక్కడ చిలుక వున్నట్లు తెలియదు. bāgā cūstētappa akkaḍa ciluka vunnaṭlu teliyadu. Unless you don't observe keenly, you అమ్మ కోపంగా పొద్దస్తమానం టీవీ చూడకని తిట్టింది. amma kōpaṃgā poddastamānaṃ ṭīvī cūḍakani tiṭṭindi. Mom scolded me angrily and told me not to watch TV all the time.	**Llama-3.3-70b:** Mother scolded for watching TV in the morning. **GPT-OSS-20b:** Mom scolded for not watching TV in the morning.	Negation omitted; Mistranslated
ఎక్కువ నీళ్లు తాగకపోతేగూడా, ఇలా వేడి చేస్తుంది. ekkuva nīḷlu tāgakapōtēgūḍā, ilā vēḍi cēstundi. Even if you don't drink much water, it makes you feel this hot.	**Bhashaverse:** Even if you drink a lot of water, it heats up.	Negation omitted
వారే లేనప్పుడు, నేనూ ఉండి ఉండను. vārē lēnappuḍu, nēnū uṇḍi uṇḍanu. When they are not there, I would not be there either.	**Meta-Llama-3-8B:** When they are not there, I will be. **Qwen3-8B:** I will remain here when they are not present.	One negation omitted
వాడు రెండుకంటే ఎక్కువ ఇవ్వడట! vāḍu reṇḍukaṇṭē ekkuva ivvaḍaṭa! He does not give more than two!	**Most of the models:** He gives more than two!	Negation omitted
అనుకున్న పని కాకపోతే మిగిలేది అలాంటి పనికిమాలిన నవ్వే. anukuna pani kākapōtē migilēdi alāṇṭi panikimālin navvē. If the expected work does not happen, what remains is only that useless smile.	**Model:** If the work expected is not done, it's not funny.	Extra negation added
వయసు మీదపడడం వలననో, లేక అనేకమంది శిష్యులలో గుర్తుపట్టకనో అయన మొదట గుర్తుపట్టలేదు. vayasu mīdapaḍaḍaṃ valanano, lēka anēkamandi śiṣyulalō gurtu paṭṭakano ayana modaṭa gurtuppaṭṭalēdu. At first he could not recognize, either because of his advancing age or he has too many disciples.	**Google Translate:** He was not recognized at first, either because of his advancing age or because many of his disciples did not recognize him.	Negation on wrong subject
అందువలన జోగయ్యని చూసీ చూడనట్లు వదిలేసేవాడు. *anduvalna jōgayyani cūsī cūḍanaṭlu vaḍilēsēvāḍu.* Hence he pretended as if he didn't see Jogayya.	**GPT-OSS-20b:** Therefore, Jogayyani would leave as if he had not seen.	Negation on wrong person

Table 11: Examples of negation translation errors made by models, categorized by error type.

	Accuracy		Precision_pos		Recall_pos		Precision_neg		Recall_neg	
Model	NEG	NOT	NEG	NOT	NEG	NOT	NEG	NOT	NEG	NOT
DeepSeek-R1-Qwen3-8B	0.767	0.720	0.923	0.985	0.257	0.536	0.752	0.594	0.991	0.988
Indic-gemma-7b-Navarasa	0.778	0.848	0.618	0.861	0.720	0.883	0.867	0.828	0.804	0.798
Meta-Llama-3-8B-Instruct	0.797	0.857	0.665	0.891	0.695	0.865	0.861	0.811	0.843	0.845
Nemotron-Cascade-8B	0.798	0.744	0.765	0.944	0.502	0.605	0.807	0.621	0.931	0.948
Qwen3-8B	0.833	0.840	0.848	0.962	0.560	0.760	0.829	0.731	0.955	0.957
Gemma-7b-it	0.739	0.758	0.584	0.871	0.537	0.695	0.800	0.656	0.829	0.849
GPT-OSS 120b	0.841	0.870	0.789	0.954	0.662	0.820	0.859	0.782	0.921	0.943
GPT-OSS 20b	0.851	0.873	0.804	0.958	0.685	0.822	0.868	0.785	0.925	0.947
Llama-3.3-70b-versatile	0.867	0.884	0.837	0.970	0.707	0.830	0.878	0.795	0.938	0.962
Gemini-2.5-Flash	0.859	0.890	0.829	0.972	0.685	0.839	0.869	0.804	0.937	0.965

Table 12: Sentiment classification performance under negated (NEG) and non-negated (NOT) sentence conditions.Precision_pos and Recall_pos refer to the positive sentiment class, while Precision_neg and Recall_neg refer to the negative sentiment class.