Lightweight Adaptation of BERT for Negation Sentiment Classification

TAO HONGMO[1] [0009-0002-3639-1499]
Graduate School of Science and Technology
Meiji University
Kawasaki, Japan
ce251024@meiji.ac.jp

Hiroyuki Kamata[2] [0009-0004-0598-4210]
School of Science and Technology
Meiji University
Kawasaki, Japan
kamata@meiji.ac.jp

Abstract— **Pre-trained language models (PLMs) achieve high accuracy on standard benchmarks for sentiment analysis. However, this performance can hide systematic weaknesses in determining the sentiment of negated sentences, for example when the phrase "not good" is still classified as positive.**
In this study, we use sentiment classification of English movie reviews in the Stanford Sentiment Treebank 2 (SST-2) as a case study to specifically examine and improve how BERT handles negated sentences.
We perform a brief additional fine-tuning of the existing BERT model on a small, automatically constructed set of lexicon-based counterfactual examples that target simple lexical negation. Experimental results on carefully paired original–negated sentences show that this procedure substantially reduces prediction errors on negated inputs while leaving overall performance on SST-2 almost unchanged.

Keywords— **BERT, sentiment analysis, linguistic negation.**

I. INTRODUCTION

Pre-trained language models (PLMs) such as BERT have achieved strong results on natural language understanding benchmarks. For example, they perform very well on the General Language Understanding Evaluation (GLUE) benchmark and on sentiment classification in the Stanford Sentiment Treebank 2 (SST-2). Building on this progress, many large language models (LLMs) have recently been released. However, evaluations on out-of-distribution (OOD) tasks reveal a different picture. In these tests, high accuracy on standard test sets does not guarantee reliable behavior on out-of-distribution examples. Among these phenomena, negation is one of the most frequent and impactful factors [1][2]. In sentiment analysis, for example, the sentences "This movie is good." and "This movie is not good." differ by only one word, but they express opposite sentiment polarity. If a model fails to take the word "not" into account, it will make incorrect decisions in applications such as review filtering.

A series of studies [3] have shown that BERT models often rely on surface keywords, and as a result, they may fail to understand the full meaning of a sentence when they encounter negation relations. Ettinger et al. [4] report that BERT is notably insensitive to the contextual effect of negation cues. Ribeiro et al. [5] propose a testing framework that makes minimal edits to original instances in tasks such as sentiment analysis. Their results show that error rates on negation-specific test templates are much higher than their performance on standard validation sets would suggest.

To address such issues, one important approach is to construct counterfactually augmented data (CAD) [6]. In a standard setup, human annotators make minimal edits to an original sentence so that the true label flips, while keeping the rest of the sentence unchanged. Kaushik et al. show that such human-written counterfactuals can improve robustness on intentionally constructed test sets that reverse the original semantics [6]. In addition, recent studies [7] have proposed methods that specifically target negation, such as unsupervised negation modules based on more sophisticated mathematical modeling, which aim to improve overall sentiment-analysis performance. However, these methods usually require additional model components and relatively heavy feature engineering [8].

In this study, we use the SST-2 English movie-review sentiment classification dataset [9] as our experimental setting. We examine how BERT behaves when encountering sentiment expressions that include negation cues such as "not." Our analysis centers on the following two questions.

i. First, if we select sentences from SST-2 that are short, structurally simple, and exhibit clear sentiment polarity, and then construct a negated counterpart for each sentence, does a fine-tuned BERT model correctly adjust its predicted sentiment under negation?
ii. Second, if it does not, can we more effectively improve this behavior by adding only a small set of negated examples generated through simple text-replacement rules, without greatly costing additional training time or altering the model architecture?

To answer these questions, we evaluate BERT on carefully constructed pairs of original and negated sentences. In addition to reporting accuracy on original and negated inputs separately (OrigAcc and NegAcc), we use two pairwise consistency metrics "BothCorrect" and "PredOpposite" to quantify whether the model (i) predicts both members of a pair correctly and (ii) assigns opposite sentiment polarity to each pair.

Overall, this study makes three contributions. First, we design a simple lexicon-based procedure for selecting naturally occurring SST-2 sentences whose overall polarity is largely determined by a single sentiment adjective and for automatically constructing their negated counterparts, which exposes a substantial gap between a standard BERT model's performance on original and negated inputs. Second, we apply a pairwise evaluation protocol with OrigAcc, NegAcc, BothCorrect, and PredOpposite to explicitly measure not only accuracy but also polarity-reversal consistency across original–negated sentence pairs. Third, we show that adding only about 3% automatically generated negated examples and fine-tuning for one additional epoch is sufficient to almost completely close this gap in negation robustness, while leaving overall validation accuracy on SST-2 largely unchanged.

II. Methodology

1) *Baseline Model*

We use the **BERT-base-uncased** checkpoint from the HuggingFace Transformers library as our baseline model [10]. The architecture is as follows.

- Encoder: A 12-layer Transformer with hidden size 768, identical to the standard BERT-base configuration.
- The standard cross-entropy loss using sentence-level labels.

We use the Stanford Sentiment Treebank 2 (SST-2) dataset as our training data. SST-2 is part of the General Language Understanding Evaluation (GLUE) [11] benchmark. We fine-tune BERT for 2 epochs on the SST-2 training set with the following hyperparameters:

- Batch size: 16.
- Maximum sequence length: 128 tokens.
- Optimizer: AdamW with learning rate 2×10^{-5} and weight decay 0.01.
- Learning-rate schedule: linear decay with the first 10% of training steps used for warmup.
- After training, the baseline model reaches about 92.89% accuracy on the development set. We call this model **BERT-baseline**.

2) *Selection Rules*

In order to analyze the effect of negation on sentiment, we generate negated sentences on a subset of SST-2. We select simple sentences from the training set that include only one sentiment word, which determines the overall sentiment. Specifically, we select only sentences that satisfy the following conditions.

a) Length constraint:
We tokenize each sentence into English words and retain only those with no more than 12 tokens. This is because short sentences are easier to negate.

b) Simple surface structure:
We discard sentences that contain words such as "*but*", "*although*", "*however*", or "*though*". This is to avoid errors caused by complex discourse or contrastive cues.

c) Sentiment-word constraint:
We manually construct a sentiment lexicon.

- The **positive lexicon (POS)** is:
  {good, great, excellent, amazing, awesome, wonderful, fantastic, nice, enjoyable, love, loved, lovely}
- The **negative lexicon (NEG)** is:
  {bad, terrible, awful, horrible, disappointing, boring, worse, worst, hate, hated}

For each sentence, we check whether it contains words from **POS** or **NEG**. A sentence is selected only if it satisfies:

- if the sentence label is positive (y = 1), it contains **exactly one POS word and no NEG word.**
- if the sentence label is negative (y = 0), it contains **exactly one NEG word and no POS word.**

As a result, the selected sentences only take forms such as "*It was good*", "*a wonderful film*", or "*a bad movie*".

d) No existing negation
Because we will later insert "*not*" before the sentiment word, we exclude sentences that already contain "*not*" or "*isn't*", such as "*not not good*" to avoid double negation.

Following these rules, we extract a set of "simple sentences" from the SST-2 training set. We split them into simple positive and simple negative subsets according to the original labels, and denote their index sets by $\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{neg}}$, respectively. These explicit selection rules are designed to yield a controlled subset of sentences in which the overall sentiment is largely anchored by a single lexical item, making them particularly suitable for isolating the effect of lexical negation.

3) *Construction Rules*

After selecting the set of simple sentiment sentences, we generate a negated version corresponding to each sentence. These negated sentences are later used for testing and fine-tuning the model. For each selected sentence, we first identify the unique sentiment word from either the **POS** or the **NEG** lexicon and then locate it in the sentence. We then construct its negated version x' using the following rules:

i. Replace w with "*not*"+ w

(for example, "*good*" → "*not good*", "*enjoyable*" → "*not enjoyable*")

ii. Flip the label so that positive examples become negative and negative examples become positive.

This rule has certain limitations. For instance, "*not good*" is often closer to neutral than to strongly negative in real usage. However, under the binary sentiment classification setup of SST-2, this serves as a useful heuristic. Moreover, our selection step has removed long sentences and cases with multiple sentiment words. This increases the probability that the single sentiment word determines the overall sentence polarity. Taken together, the selection and construction rules define a highly controlled instantiation of CAD for lexical negation on naturally occurring SST-2 sentences.

4) *Negation Test Set and Consistency Metrics*

For evaluation, we generate a dedicated negation test set. To construct this set, we sample from the simple sentiment sentences obtained in the previous step as follows:

- 200 positive sentences and their 200 negated versions (rewritten from positive to negative)
- 200 negative sentences and their 200 negated versions (rewritten from negative to positive)

In total, we obtain $N = 400$ sentence pairs $\{(x_i, y_i\ x_i', y_i')\}_{i=1}^{N}$, where x_i is the original sentence and x_i' is the negated sentence, with labels y_i and y_i', respectively.

When building the test set, we store the indices of all 400 sampled sentences in the original training data. In the later augmentation-based training, we exclude these indices to ensure that none of the test-set sentences are used as training examples. On this test set, we adopt a simple pairwise evaluation protocol with the following metrics:

a) Original accuracy (OrigAcc):

The accuracy when the model is evaluated on all original sentences in the test set.

b) Negated accuracy (NegAcc):

The accuracy when the model is evaluated only on the negated sentences.

c) Both-correct rate (BothCorrect):

A sentence pair (x_i, x_i') is counted as correct only if the model predicts both the original sentence x_i and its negated counterpart x_i' correctly.

d) Opposite-prediction rate (PredOpposite):

The proportion of sentence pairs (x_i, x_i') for which the model's predicted labels for x_i and x_i' are different. **Even if both predictions are wrong with respect to the true labels**, we still count the pair as successful under this metric as long as the two predicted labels differ. In practice, we focus more on **BothCorrect**, because opposite predictions alone do not guarantee that the predictions are correct. This test set, together with the pairwise protocol, provides a convenient diagnostic probe for negation robustness that can be reused with other sentiment models trained on SST-2.

5) Constructing Negation-Augmented Training Examples

In the previous section, we constructed a negation test set used solely for evaluation. Here, we apply the same negation construction to the training dataset to obtain additional training examples.

We start from the previously selected simple positive and negative sentences. Formally, let $\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{neg}}$ denote their indices in the training data. We remove all indices that have been used for the negation test set and obtain the remaining index sets $\tilde{\mathcal{L}}_{\text{pos}}$ and $\tilde{\mathcal{L}}_{\text{neg}}$.

From these sets, we select sentences for augmentation as follows:

- from $\tilde{\mathcal{L}}_{\text{pos}}$, we select at most 1000 simple positive sentences;
- from $\tilde{\mathcal{L}}_{\text{neg}}$, we select at most 1000 simple negative sentences.

For these selected sentences, we first apply exactly the same negation rule as before by inserting "*not*" before the sentiment word and flipping the label. We then discard the original sentences and use only the negated sentences as training examples.

In total, we generate 1,765 negation-augmented training instances, which corresponds to roughly 3% of the original training set (67,349 instances). We denote this augmented dataset by $\mathcal{D}_{\text{neg-aug}}$. We then combine it with the original training set $\mathcal{D}_{\text{train}}$ to obtain the augmented training data

$$\mathcal{D}_{\text{train}}^{\text{aug}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{neg-aug}}$$

After augmentation, the training set size is about 69,114 instances.

6) Incremental Fine-tuning Based on Baseline Weights

We do not retrain BERT from scratch. Instead, we adopt a two-stage procedure.

a) Training the baseline model (BERT-baseline)

We train BERT for 2 epochs on the original training set $\mathcal{D}_{\text{train}}$, following the same training setup as in the baseline model.

b) Fine-tuning with augmented data

We fine-tune the baseline model for one additional epoch on the augmented training set $\mathcal{D}_{\text{train}}^{\text{aug}}$. We then train for one additional epoch, while keeping the same optimizer and hyperparameters as in the baseline stage—AdamW with a learning rate of 2×10^{-5}, weight decay 0.01, and a linear warmup-and-decay learning-rate schedule. This second stage requires only about 4320 additional training steps, and its training time is comparable to that of a single baseline epoch.

After this fine-tuning stage on top of the baseline model, we refer to the resulting model as **BERT-aug-neg**. In the following experiments, we compare **BERT-baseline** and **BERT-aug-neg** on both the SST-2 development set and our negation test set.

III. Experimental Results

Table I shows the validation accuracies of the baseline model (BERT-baseline) and the negation-augmented model (BERT-aug-neg) on the SST-2 development set.

TABLE I. ACCURACY ON THE FULL DEVELOPMENT SET

Model	**Training data**	**Training epochs**	**Accuracy (val_acc)**
BERT-baseline	Original training set (67,349)	2 epochs	0.9289
BERT-aug-neg	Augmented training set (69,114)	baseline + 1 epoch with augmentation	0.9174

When trained on the original SST-2 training set, BERT-baseline achieves an accuracy of 92.89% on the development set, which falls within the typical range of 91-93% reported for BERT on SST-2 in public implementations.

After adding about 1,765 negation-augmented training examples and fine-tuning the baseline model for one additional epoch, the validation accuracy of the augmented model **BERT-aug-neg** slightly decreases to 91.74%. This suggests that while our negation-based augmentation introduces some perturbation, it does not result in a noticeable drop in overall performance.

We previously constructed a negation test set containing 400 pairs of original sentences and their corresponding negated versions. Table II reports the detailed results of **BERT-baseline** and **BERT-aug-neg**.

TABLE II. PERFORMANCE COMPARISON ON THE NEGATED-SENTENCE TEST SET

Model	OrigAcc	NegAcc	BothCorrect	PredOpposite
BERT-baseline	0.9975	0.7600	0.7575	0.7575
BERT-aug-neg	0.9975	0.9975	0.9950	0.9950

From the table we observe:

Performance on original sentences(OrigAcc):

- ◆ Both models achieve an accuracy of 0.9975 on original sentences. This indicates that negation-focused augmentation does not degrade the model's ability on the original inputs.

Performance on negated sentences(NegAcc):

- ◆ On negated sentences, the baseline model achieves an accuracy of 0.7600.
- ◆ The augmented model improves negated-sentence accuracy to 0.9975. This shows that **BERT-baseline** exhibits substantial bias when handling structures such as "*not good*", "*not enjoyable*", and "*not love*". After fine-tuning with our generated negated samples, the model corrects almost all of these errors, with 399 out of 400 negated sentences classified correctly.

Original–negated pairwise correctness(BothCorrect):

- ◆ The baseline model's BothCorrect is 0.7575.
- ◆ The augmented model's BothCorrect rises to 0.9950. This demonstrates that our augmentation improves the model's understanding of polarity reversal.

Proportion of opposite predictions(PredOpposite):

- ◆ The baseline model's PredOpposite is 0.7575.
- ◆ The augmented model's PredOpposite reaches 0.9950, which is numerically identical to its BothCorrect score. The augmented model not only produces opposite predictions for each pair, but also yields predictions that are almost always correct.

Overall, **BERT-aug-neg** substantially enhances robustness to negation. This requires only one additional epoch of fine-tuning on top of the baseline and uses approximately 3% augmented samples.

IV. ANALYSIS OF RESULTS

To further understand how the augmented model's behavior changes under negation, we analyze the classification results and focus on three types of errors:

a) **CASE_FIXED**: for a given negated sentence, the prediction is incorrect for the baseline model but correct for the augmented model.

b) **CASE_REGRESSED**: for a given negated sentence, the prediction is correct for the baseline model but incorrect for the augmented model.

c) **CASE_BOTH_WRONG**: for a given negated sentence, the prediction is incorrect for both models.

On the negation test set comprising 400 negated sentences, the baseline model attains 304 correct predictions whereas the augmented model attains 399; the counts for **CASE_FIXED**, **CASE_REGRESSED**, and **CASE_BOTH_WRONG** are 95, 0, and 1, respectively, indicating that the augmentation training stage introduces virtually no new errors on the negated sentences. Below are representative examples (true labels in parentheses: 1 = positive, 0 = negative).

i. **Original sentence**:

" *if good-hearted* " (true = 1)

Negated sentence:

" *if not good-hearted* " (true = 0)

Baseline prediction results:

original sentence = 1; negated sentence = 1

Augmented model prediction results:

original sentence = 1; negated sentence = 0

ii. **Original sentence**:

"*occasionally enjoyable*" (true = 1)

Negated sentence:

"*occasionally not enjoyable*" (true = 0)

Baseline prediction results:

original sentence = 1; negated sentence = 1

Augmented model prediction results:

original sentence = 1; negated sentence = 0.

The following example belongs to **CASE_BOTH_WRONG**, where both models predict the negated sentence incorrectly:

iii. **Original sentence**:

"*the people who loved the 1989 paradiso will prefer this new version*" (true = 1)

Negated sentence:

"*the people who not loved the 1989 paradiso will prefer this new version*" (true = 0)

Baseline prediction results:

original sentence = 1; negated sentence = 1

Augmented model prediction results:

original sentence = 1; negated sentence = 1

This case reveals two types of difficulty:

(1) **Grammatical aspect:** the rule-based rewrite "*the people who not loved ...*" is unnatural in English; more idiomatic forms would be *"who did not love*" or "*who do not love*", and this grammatical awkwardness may affect the model's prediction.

(2) **Semantic aspect**: even if we use "*did not love*", the sentence still emphasizes "*will prefer this new version*" so the overall sentiment is still likely to be interpreted as positive. Therefore, simply flipping the label when generating such counterfactual examples is not always appropriate.

This example indicates that counterfactual data augmentation (CAD), although simple and time-efficient, can introduce semantic noise, and in such cases the label should not be flipped in a purely mechanical way. More generally, it illustrates that even carefully designed rule-based CAD remains vulnerable to cases where the overall discourse sentiment is not fully controlled by the locally negated word.

V. Conclusion and Future Work

In this work, we investigated the robustness of a BERT model to negation in the SST-2 sentiment classification task. Our experiments show that, although a standard fine-tuned BERT model achieves an accuracy of 0.9975 on the original sentences, its accuracy on the corresponding negated sentences is only 0.7600. To address this issue, we proposed a data augmentation method that is entirely based on simple text-replacement rules and requires no additional human annotation. After one epoch of fine-tuning on top of the baseline model with the augmented data, the proportion of sentence pairs for which both the original and the negated sentence are predicted correctly increases from 0.7575 to 0.9950. These results indicate that the robustness of the model to negation can be substantially improved without modifying the model architecture. This suggests that the observed weakness is not due to an inherent inability of BERT to represent negation, but rather to a lack of appropriate contrastive supervision for simple lexical negation patterns during fine-tuning.

Although our method significantly improves robustness to negation, it still has limitations. Even with strict filtering, inserting "*not*" before a single sentiment word can produce grammatically awkward sentences. Therefore, it remains an open question whether this approach is equally effective for more complex forms of negation, which requires further experimentation and discussion; extending our framework to such cases is part of our planned future work. A natural next step is to combine our lightweight, task-specific augmentation strategy with more sophisticated syntactic or semantic negation handling methods, in order to cover a broader range of negation phenomena while explicitly controlling label noise.

Acknowledgments

This research was partially supported by the Joint Research Fund from KDS Co., Ltd and ANIMO Co., Ltd.

References

[1] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo, "A survey on the role of negation in sentiment analysis," in *Proc. Workshop Negation Speculation Nat. Lang. Process*., Uppsala, Sweden, Jul. 2010, pp. 60–68.

[2] J. Barnes, L. Øvrelid, and E. Velldal, "Sentiment analysis is not solved! Assessing and probing sentiment classification," in *Proc. 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy, Aug. 2019, pp. 12–23.

[3] N. Kassner and H. Schütze, "Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Online, Jul. 2020, pp. 7811–7818.

[4] A. Ettinger, "What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 34–48, 2020.

[5] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh, "Beyond accuracy: Behavioral testing of NLP models with CheckList," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Online, Jul. 2020, pp. 4902–4912.

[6] D. Kaushik, E. H. Hovy, and Z. C. Lipton, "Learning the difference that makes a difference with counterfactually-augmented data," *arXiv preprint* arXiv:1909.12434, 2019.

[7] N. Punetha and G. Jain, "Optimizing sentiment analysis: A cognitive approach with negation handling via mathematical modelling," *Cogn. Comput*., vol. 16, no. 2, pp. 624–640, Nov. 2023.

[8] M. M. Hossain and E. Blanco, "Leveraging affirmative interpretations from negation improves natural language understanding," in *Proc. 2022 Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Abu Dhabi, United Arab Emirates, Dec. 2022, pp. 5833–5847.

[9] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in Proc. *2013 Conf. Empirical Methods Nat. Lang. Process. (EMNLP)*, Seattle, WA, USA, Oct. 2013, pp. 1631–1642.

[10] T. Wolf *et al*., "HuggingFace's transformers: State-of-the-art natural language processing," *arXiv preprint* arXiv:1910.03771, 2019.

[11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Brussels, Belgium, Nov. 2018, pp. 353–355.