Leveraging Sentiment Adjectives in Instruction Tuning of LLMs for Zero-Shot Sentiment Classification

Yang Zhao[1] Masayasu Muraoka[1] Issei Yoshida[1]
Bishwaranjan Bhattacharjee[2] Hiroshi Kanayama[1]
[1]IBM Research - Tokyo, 19-21 Nihonbashi Hakozaki-cho, Chuo City, Tokyo, 103-8510, Japan
[2]IBM Research, Yorktown Heights, New York 10598, USA
yangzhao@ibm.com {mmuraoka,issei,hkana}@jp.ibm.com bhatta@us.ibm.com

Abstract

Instruction tuning significantly improves the performance of LLMs in tasks such as sentiment classification. In this work, we propose a simple yet efficient instruction augmentation method which does not rely on any actual labeled sentiment instances. With just 240 pseudo-instruction instances, the proposed method significantly improves the sentiment classification performance across several LLMs on 12 sentiment benchmark datasets, increasing scores by 30 points and outperforming LLMs that utilize more complex instruction tuning methods by 5.1 points.

1 Introduction

Sentiment analysis has long been an established area of research in Natural Language Processing (NLP). With recent advances in large language models (LLMs), impressive zero-shot performance in sentiment analysis was achieved by instruction-tuned LLMs [5, 18, 14]. A typical sentiment **Instruction Instance** is a tuple with three components *(T, I, O)*:

- **Instruction Text (T)**: *Classify the following sentence into either positive, neutral or negative sentiment.*
- **Input (I)**: *A movie journey worth taking.*
- **Output (O)**: *The sentiment is positive.*

where the instruction text (**T**) refers to the user instruction. It usually specifies the desired outputs; the input (**I**) refers to the input sentence or document for the sentiment task; the output (**O**) refers to the ground truth answer corresponding to the instruction text.

Previously, many sentiment analysis studies have utilized actual training instances in sentiment benchmark datasets as Input (**I**) and corresponding labels as Output (**O**) for instruction tuning. For example, the work [21] instruction-tuned LLMs across various NLP tasks, including four sentiment datasets, while the work [3] further expanded this approach to more than 1,800 NLP tasks. Considering sentiment classification spans diverse domains such as finance, restaurants, movies, and politics, obtaining a large number of domain-specific labeled instances for instruction tuning is labor-intensive and inefficient.

To enhance this aspect, we propose a simple-yet-efficient instruction augmentation method to construct sentimental adjective-based pseudo instructions that do not rely on any training instances in sentiment benchmark datasets. Subsequently, we instruction-tune Llama2-7b,13b,70b base models and the Falcon-40b base model and evaluate their zero-shot performance on 12 sentiment benchmark datasets. The results show that instruction-tuned models significantly outperform the base models by 30 points and other instruction-tuned models by an average of 5.1 points.

2 Sentimental Adjective-based Instruction Construction

We herein describe the steps to construct pseudo instances using sentimental adjectives. Section 2.1 outlines the process for collecting diverse sentiment instruction text (**T**) from various corpora. Section 2.2 details the steps of constructing instruction using sentimental adjectives for Input (**I**) and Output (**O**).

2.1 Instruction Text (T) Collection

User instructions exhibit a wide variety of paraphrasing. To increase the diversity, we collect sentiment in-

struction text (**T**) from five widely-used instruction datasets written by either human annotators or LLMs, as follows: (1) SuperNI [20], which contains 96k instructions written by humans covering 1600+ NLP tasks. (2) Alpaca[1] [16], which contains 52k instructions generated by GPT-3 (davinci-003). (3) Self-instruct [19], which contains 82k instructions generated by GPT-3 (vanilla). (4) Unnatural Instructions [8], which contains 68k instructions generated by GPT-3 (davinci-002). (5) Baize [22], which contains 210k instruction instances created by prompting ChatGPT and letting it converse with itself. We extracted all the instruction text (**T**) from these datasets and retained instruction texts only if they contain the terms 'sentiment', 'positive', 'negative', and 'neutral'. Finally, 110 diverse sentiment instruction text (**T**) are yielded, and we empirically determine to use 80 for training and 30 for testing during instruction tuning. For the aspect-based sentiment classification task, we add *with respect to the TARGET* to the instruction text and replace TARGET with the specific aspect.

2.2 Sentimental Adjective (I, O) Pair

Inspired by the concept of evaluative adjectives in linguistics, we describe the four steps to automatically collect pairs of instruction input (**I**) and output (**O**). Evaluative adjectives often express value judgments and convey opinions, emotions, or subjective interpretations. For instance, adjectives like *beautiful* imply a positive sentiment, while *awful* suggests a negative one. We refer to our collected adjectives as *sentimental adjectives*.

Step 1. Collect sentimental adjective candidates
We start by collecting adjectives from SentiWordNet 3.0[2] [2] where each sense of an adjective word w is assigned two scores: a positive score (S_{pos}) and a negative score (S_{neg}) where $0 \leq S_k \leq 1$ and $k \in \{pos, neg\}$. The selection criteria is:

1. Choose all words where at least one of its senses meets the criteria: $S_{pos} \geq r$ and $S_{neg} = 0.0$ to compile positive word list L^1_{pos}
2. Choose all words where at least one of its senses meets the criteria:: $S_{neg} \geq r$ and $S_{pos} = 0.0$ to compile negative word list L^1_{neg}
3. Choose all words where at least one of its senses meets the criteria: $S_{pos} = 0.0$ and $S_{neg} = 0.0$ to compile neutral word list L^1_{neu}

We empirically determine the threshold r to trade off between the number and quality of adjectives. Please see Table 3 in Appendix for L^1.

Step 2. Align with sentiment word sense.
This step aims to refine the adjective lists in Step 1. For instance, one sense of the word 'fresh' meets the criteria $S_{neg} \geq 0.75$ and $S_{pos} = 0.0$, this word is therefore included in the negative list L^1_{neg}. However, "fresh" often conveys a non-negative meaning, typically referring to something new or unused. including this word in negative list may confuse the model during instruction tuning. To address this, we utilize pre-defined positive (V_{pos}) and negative (V_{neg}) vocabularies in the paper [10]. Words in lists L^1_{pos} and L^1_{neg} are excluded if they do not appear in V_{pos} and V_{neg}, respectively. Words in L^1_{neu} are removed if they appear in either V_{pos} or V_{neg}. This process results in three refined lists: L^2_{pos}, L^2_{neg}, and L^2_{neu}. Please see Table 4 in Appendix for L^2.

Step 3. Rank word by frequency.
This step focuses on selecting more domain-agnostic words by leveraging frequency information. We use English Wikipedia[3] to obtain word frequency for ranking adjectives in each list in descending order based on their frequency. If a adjective in L^2_{pos}, L^2_{neg}, and L^2_{neu} is not in the wiki frequency list, its frequency would be set to zero. After ranking, frequent words such as *best*, *great*, and *important* appear at the top of the positive list, whereas the original words in the list are *legendary*, *solid*, and *gallant*. We note the ranked lists as L^3_{pos}, L^3_{neg}, and L^3_{neu}. Please see Table 5 in Appendix for L^3.

Step 4. Add negation words.
This step helps LLMs to better handle sentences containing negation words. We add the negation word *not* directly before adjectives (e.g., *not beautiful*) for X% of instances in only L^3_{pos} and L^3_{neg}. Subsequently, adjectives with negation from the positive list are transferred to the negative list and vice versa. This process yields the final lists: L^4_{pos},

1) https://github.com/gururise/AlpacaDataCleaned/
2) https://github.com/aesuli/SentiWordNet. It is under CC BY-SA 4.0 license.
3) https://jwsmythe.com/tools/wordlist/wikipedia-word-frequency-master/results/enwiki-2023-04-13.txt

L^4_{neg}, and L^4_{neu} (where $L^4_{neu} = L^3_{neu}$). Please see Table 6 in Appendix for L^4.

After completing steps 1 to 4, we take the first instruction text **T** from 80 instruction texts in Section 2.1, the first adjective from L^4_{pos} and *positive* to form the first tuple (**T**, **I**, **O**); Continue this process until the 80th instruction text is taken. Then, we obtained 80 tuples for the positive class, 80 tuples for the negative class, 80 tuples for the neural class respectively.

3 Experiment

3.1 Experimental Setup

The constructed 240 tuples are split into 80% for the training set and 20% for the development set. We set the threshold r in SentiWordNet 3.0 in Step 1 to 0.75, and negation word percentage X to 10%, according to performance on the development set. For training, we follow the paper [17] by utilizing an auto-regressive objective and zeroing out the loss on tokens from the user prompt, including instruction text and input, while backpropagating only on instruction output. Of the 110 instruction texts, we use 80 for model training and development, and remaining 30 for testing. During training, we employ the efficient parameter tuning technique, LoRA [9], with a LoRA rank of 8 and LoRA alpha of 32. We set learning rate to 2e-4 and batch size to 2. During inference, we follow previous work [6] to load models in the 8-bit mode which significantly speeds up the inference and has negligible impact on the final performance. We set the maximum number of generated tokens to 20. All the experiments are conducted using one A100 GPU.

Evaluation Metric

Since all the instruction texts we collected explicitly specify the output space as positive, negative, or neutral label, we adopt the following metric for calculating instance-wise accuracy: 1) Score 1 if the output string contains the ground-truth label and does not contain other classes' ground-truth labels (case insensitive); 2) Score 0, otherwise. We observed a high correlation score between the human annotator and this automatic metric, So we decided to use this metric for all datasets.

3.2 Dataset

We experiment with 7 general sentiment classification datasets, i.e., SST-2 [15], IMDB, Yelp, Amazon datasets from [11], Airline[4] Debate[5], financial phrasebank [13] as well as 5 aspect-based sentiment classification datasets[6] from the Workshop on Semantic Evaluation (SemEval) in 2014, 2015, and 2016.

Table 1 shows the statistics of each dataset. We paired each sentence from the sentiment benchmark datasets with 30 instruction texts for testing. For instance, in the case of SST-2, this resulted in 1,821 × 30 = 54,630 instances used for testing instruction-tuned models. The same procedure was applied to the other datasets.

Table 1: Statistics of sentiment classification datasets.

Dataset	Domain	Size	# Class	Aspect
SST-2	Movie	1,821	2	no
Yelp	Restaurant	1,000	2	no
Amazon (Amaz)	Product	1,000	2	no
IMDB	Movie	1,000	2	no
Airline	Operation	1,000	3	no
Debate (Deba)	Politics	1,000	3	no
PhraseBank (PB)	Finance	970	3	no
SemEval-14lap	Laptop	543	3	yes
SemEval-14res	Restaurant	994	3	yes
SemEval-15res	Restaurant	485	3	yes
SemEval-15hot	Hotel	215	3	yes
SemEval-16res	Restaurant	514	3	yes

3.3 Models

We instruction-tuned Llama2 base model [17], and falcon-40b base model [1] using our constructed 240 instruction tuples (**T**, **I**, **O**), noted as **base+ours**. In addition, we consider the following comparison methods:

base+ours w/o adjective Previous works, such as [12], have pointed out that some instruction-tuned models do not fully utilize instructions, and that the impressive performance gains from instruction tuning may stem from models learning superficial patterns, such as the output space and format. To verify this, we replaced the sentimental adjectives with empty strings to ablate the input, while keeping the instruction text and output format unchanged.

4) https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment. 1k instances is used only.
5) https://www.kaggle.com/datasets/crowdflower/first-gop-debate-twitter-sentiment. 1k instances is used only.
6) https://github.com/kevinscaria/InstructABSA/tree/main/Dataset

Table 2: Accuracy of zero-shot sentiment classification on 12 benchmark datasets. Best results associated with the same base model are in bold.

Dataset	SST-2	Yelp	Amaz	IMDB	Deba	Airline	PB	14hap	14res	15res	15hot	16res	Ave.
△ Lexicon-match baseline	59.2	64.7	69.6	69.0	55.4	63.6	56.0	68.9	81.5	74.4	74.9	78.4	67.9
#1 llama2-7b-base	49.4	51.2	40.8	46.7	34.7	37.4	27.3	40.0	61.5	53.8	61.5	61.6	47.2
#2 llama2-7b-chat	78.8	88.8	83.5	86.2	61.6	**69.9**	**60.5**	75.5	83.6	78.9	71.5	75.7	76.2
#3 base+ours w/o adjective	38.9	35.5	32.3	37.9	35.2	35.6	29.9	18.3	13.8	24.9	16.4	13.4	27.7
#4 base+ours	**89.5**	**96.1**	**94.1**	**94.4**	**62.0**	67.6	53.5	**82.1**	**88.4**	**86.3**	**84.4**	**89.4**	**82.3**
$1 llama2-13b-base	47.0	52.5	43.4	49.3	36.1	40.8	41.7	46.1	61.2	56.7	58.0	57.9	49.2
$2 llama2-13b-chat	71.2	79.0	75.0	77.9	62.9	69.5	59.1	**68.9**	76.9	71.4	63.1	65.4	70.0
$3 base+ours w/o adjective	49.6	50.4	44.4	50.7	38.7	43.1	26.3	28.0	35.1	41.9	33.9	24.9	38.9
$4 base+ours	**80.5**	**88.4**	**75.9**	**86.1**	**63.1**	**69.8**	**62.0**	62.9	**81.6**	**78.1**	**73.0**	**77.2**	**74.9**
&1 llama2-70b-base	55.8	42.7	43.7	48.1	34.5	39.1	31.6	44.9	45.1	47.0	44.3	54.8	44.3
&2 llama2-70b-chat	81.9	90.0	87.6	88.6	**64.8**	**72.6**	**68.8**	74.5	80.9	77.1	72.3	67.8	77.2
&3 base+ours w/o adjective	72.4	80.5	75.8	77.4	43.6	51.1	29.4	64.3	77.1	72.3	71.1	67.0	65.2
&4 base+ours	**92.5**	**97.9**	**95.8**	**96.3**	63.0	71.1	55.3	**80.4**	**89.0**	**85.6**	**88.3**	**85.0**	**83.4**
◇1 falcon-40b-base	69.9	72.1	61.8	63.1	36.6	42.5	27.5	50.1	65.8	66.3	60.7	67.2	57.0
◇2 falcon-40b-instr.	78.9	89.2	80.0	83.2	51.5	55.2	40.3	74.7	**86.3**	**81.3**	**83.3**	85.3	74.1
◇3 base+ours w/o adjective	63.6	58.7	46.4	53.4	36.0	38.9	23.8	35.7	56.5	51.7	51.0	53.2	47.4
◇4 base+ours	**92.0**	**91.2**	**87.8**	**88.1**	**55.0**	**62.0**	**43.2**	**77.8**	84.1	80.6	80.3	85.3	**77.3**

lexicon-match baseline We add a sentiment lexicon match-based model [7], which directly utilizes the presence of positive (e.g., *great*, *good*, and *nice*) and negative words (e.g., *sad*, *bad*, and *worse*) to determine the sentiment polarities. This aims to determine if good performance can be achieved through simple sentimental word matching, without injecting these sentimental adjectives via instruction tuning.

llama2 chat model The Llama2 chat model began supervised fine-tuning with instructions from 1.8K NLP tasks [4]. The model was further fine-tuned on 27,540 annotated instructions and millions of human preference data via reinforcement learning. We believe this provides a powerful baseline, even for our sentiment classification task.

falcon chat model It is also known as the Falcon-40B-Instruct model[7], which is fine-tuned on hundreds of thousands of QA and dialog instances from Quora, Stack Overflow, and MedQuAD questions.

4 Result and Analysis

Table 2 shows comparison results and our observations are as follows:

(1) Our instruction-tuned models (**base+ours**) outperform all base models by 30 points and even all chat models by 5.1 points on average. Moreover, our instruction-tuned Llama2-70B model achieves the best average performance, suggesting that model size remains an important factor in the effectiveness of instruction tuning.

(2) The results of **base+ours w/o adjective** show significant performance degradation for Llama2-7B (#3), Llama2-13B ($3), and Falcon-40B (◇). While the "empty-input" instruction tuning boosts Llama2-70B's performance to some extent (&3), combining it with our sentimental adjectives achieves the best performance (&4). This verifies that the improvements are largely not attributed to learning the output space formats, such as positive and negative labels, as reported by previous work [12].

(3) To investigate whether our *base+ours* models simply memorize sentimental adjectives for making predictions, we added a sentiment lexicon match-based model for comparison. The results show that our models significantly outperform this baseline (△), indicating that incorporating sentimental adjectives into LLMs through instruction tuning equips the models to handle not only straightforward sentiment lexicon-based cases but also more challenging cases lacking explicit sentiment lexicons.

5 Conclusion

In this work, we create pseudo sentimental instructions to fine-tune LLMs. Experiments show significant performance gains on various sentiment benchmarks. Notably, it requires no ground-truth training data and generalizes well across domains. Future work will extend this approach to fine-grained emotion classification.

7) https://huggingface.co/tiiuae/falcon-40b-instruct

References

[1] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. **arXiv preprint arXiv:2311.16867**, 2023.

[2] Stefano Baccianella, Andrea Esuli, Fabrizio Sebastiani, et al. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In **Lrec**, Vol. 10, pp. 2200–2204, 2010.

[3] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. **arXiv preprint arXiv:2210.11416**, 2022.

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. **Journal of Machine Learning Research**, Vol. 25, No. 70, pp. 1–53, 2024.

[5] Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky. LLMs to the Moon? Reddit market sentiment analysis with large language models. In **Companion Proceedings of the ACM Web Conference 2023**, pp. 1014–1019, 2023.

[6] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. **Advances in Neural Information Processing Systems**, Vol. 35, pp. 30318–30332, 2022.

[7] Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In **Proceedings of the international AAAI conference on web and social media**, Vol. 8, pp. 216–225, 2014.

[8] Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, Toronto, Canada, July 2023.

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

[10] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In **Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining**, pp. 168–177, 2004.

[11] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. From group to individual labels using deep features. In **Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining**, pp. 597–606, 2015.

[12] Po-Nien Kung and Nanyun Peng. Do models really learn to follow instructions? an empirical study of instruction tuning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 1317–1328, Toronto, Canada, July 2023. Association for Computational Linguistics.

[13] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. **Journal of the Association for Information Science and Technology**, Vol. 65, , 2014.

[14] Kevin Scaria, Himanshu Gupta, Siddharth Goyal, Saurabh Arjun Sawant, Swaroop Mishra, and Chitta Baral. Instructabsa: Instruction learning for aspect based sentiment analysis. **arXiv preprint arXiv:2302.08624**, 2023.

[15] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In **Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing**, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[16] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[17] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. **arXiv preprint arXiv:2307.09288**, 2023.

[18] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. **arXiv preprint arXiv:2306.04751**, 2023.

[19] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, Toronto, Canada, July 2023.

[20] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 5085–5109, 2022.

[21] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. **arXiv preprint arXiv:2109.01652**, 2021.

[22] Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. **arXiv preprint arXiv:2304.01196**, 2023.

positive words	negative words	neutral words
sophisticated	contemptible	last-ditch
magna-cum-laude	bogus	alate
gorgeous	salt	floored
boss	unfree	quadrilateral
heaven-sent	hidden	forty
exhaustive	inhumane	french-speaking
superb	humble	combined
healthy	false	client-server
...	...	...

Table 3: **Step 1. Collect sentimental adjectives candidates.**

positive words	negative words	neutral words
sophisticated	contemptible	alate
gorgeous	bogus	quadrilateral
superb	inhumane	forty
healthy	false	french-speaking
meticulous	precarious	combined
perfect	upset	client-server
sweet	numb	trojan
coherent	indelicate	diagonal
...	...	...

Table 4: **Step 2. Align with sentiment word sense.**

positive words	negative words	neutral words
best	dead	new
great	poor	more
important	difficult	national
good	unable	most
better	bad	many
supreme	wild	american
golden	cold	early
greatest	offensive	high
...	...	...

Table 5: **Step 3. Rank word by frequency.**

positive words	negative words	neutral words
best	dead	new
great	poor	more
important	difficult	national
good	unable	most
better	bad	many
supreme	wild	american
golden	cold	early
not offensive	not greatest	high
...	...	...

Table 6: **Step 4. Add negation words.**