Influencing Sentiment in LLM Output via Token Selection Manipulation

Kaja Praprotnik*
kaja.praprotnik@famnit.upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Niki Hrovatin*
niki.hrovatin@famnit.upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Jernej Vičič*
jernej.vicic@upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Domen Vake*
domen.vake@famnit.upr.si
University of Primorska, UP FAMNIT
Koper, Slovenia

Abstract

In this work, we explore whether a static lexicon-derived token bias can steer a Large Language Model toward more positive or negative outputs, without any additional training. On Llama-3.1-8B-Instruct, a lexicon-derived prefix-count ratio lookup table manipulates the logits at each decoding step. Two login manipulations are tested: additive bias and multiplication/division boosting, each at several weights. The additive approach steers sentiment in the positive direction and is scored with the sentiment tool VADER, where the additive bias produces no significant steering. Based on that result, we approach the problem with a more aggressive multiplication/division method. The multiplication/division boosting produces significant sentiment steering in both directions, with small effect sizes. Steering is more effective in the positive direction.

Keywords

Sentiment Analysis, Large Language Models, Sentiment Steering

1 Introduction

Large Language Models (LLM) are incorporated in various fields, including service-oriented industries such as chatbots. Chatbots can be used in healthcare [3] and education [16] as an accessible communication tool. They can also be used as customer support [22] or sales assistant [5], helping users with various tasks. Interacting with a user depends on setting an appropriate tone of the conversation. A positive and uplifting tone could be used in an e-commerce setting, where friendliness and empathy are positively associated with trust in the chatbot [6]. In general, emotionally sensitive chatbots might be perceived as more capable and trustworthy to users in comparison to emotionally insensitive ones and using positive words has a positive impact on users [4, 25]. Sentiment steering also raises ethical concerns, particularly for vulnerable users, as increased trust in emotionally responsive chatbots may heighten susceptibility to influence and encourage overreliance. Responsible use should therefore include transparency about the system's artificial nature, user control over emotional adaptation, and safeguards against manipulative language or exploitation of emotional vulnerability. A detailed examination of these ethical issues is beyond the scope of this paper, but they remain important deployment considerations.

It might be worth exploring different ways of steering LLM outcomes into the right emotional direction. In this work, we explore whether we can steer the sentiment of LLM output by logit boosting based on a sentiment lexicon at each decoding step without any additional model training.

2 Related Work

A few papers are exploring different techniques for approaching the sentiment steering problem in LLMs. By using Controllable Text Generation (CTG) techniques, we can direct the model to reach specific objectives [17]. For example, using reinforcement learning (RL) fine-tuning through human feedback, the work of Ziegler et al. [27] solves the task of using high sentiment or physically descriptive language in stylistic continuation. Another way to simply influence the sentiment of the LLM's output is by prompt engineering. Monafal et al. [20] find that the model can be steered through task-specific instructions (prompts) towards generating summaries with a positive tone. An alternative strategy for controlled language generation is the Plug and Play Language Model (PPLM)of Dathathri et al. [8]. This method combines a pretrained large transformer-based model with simple attribute classifiers. PPLM can be used for language detoxification. A different approach to CTG is activation engineering [23], which uses the Activation Addition (ActAdd) technique. Based on this and on Concept Activation Vectors (CAVs) [15], the paper by Zhang et al. [26] introduce GCAV Framework. GCAV framework trains a CAV for a specific concept, such as toxicity, and then uses it to influence a model to control that same concept. Hiranandani et al. [9] propose Plugin, a token-level probability reweighting framework.

Liu et al. [18] modify logits directly at the decoding time by combining the base model's logits with the logits of additional expert and anti-expert models at each decoding step. The work of Yang and Klein [24] propose a related approach, where the base model's logits are modified at decoding time by combining them with the output of a trained binary classifier.

Close to our approach, An et al. [2] propose SWAI, a training-free steering method that biases token logits, using z-normalized one-vs-rest log-odds scores from labeled corpora. Similar to SWAI, our first approach applies a constant bias value. Unlike SWAI, which restricts biasing to a top-K set and recomputes the bias token set in each step, we do not restrict to top-K and instead select tokens from a precomputed, lexicon-driven token set. We test it on RLHF-instruction-tuned Llama-3.1-8B-Instruct, while

*These authors contributed equally.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.116

SWAI tests it on non-instruction-tuned models. An et al. [1] apply a similar method to apply it to a diffusion instruction-tuned model, where they find a clear asymmetry between toxic and non-toxic steering and suggest that instruction-tuned DLMs have a strong internal resistance to toxic continuations. In our work, we explore whether this asymmetry extends to sentiment steering in an autoregressive instruction-tuned model.

3 Problem formulation

Let T be a set of all the tokens in the tokenizer. Let $W = P \cup N$ be all of the words in the sentiment lexicon [11], where P is a set of all the positive words (e.g. *amusing*, *awesome*) and N is a set of all the negative words (e.g. *agonize*, *brutality*).

We say that a token $t \sqsubseteq_{\text{pre}} w$ for $w \in W$ if t is a *prefix* of a word w for any $t \in T$. Then, let *positive count* ($C^P(t)$) of a token t for any $t \in T$ be the number of the positive words $p \in P$ such that $t \sqsubseteq_{\text{pre}} p$, that is,

$$C^P(t) = |\{p \mid t \sqsubseteq_{\text{pre}} p, p \in P\}|. \quad (1)$$

Similarly, a *negative count* $C^N(t) = |\{n \mid t \sqsubseteq_{\text{pre}} n, n \in N\}|$ for any token $t \in T$. The *positiveness* of a token t for any $t \in T$ is then defined as

$$f^P(t) = \frac{C^P(t)}{C^P(t) + C^N(t)}. \quad (2)$$

This represents the probability of a token $t \in T$ turning into a positive word.

A token $t \in T$ is considered to be *positive* if its *positiveness* is more than 50%, *negative* - the *positiveness* is less than 50%, and *neutral* - the *positiveness* is exactly 50%. We construct sets of *positive* tokens T^+, *negative* tokens T^-, and *neutral* tokens $T^=$, so that tokens can be re-weighted at each decoding step based on the desired sentiment. Simply put: we increase the value of the desired logits by adding weight before they are converted into probabilities by the softmax function. Figure 1 presents the described flow.

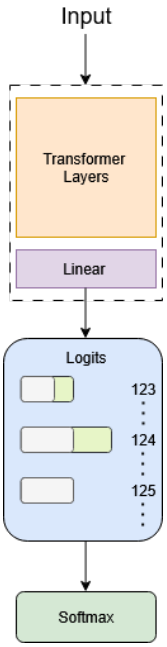

Figure 1: The value of logits is manipulated before the conversion.

The objective of the algorithm is to steer the model's output to the desired sentiment target by affecting token selection in the decoding process.

The algorithm alters the output logits before token selection, biasing it toward tokens that align with the desired emotional tone. Here the target is a more positive outcome, so the probability of selecting positive tokens is increased in two ways: an additive approach, described in Section 3.3, and the multiplicative approach described in Section 3.4.

3.1 Prompt dataset

To test our algorithm for three differently weighted cases and the baseline, we use an open-source instruction-following records dataset, databricks-dolly-15k [7] to prompt our model. The dataset consists of over 15,000 human-generated records with categories such as: *Creative Writing*, *Closed QA*, *Open QA*, *Summarization*, *Information Extraction*, *Classification* and *Brainstorming*. Categories *Closed QA*, *Open QA*, *Summarization*, *Information Extraction* and *Classification* are not suitable for our approach as their output adheres to the sentiment of the initial prompt by construction. Thus, in our experiment we use two categories from the dataset, *Creative Writing* and *Brainstorming*. The two categories are picked so that the model is not limited by the correctness of the answer from the start, but rather allow the maximum expression freedom. *Creative Writing* category contains 709 prompts and *Brainstorming* 1766.

3.2 Experimental setup

Open-source models allow direct access to the tokenizer and the output logits, which is necessary for our logit reweighting approach. We choose the Llama-3.1-8B-Instruct model, setting the decoding parameters based on the example provided by Meta [19]: $temperature = 0.6$ and $top_p = 0.9$. We keep $top_k = 50$, as specified in the Hugging Face transformers library [13].

Each prompt is run on different condition groups, using *apply_chat_template* from the Hugging Face transformers library [12] to format the input together with `add_special_tokens=False` to avoid duplicating special tokens. The template omits the `"role": "system"` field, to avoid sentiment biases before logit manipulation. Human evaluation sets the conditions on the output. Short text is read more carefully than long text, which the evaluator skims. The text also has to end on a sentence boundary. Generation therefore stops either at `eos_token_id` or, once the maximum of 300 tokens is reached, at the next end of sentence punctuation mark (!?.). A fixed random seed (1234) ensures reproducibility across all runs. The generation is run on A10G GPU. After the generation the outputs are evaluated with VADER (Valence Aware Dictionary and Sentiment Reasoner) a lexicon and rule-based sentiment analysis tool [14]. The VADER handles the slang and other informalities that our prompts might produce. For each output, VADER calculates the compound score, which has a value between −1, which is the most negative, and +1, which is the most positive. The sentiment of the text is considered *positive* if the compound score is greater than or equal to 0.05. The sentiment of the text is considered *neutral* if the compound score is between -0.05 and 0.05 and negative if the compound score is less than or equal to -0.05 [14].

3.3 Additive approach

Our first approach follows the principle of OpenAI's *logit_bias* parameter [21]. Manipulation of the logits is done by applying

an additive bias to the logits of positive sentiment tokens T^+ at every generation step. The additive approach applies the same magnitude of steering for every token in the selected set. Logits are modified at each decoding step:

$$\ell_i' = \ell_i + (w - 1), \quad \forall i \in T^+ \tag{3}$$

where ℓ_i is the logit of token i, w is the manipulation weight. Additive bias is chosen, so a token, whether it has a positive or negative logit, would be boosted. The bias is defined as $(w - 1)$, so $w = 1$ is the baseline.

The outputs also have to be complete, and not end in the middle of a sentence: an incomplete text invites the evaluators to judge it on its incompleteness rather than on its the sentiment. Outputs that reach the token limit are therefore eliminated, and those ending with the *eos* token or at a sentence boundary are kept. To determine the range of weights with complete outputs, we conduct a preliminary study. Applying the additive formula to the logits, outputs are generated from 12 prompts, in the range from 1.0 to 2.0 in steps of 0.1. All of the outputs terminate within the token limit.

Since all of the weights produce complete outputs, the main experiment is conducted with weights that are equally spaced by 0.25, starting from the baseline. From each category, *creative writing* and *brainstorming* we pick the 100 most positive, 100 most negative and 100 random neutral prompts. After removing duplicate prompts, 99 negative prompts and 97 neutral prompts in the *creative writing* category are left. On each of the six groups (positive prompts, negative prompts, neutral prompts of the two categories) weights $w \in \{1.0, 1.25, 1.5, 1.75\}$ are applied to the additive formula and steer in a positive direction.

The outputs are tested with Shapiro-Wilk tests, which show non-normality in each group. We use the Friedman test to compare the repeated measurements at different weights for the same prompts.Wilcoxon signed-rank tests with Bonferroni correction are also used for pairwise comparisons to the baseline and Kendall's W for the effect size. Five condition groups show no significant difference across weights, with p ranging from 0.7 to 0.88 and Kendall's $W < 0.005$. One condition group (category "Brainstorming", 100 most positive prompts) shows $p = 0.04$ but with a weak effect ($W = 0.028$) and no significance after the Bonferroni correction. In this group, the steering shifts the sentiment in the negative direction even though we intend to steer into the positive direction. After the correction, no pairwise comparison with the baseline reaches significance in any group, with the lowest corrected $p = 0.31$.

We raise weights to $w \in \{1.0, 2.0, 3.0, 5.0\}$, checking whether previous weights were too small. The Friedman test on a small sample size of 24 prompts (one duplicate instruction removed) does not show a significant difference ($p = 0.09$). The bias moves 0.5 to 1.3 extra tokens into the top-k set, while only 0.09 to 0.3 of them reach the top-p. Based on this, we decide that rather than testing an additive approach for steering into the negative direction, we try a different, more aggressive approach.

3.4 Multiplication/division of logits

An alternative manipulation of the logits is explored to achieve a stronger boost. Positive logits are boosted by multiplying them by different weights, and negative tokens by dividing them by the same weights. Our approach mirrors the work of Hollows [10], which uses a similar approach for penalizing repetition logits. The multiplication of logits does not apply the same magnitude of steering to each token in the selected set. A token with a logit further from zero will move further than the token closer to zero. The multiplication/division formula makes the manipulation depend on the arbitrary zero-point of the logits [10]. The arbitrariness of the zero-point of logits comes from the softmax function, which depends only on the differences between logits. Shifting all of the logits by the same constant produces the same output, which means the steering strength is dependent on where the model's logits lie.

The range of weights is established based on the completeness of the output text. Because the steering of the logits by multiplication is larger than with the additive formula, the upper weight is set to $w = 1.25$. The range between the baseline and upper weight in steps of 0.05 is tested, that is $w \in \{1.0, 1.05, 1.10, 1.15, 1.20, 1.25\}$. We generate outputs for 25 prompts at each weight. All the outputs generated with weights from $w = 1$ to $w = 1.20$ terminate within the token limit, while at $w = 1.25$ 24% of the outputs reach the token limit.

This upper limit is kept for the main experiment and the outputs that reach the token limit are filtered out for the human evaluation. The main experiment uses weights $w \in \{1.0, 1.05, 1.1, 1.15, 1.20, 1.25\}$, where $w = 1$ is a baseline and $w = 1.25$ is the upper limit. In each case, the function with the set of weights is applied to the logits in each decoding step. Each weight has a different steering magnitude, and unlike the additive formula, it scales differently between each token.

Since it is hard to evaluate sentiment from the text that does not form full sentences, the prompts are filtered. Prompts are removed if they begin with "Give me a list" or their response in the dataset contains a list, has four or fewer words, or has no verb. We take 60 prompts with the highest VADER score from the filtered prompts, 60 with the lowest VADER score, and 60 random prompts with a neutral score. Combining the two categories within each sentiment, gives 120 prompts for each sentiment group. A fixed random seed of 2025 is used.

The Friedman test shows a significant difference across weights in all of the condition groups, with $p < 0.001$ after the Bonferroni correction. Effect sizes for steering into the positive direction range between 0.255 and 0.267, while effect sizes for steering into the negative direction range between 0.081 and 0.154. A similar steering asymmetry is observed in the work of An et al. [1], who report model resistance to toxic steering in an instruction-tuned diffusion model, even though the steering goal, model type and method differ from ours. The pairwise comparisons against baseline reach significance at $w = 1.1$ in 4 groups: positive and neutral prompts steered into the positive direction and positive and neutral prompts steered into the negative direction. The group with negative prompts steered into the positive direction reaches significance in pairwise comparison against the baseline at $w = 1.05$ and the group with negative prompts steered into the negative direction reaches significance in pairwise comparison against the baseline at $w = 1.15$. All significant comparisons shift the sentiment into the intended direction. For example, steering into the negative sentiment with a weight of $w = 1.25$ for the negative-input prompt: "Write a short story (under 250 words) about meditation. The narrator should be uncomfortable." produces an output that ends with: "My mind refused to be stilled, a rebellious force refusing to be tamed by my feeble attempts at serenity." (VADER −0.98), while the last sentence for steering into positive sentiment is: " I might not have relaxed as promised, but at least we were free to go." (VADER +0.99).

4 Limitations

The main evaluation tool in this work is VADER, which determines the sentiment based on a sentiment lexicon. Our steering method is also lexicon-based, using the Hu and Liu lexicon with 32.6% of words also appearing in VADER's lexicon. This may result in some evaluation bias, so the results should be paired with a human evaluation that is yet to be conducted. The effect sizes are small and the multiplication/division formula can steer differently, based on the logits' position relative to zero. We do not test whether restricting the additive bias to a top-K candidate set would resolve the failure we diagnose. Results are limited to Llama-3.1-8B-Instruct.

5 Conclusion and Future Work

We test whether sentiment steering is possible with lexicon-driven logit manipulation on Llama-3.1-8B-Instruct, under two different logit manipulations: an additive and a multiplication/division approach. The additive approach shows no significant effect, while the multiplication/division approach shows significant bidirectional steering in all of the condition groups. Future work should test the steering approaches on different models and examine the effect based on the positioning.

References

[1] Hyeseon An and Yo-Sub Han. 2026. Dlm-swai: steering diffusion language models before they unmask. *arXiv preprint arXiv:2605.29626*.

[2] Hyeseon An, Shinwoo Park, Hyundong Jin, and Yo-Sub Han. 2026. Steering language models before they speak: logit-level interventions. *arXiv preprint arXiv:2601.10960*.

[3] Lekha Athota, Vinod Kumar Shukla, Nitin Pandey, and Ajay Rana. 2020. Chatbot for healthcare system using artificial intelligence. In *2020 8th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)*. IEEE, 619–622.

[4] Antonin Brun, Ruying Liu, Aryan Shukla, Frances Watson, and Jonathan Gratch. 2025. Exploring emotion-sensitive llm-based conversational ai. *arXiv preprint arXiv:2502.08920*.

[5] Sijia Cheng, Wen-Yu Chang, and Yun-Nung Chen. 2025. Exploring personality-aware interactions in salesperson dialogue agents. In *Proceedings of the 15th International Workshop on Spoken Dialogue Systems Technology*, 60–71.

[6] Xusen Cheng, Ying Bao, Alex Zarifis, Wankun Gong, and Jian Mou. 2022. Exploring consumers' response to text-based chatbots in e-commerce: the moderating role of task complexity and chatbot disclosure. *Internet Research*, 32, 2, 496–517.

[7] Mike Conover et al. 2023. Free dolly: introducing the world's first truly open instruction-tuned llm. Retrieved June 30, 2023 from https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

[8] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: a simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

[9] Gaurush Hiranandani, Haolun Wu, Subhojyoti Mukherjee, and Sanmi Koyejo. 2025. Logits are all we need to adapt closed models. *preprint arXiv:2502.06806*.

[10] Peter Hollows. 2026. Gauge dependence and structured-output corruption in sign-branched repetition penalties: measurements across models, inference stacks, and alternative repetition controls. *arXiv preprint arXiv:2607.09791*.

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

[12] Hugging Face. 2026. Chat templates. https://huggingface.co/docs/transformers/en/chat_templating. Transformers documentation, Hugging Face. (2026). Retrieved July 29, 2026 from.

[13] Hugging Face. 2024. Generation. Retrieved July 19, 2026 from https://huggingface.co/docs/transformers/main_classes/text_generation.

[14] Clayton Hutto and Eric Gilbert. 2014. 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* number 1. Vol. 8, 216–225.

[15] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*. PMLR, 2668–2677.

[16] Lasha Labadze, Maya Grigolia, and Lela Machaidze. 2023. Role of ai chatbots in education: systematic literature review. *International journal of Educational Technology in Higher education*, 20, 1, 56.

[17] Xun Liang et al. 2024. Controllable text generation for large language models: a survey. *arXiv preprint arXiv:2408.12599*.

[18] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A Smith, and Yejin Choi. 2021. Dexperts: decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 6691–6706.

[19] Meta AI. 2024. Llama3/example_chat_completion.py. https://github.com/meta-llama/llama3/blob/main/example_chat_completion.py. GitHub repository. Accessed: 2026-07-19. (2024).

[20] Victor Monafal, Lawrence Patterson, Augustus Petrov, and Nicholas Robertson. 2024. Optimizing positive content generation in prompt-based abstractive summarization with large language models.

[21] OpenAI. 2024. Using logit bias to alter token probability with the OpenAI API. https://help.openai.com/en/articles/5247780-using-logit-bias-to-alter-token-probability-with-the-openai-api. Accessed: September 22, 2026. OpenAI Help Center, (2024).

[22] Elitza Stoilova. 2021. Ai chatbots as a customer service and support tool. *ROBONOMICS: The Journal of the Automated Economy*, 2, 21–21.

[23] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

[24] Kevin Yang and Dan Klein. 2021. Fudge: controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3511–3535.

[25] Jeewoo Yun and Jungkun Park. 2022. The effects of chatbot service recovery with emotion words on customer satisfaction, repurchase intention, and positive word-of-mouth. *Frontiers in psychology*, 13, 922503.

[26] Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. 2025. Controlling large language models through concept activation vectors. *arXiv preprint arXiv:2501.05764*.

[27] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.