SENSESHIFT: Continuous Sentiment-Controlled Text Generation via Encoder-based Mask Infilling

Shahed Masoudian[1], Markus Frohmann[2,3], Emmanouil Karystinaios[1], Navid Rekabsaz[2], Markus Schedl[1]
[1] Johannes Kepler University Linz
[2] Thomson Reuters Labs
[3] University of Toronto, Vector Institute
Correspondence: shahed.masoudian@jku.at

Abstract

Recent controllable text generation (CTG) for sentiment control has largely focused on decoder-based large language models, making causal attention the dominant paradigm. While effective for fluent generation, these models still struggle to satisfy complex constraints and follow fine-grained sentiment signals specified by users. Existing sentiment-aware CTG methods typically simplify the problem by treating sentiment either as a coarse categorical label (e.g., positive or negative) or as a single fine-grained control signal applied to an entire document. Consequently, more challenging settings such as sentence-level sentiment control within long-form text remain underexplored. To address these limitations, we introduce SENSESHIFT, an encoder-based framework for fine-grained sentence-level CTG. Unlike standard decoder architectures, SENSESHIFT leverages bidirectional attention, quantized sentiment signals, and iterative mask infilling to generate local sentences conditioned on target sentiment intensity. Empirical evaluations on story and review generation demonstrate that SENSESHIFT achieves stronger sentiment controllability while maintaining text quality and robustness to out-of-domain generation compared to larger decoder-based baselines. [1]

1 Introduction

Controllable Text Generation (CTG) aims to steer generated text toward desired attributes specified using external conditions (Zhang et al., 2023). In sentiment-aware CTG, the goal is to regulate the emotional tone of text based on users desire (Lorandi and Belz, 2023). Prior works investigated categorical sentiment control for desired segments (Yuan et al., 2022; Liang et al., 2024) and fine-grained sentiment control of full-document (Luo et al., 2019b; Tang and Wu, 2021). These works predominately rely on decoder LLMs (Yang et al., 2024), making causal attention a default paradigm for sentiment-aware CTG. We argue that this choice of architecture is not best fit for scenarios where users wish to rewrite sentences of a document with a desired fine-grained sentiment value, an area still underexplored.

Recent studies show that decoder-only architectures are fundamentally constrained by their unidirectional information flow (Kopiczko et al., 2025). Moreover, CTG via prompting requires precise task definition which often result in unstable or weakly controlled outputs, especially under ambiguous or subtle sentiment targets (Labroo et al., 2026; Laban et al., 2025; Zhao et al., 2021). In particular, controlling sentiment *intensity* even as categorical attribute (e.g., *happy* vs *very happy*) remains challenging for prompting-based approaches. We provide empirical evidence for this limitation in our task setting in Appendix A.1.

To address these limitations, we introduce SENSESHIFT, a novel encoder framework for fine-grained sentiment-aware in-text generation. Unlike decoder-based approaches that rely on causal attention and generate strictly left-to-right, SENSESHIFT leverages bidirectional attention together with iterative mask infilling. This enables in-text generation conditioned jointly on prior and posterior context, so that generated sentences integrate more naturally with the surrounding text. SENSESHIFT consists of three stages: automatic sentiment signal construction and quantization, control-aware MLM fine-tuning, and iterative mask infilling at inference. Together, these enable fine-grained manipulation of sentiment in arbitrary sentences within a long text. Figure 1 illustrates how SENSESHIFT generation differs from decoder prompting.[2]

[1] The source code for experimental setup and training is available at https://github.com/ShawMask/SenseShift, and the model checkpoints for inference can be found at https://huggingface.co/shawhed/SenseShift-large.

[2] Qualitative examples of text modified with SENSESHIFT

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 22144–22163
October 24-29, 2026 ©2026 Association for Computational Linguistics

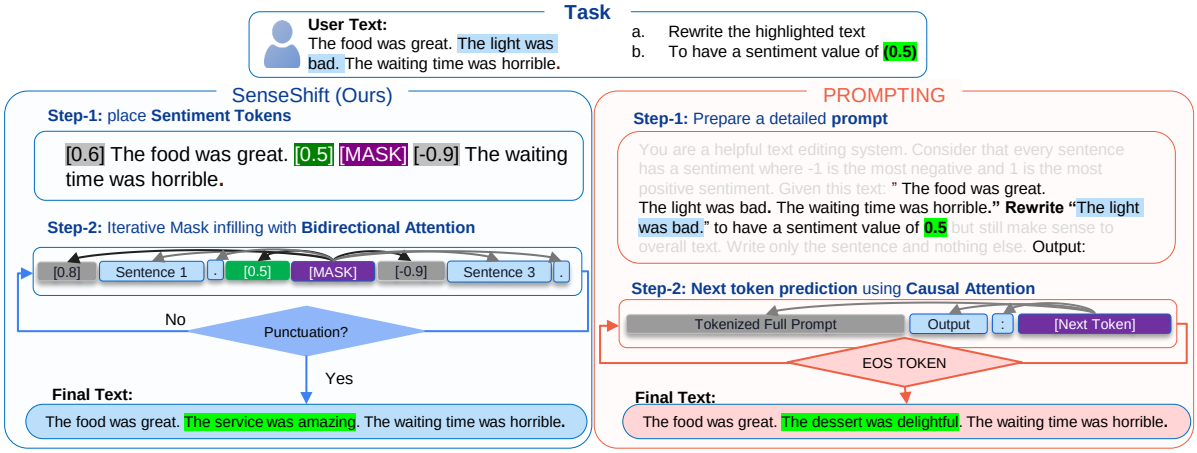


Figure 1: SENSESHIFT VS decoder-based prompting for sentiment CTG. Based on a given text, target sentence and sentiment, SENSESHIFT places sentiment tokens before each sentence. encoder-architecture allows SENSESHIFT to predict masked tokens attending to each target sentiment while distinguishing prior context from posterior.

We instantiate SENSESHIFT on MODERNBERT (Warner et al., 2025) *base* and *large* substantially smaller than the decoder baselines. Through training and evaluation on review and story-writing datasets we demonstrate empirically that SENSESHIFT consistently outperforms decoder-based baselines including prompting, activation steering, and instruction tuned models many times its size on sentiment control while maintaining fluency and contextual fit. Human evaluation confirms this: annotators judge SENSESHIFT outputs slightly more fit to context compared to a larger decoder baseline model while also preferring SENSESHIFT for sentiment alignment. Together, these findings suggest that encoder-based generation is an effective and competitive alternative to autoregressive decoding for fine-grained sentiment-aware in-text generation.

2 Related Work

Early approaches to sentiment and style control used latent-variable or recurrent formulations to inject attribute information during generation (Hu et al., 2017; Peng et al., 2018). These works assume that sentiment is a categorical label and steer generation toward those classes, for tasks such as poetry generation (Shao et al., 2021), multi-aspect control over sentiment and topic (Ding et al., 2023), reinforcement-style unlearning (Lu et al., 2022), and contrastive prefix methods (Qian et al., 2022; Zheng et al., 2023). Decoding-time steering methods (Pascual et al., 2021; Krause et al., 2021; Liu et al., 2021; Yang and Klein, 2021) and supervised fine-tuning approaches (Yang et al., 2024; Żal et al., 2024) share the same coarse-grained, document-level limitation. Across all these lines, sentiment is treated as a polarity signal and control operates over full sequences.

A smaller body of work has explored fine-grained control signals, including early work on controlling gender information within encoders for fairness (Masoudian et al., 2024) using gated adapters. Subsequent works in decoder-based generation focuse on fine-grained sentiment transfer via Gaussian kernel conditioning on standalone sentences (Luo et al., 2019b), scorer-based unsupervised control (Jain et al., 2019), emotional control in story generation via decoder-side guidance (Wang et al., 2022), and continuous interpolation mechanisms (Kangaslahti and Alvarez-Melis, 2024; Samuel et al., 2025). These methods target sentiment granularity but operate at the document level (i.e., ignoring sentiment dynamics).

The works structurally most similar to ours introduce positional awareness into CTG. Story-ending control uses fine-grained sentiment and Gaussian kernel conditioning to guide only the concluding segment of stories (Luo et al., 2019a). Other methods use conditional variational autoencoder, aspect-level disentanglement, hierarchical templates, and dynamic attribute graphs to control sentiment of parts of a text (Qiao et al., 2020; Yuan et al., 2022; Nawezi et al., 2023; Liang et al., 2024). These works demonstrate the value of structural

can be found in Figures 6 and 7.

awareness but use categorical sentiment and do not move beyond decoder-based architecture.

From an architectural perspective, SENSESHIFT is most closely related to the text infilling paradigm (Donahue et al., 2020). Unlike their work, which fine-tunes a decoder model on masked text without attribute signal, SENSESHIFT leverages the encoder that attends jointly to both surrounding context and a quantized sentiment intensity signal before generating the target sentence.

3 Methodology

We aim to have sentence-level and fine-grained control over the sentiment of generated text which is particularly suitable for encoders. Their bidirectional attention and masked token prediction allow them to generate within a long context, a special feature that autoregressive decoders do not inherently have (Kopiczko et al., 2025). SENSESHIFT achieves this goal using the following steps:

3.1 Automatic Sentiment Signal Construction and Quantization

First key obstacle for fine-grained sentiment control is the lack of high-quality labeled sentence-level sentiment in long-form corpora, which can be used to train the models. We automatically derive sentiment scores using VADER (Hutto and Gilbert, 2014) as done in prior works (Konen et al., 2024) with one major difference: We extract the sentiment scores per sentence. We chose VADER over more accurate but slower alternatives such as RoBERTa (Barbieri et al., 2020) because it is fast, deterministic, and derived from human annotation, making it practical for large-scale preprocessing and online inference without adding additional complexity, parameters, and latency.

Given a document $\mathcal{D}$ segmented into m sentences $\mathcal{D} = \{s_1, s_2, \ldots, s_m\}$, we compute for each sentence s_i the sentiment score $v_i \in [-1, 1]$. Scores are then quantized to a discrete control grid with step size

$$\delta = 0.1 : \sigma_{s_i} = \text{round}\left(\frac{v_i}{\delta}\right) \cdot \delta$$

Each quantized value is mapped to a unique sentiment token $[\sigma_{s_i}]$ and placed at the beginning of its respective sentence. This converts unlabeled document into sentence-level controllable training data, providing the model with local sentiment intensity signals rather than a single document-level sentiment.

3.2 Control-Aware MLM Fine-Tuning

Training SENSESHIFT is standard Mask Language Modeling (MLM) under two control-aware modifications. First, sentiment tokens $[\sigma_{s_i}]$ are never masked and remain visible throughout training as persistent conditioning anchors.[3] Second, we increase the masking ratio to 40%, to encourage stronger dependence on both surrounding context and sentiment anchors when reconstructing masked content rather than relying on local co-occurrence patterns.

3.3 Iterative Mask Infilling

Predicting all masked tokens in parallel often results in repetitive and incoherent generations, since top token probabilities are predicted independently without sequence-level awareness (see Appendix A.3). To address this, we mimic iterative generation and beam search generation of decoders and implement a simplified iterative mask infilling approach (Wu et al., 2016; Vijayakumar et al., 2016) inspired by prior works.

During generation, the original sentiment token of a target sentence $[\sigma_{s_i}]$ is replaced with a user-defined target sentiment $[\sigma'_{s_i}]$. The target sentence s_i is fully corrupted with [MASK] token, producing a masked input X'_{masked}. Starting from the first masked position after the sentiment token, SENSESHIFT predicts one token at a time. At each step t, the model computes a probability distribution over the vocabulary given the current context Z_t, which includes the surrounding document, previously predicted tokens, and the target sentiment signal:

$$P(\cdot \mid Z_t) = \text{Softmax}\left(\frac{f_\theta(Z_t)}{\tau}\right)$$

where $f_\theta(Z_t)$ is the model's logit vector and τ is a temperature parameter controlling output sharpness.

Beam Search and Scoring. To explore the high-probability candidate space, we maintain B active beams from the top-k tokens. Candidates are evaluated using a length-normalized log-probability

[3] Initial experiments revealed that when sentiment tokens are trainable, overall performance degrades, as the model tends to reproduce them rather than using them as generation constraints.

score to prevent bias toward shorter completions:

$$S = \frac{1}{\text{LP}(n)} \sum_{t=1}^{n} \log P(w_t \mid Z_t)$$

where $\text{LP}(n) = \frac{(5+n)^\alpha}{(5+1)^\alpha}$. Here n is the number of predicted tokens, α is the length penalty hyperparameter, and 5 is a smoothing constant following Wu et al. (2016). To prevent beams from collapsing into repetitive outputs, we apply a diversity penalty γ that discourages multiple beams from selecting the same token. Formally, when scoring token w for beam b, we count how many higher-ranked beams $b' < b$ have already selected w at the current step and penalize accordingly:

$$S_{\text{final}} = S - \gamma \cdot \sum_{b' < b} \mathbf{1}[w_{b'} = w]$$

.

A beam is marked complete when the predicted token belongs to the terminal punctuation set,[4] at which point any remaining [MASK] tokens in that beam are replaced with [PAD] tokens. Generation is also capped at a maximum of 30 tokens to prevent degenerate outputs.[5] The resulting sentence s'_i is conditioned on the surrounding bidirectional context $\mathcal{D} \setminus \{s_i\}$, the neighboring sentiment signals, and the target constraint σ'_{s_i}.

4 Experimental Setup

We deployed SENSESHIFT on MODERNBERT (Warner et al., 2025) in two sizes namely MODERNBERT-E-0.15B (base, 149M parameters and MODERNBERT-E-0.4B (large, 395M parameters), a pre-trained bidirectional encoder we preferred over predecessors such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) due to its significantly higher contextual length and modernized training recipe. Regardless, SENSESHIFT framework is considered independent of model and can be deployed on any other encoder model with Masked Language Modeling. For evaluation we compared SENSESHIFT across three baseline categories: prompting, activation steering, and fine-tuning methods, all of which we introduce in detail in the following. To assess our method, we train and evaluate on two distinct datasets, measuring both in-domain performance and out-of-domain (OOD) performance. The latter specifically checks for robustness and is relevant for prompting, which by nature operates domain-agnostic. Our dataset selection ensures that both text structure and sentiment dynamics vary throughout the text.

TinyStories (Eldan and Li, 2023) is a corpus of ∼4.5M short stories generated by GPT-3.5/4.0. The stories are two to three paragraph while including various arcs (e.g., bad ending, inclusion of a twist, dialogues) resulting in sentiment changes.

Yelp Reviews (Zhang et al., 2015) ∼300K samples of human-written reviews spanning multiple service domains with user ratings. We discard user ratings and capture sentence-level sentiments (e.g., praising food quality while criticizing service) using the approach described in Section 3.1.

We remove formatting artifacts, discard single-sentence documents, and exclude texts exceeding 500 words, allowing high-quality, structurally complete data. From each dataset, 2000 samples are held out as test sets, unseen during training/validation. SENSESHIFT is trained/validated independently on each dataset but tested on both. Training hyperparameters, sentiment distribution and datasets statistics, are reported in Appendix A.2.3.

4.1 Baselines

We compare SENSESHIFT against four controllable generation paradigms, covering the main families of LLM-based sentiment control. Baselines include open-weight models: MODERNBERT-D-28M (28M)[6], LLaMA Instruct (3.2-3B, 3.1-8B), Gemma2 Instruct (2B, 9B), Qwen 3 Reasoning (32B), and OSS model (20B, 120B), as well as a closed-source GPT4O-MINI. Due to GPU memory constraints models up to 3 billion parameters were trained for all controlling methods while larger models are only controlled using zero-shot prompting. Details of prompts and instructions can be found in Section A.2.1. The sentiment controlling methods are as follow:

Prompting. We asked models to replace a randomly selected sentence with one sentence matching a numerically specified target sentiment.

Instruction Tuning. Fine-tunes decoder models specifically for sentence generation given a target

[4] Punctuation tokes are {. , ; , ! , ?}.

[5] This threshold can alternatively be set to the original sentence length to enforce length-matched generation.

[6] A small version of MODERNBERT architecture trained from scratch for next token prediction with causal attention.

sentiment score. Models are trained to replace a `[Missing-Sentence]` with a substitute.

Token Instruction Tuning. Extends instruction tuning with the quantized sentiment tokens from SENSESHIFT (Section 3), conditioning generation on the target sentiment at inference. This decoder baseline shares SENSESHIFT's sentiment representation, isolating the effect of bidirectional infilling.

Activation Steering. Identifies and steers the activation vectors of models without training, following Rimsky et al. (2024). During decoding, internal activations are steered along a sentiment axis learned from contrastive sentence pairs, to control generated output without fine-tuning. Full implementation details are provided in Appendix A.2.2.

4.2 Evaluation Metrics

We utilize standardized evaluation protocol applied identically to all baselines and SENSESHIFT. For each document, we randomly select a sentence and assign it a random target sentiment. Models then generate the sentence conditioned on the target. All generated outputs are scored using VADER, the same sentiment analyzer used for training of all baselines, ensuring a consistent and unbiased comparison across baselines (except for zero-shot prompting). Our metrics capture two key aspects: (1) adherence to the target sentiment signal, and (2) fluency and fitness of the generated sentence with its surrounding context.

Delta Sentiment (Δ_s). Measures the mean absolute error between the target sentiment and the generated sentence sentiment. For a sentence s_i from document j, this is defined as

$$\Delta_{s_i} = |\hat{\sigma}_{s_i} - \sigma'_{s_i}|$$

where $\hat{\sigma}_{s_i}$ is the new sentiment of the generated sentence and σ'_{s_i} is the target sentiment.

Sentiment Accuracy ($Acc.$). We evaluate sentiment accuracy based on standard positive, neutral, and negative categories. We define neutral band as $[-0.2, 0.2]$ (wider compared to VADER threshold of ± 0.05 (Hutto and Gilbert, 2014)) to avoid considering near-neutral intensity targets as failed control, and uniformly across all baselines and original sentences. Scores higher and lower than this range are labeled positive or negative respectively.

Correlation ($Corr.$). measures the Pearson correlation between target sentiment σ'_{s_i} and generated sentence sentiment $\hat{\sigma}_{s_i}$. A value of 1.0 indicates perfect linear adherence to the control signal, while 0.0 indicates complete failure in sentiment steering.

Perplexity ($PPL.$). is computed at the sentence level using GPT-2 and serves as a standard proxy for fluency and coherence. It measures the predictability of generated tokens but does not capture alignment with the surrounding document context, motivating the following metric.

Fitness (Δ_f). Standard n-gram metrics such as BLEU or ROUGE are not the best choice to measure contextual fitness because the generated sentence may intentionally diverge from the original sentence through sentiment manipulation. Hence, we propose a composite metric combining semantic similarity and entity-level overlap. For a generated sentence s'_i at position i, we compute cosine similarity scores using Sentence-BERT embeddings (Reimers and Gurevych, 2019)[7] for the preceding context $s_{<i}$ and following context $s_{>i}$ separately. Boundary sentences only have single similarity: $CS_{<i}$ alone for the final sentence and $CS_{>i}$ alone for the first sentence.

We complement the similarity score with an *entity overlap* score that checks whether named entities in the generated sentence are grounded in the surrounding document. Formally we write:

$$EC_i = \frac{|\text{Entities}(s'_i) \cap \text{Entities}(\mathcal{D} \setminus \{s_i\})|}{|\text{Entities}(s'_i)|}$$

and when s'_i contains no named entities, we set $EC_i = 1$, since the absence of entities introduces no grounding violation. The fitness score for a non-boundary sentence combines the two similarity scores with the entity overlap score, weighting the latter by 2 so that entity grounding contributes on par with the aggregate similarity signal:

$$C_i = \frac{CS_{<i} + CS_{>i} + 2 \cdot EC_i}{4}$$

Entity overlap flags fabricated entities but does not verify that entities are used in contextually appropriate relations, which we leave to the similarity terms.

For every document we compute the fitness score for the generated and original sentence, and report the calibrated fitness score as:

$$\Delta_{f_i} = C(s'_i) - C(s_i)$$

[7] Specifically *all-mpnet-base-v2*.

A value of zero indicates no relative change in contextual fit, while a positive value indicates improved fitness, and a negative value indicates degradation. We assess whether Δ_{f_i} significantly diverges from zero using a two-sided t-test, with results reported in Section 5.

Human Evaluation. We complement automatic evaluation with a user preference study comparing SENSESHIFT against the closest competitive baseline, namely GEMMA2-2B trained with token instruction tuning. Eight annotators evaluated 100 parallel documents (55 stories and 45 reviews) generated by both models using the same target sentence and sentiment intensity. They selected their preferred anonymized output based on three quality criteria: *contextual fit* (fitness with the surrounding text), *grammatical fluency* (sentence fluency independent of context), and *sentiment following* (alignment with the target sentiment value on a -1 to $+1$ scale). For each criterion, annotators chose among four options namely *A*, *B*, *Both*, or *Neither* without knowing which model produced each sentence. We report the preference percentages in Section 5, with full details provided in Appendix A.4.

5 Results and Discussion

Table 1 reports results on the story and review datasets. The top block include prompting baselines that require no training and are domain agnostic. In the table, columns PPL_{old} and PPL_{new} demonstrate the perplexity of the sentences before and after the generation respectively. Note that sentences are selected randomly from the same pool of 2000 evaluation samples hence the selected sentencce perplexity value has variation. More over in general the original PPL_{old} of the review dataset is high due to unconventional human review writing style (e.g., *best best best food that i had*) which shows reveals itself in old sentence perplexity value. The old perplexity of sentences is used as a reference point to give some idea about the distribution of sentence perplexity randomly selected for evaluation and how models new generation perplexity compares with the old sentences. The **In-Domain Evaluation** block reports models trained and evaluated on the same dataset, while the **Out-of-Domain Evaluation** block reports models trained on one dataset and evaluated on the other. Note that we use green and red color code to distinguish models trained on story from review. For instance models trained on story are evaluated on story as in-domain while same models evaluated on review as OOD evaluation. The OOD block is provided to allow a fairer comparison with domain agnostic prompting and should be read as robustness check not generalization claim.

Among prompting baselines, sentiment following improves with scale: GEMMA2-9B already has decent sentiment accuracy of 0.55 and OSS-120B achieves the highest correlation (0.793 on story, 0.74 on review) and accuracy (0.66, 0.64), though at a very high computational cost. Its elevated perplexity relative to OSS-20B is likely attributable to more complex vocabulary penalized by the GPT-2 scorer. Across prompting baselines, Δ_f on story are often significantly negative, indicating that prompted sentence generations are typically less contextually well-fitted than the original sentence even-though models have access to the original sentence during prompting unlike other baselines.

Standard instruction tuning improves Δ_s, Corr. and Acc. over the corresponding prompting baselines, but gains are uneven. MODERNBERT-D-28M is the most unstable, with perplexity spiking above 400 and better but significant Δ_f drops, likely reflecting the small size of the model. Token instruction tuning improves even more: GEMMA2-2B during In-domain evaluation improves accuracy from 0.49 to 0.75 on stories and 0.49 to 0.67 for reviews as with Δ_s of 0.21 by injecting SENSESHIFT sentence-level controllable sentiment tokens. However, most decoder models under this paradigm when evaluated as out-of-domain show significant Δ_f drops (except for LLAMMA3.2-3B trained on story evaluated on review), suggesting that their generation, while grammatically acceptable, often break contextual fitness. Steering underperforms across all model sizes, with $\Delta_s \approx 0.53$–0.56, near-zero or negative correlations, accuracy in the 0.28–0.39 range, and perplexity up to 221, frequently worse than the corresponding prompting baseline even when steering vectors are extracted from the same dataset. With steering only GEMMA2-2B shows a marginal improvement, hinting that activation steering might be poorly suited for fine-grained sentiment control at sentence level.

Overall SENSESHIFT achieves the strongest overall performance. MODERNBERT-E-0.4B achieves the lowest perplexity (53.6 on Story, with MODERNBERT-E-0.15B at 55.3 on Review) during in-domain evaluation, the smallest sentiment devi-

Table 1: Overall comparison between baselines and SENSESHIFT. Significantly different changes (p-value<0.01) compared to the 0 baseline are marked with an asterisk. Note that columns under green are same models trained on story and columns under red are same models trained on review. in table PPL_{old} refers to perplexity of the original sentence calculated using GPT-2 and PPL_{new} refers to generated sentence perplexity

Model	Story Evaluation						Review Evaluation					
	PPL_{old}	PPL_{new} ↓	Δ_s ↓	Δ_f ↑	Corr. ↑	Acc. ↑	PPL_{old}	PPL_{new} ↓	Δ_s ↓	Δ_f ↑	Corr. ↑	Acc. ↑
Prompting												
MODERNBERT-D-28M	80.6	255.5	0.56	-0.29*	0.07	0.36	222.3	161.2	0.69	-0.13*	-0.19	0.23
LLAMMA3.2-3B	81.1	118.1	0.50	-0.05*	0.25	0.41	239.4	113.1	0.51	-0.00	0.21	0.38
GEMMA2-2B	88.6	139.2	0.56	-0.00	0.15	0.37	245.1	117.3	0.55	0.02*	0.14	0.37
GEMMA2-9B	83.6	126.2	0.39	-0.03*	0.65	0.55	268.3	97.8	0.44	0.00	0.53	0.50
LLAMMA3.1-8B	73.4	92.7	0.48	-0.05*	0.34	0.44	236.2	81.7	0.49	-0.00	0.27	0.41
GPT4O-MINI	82.6	82.6	0.44	-0.00	0.53	0.54	253.8	159.7	0.42	-0.00	0.48	0.49
OSS-20B	74.2	97.5	0.31	-0.04*	0.75	0.64	236.2	107.1	0.34	-0.00	0.68	0.59
QWEN3-32B	73.1	81.6	0.32	-0.08*	0.76	0.65	229.3	92.1	0.35	-0.01	0.66	0.58
OSS-120B	72.1	110.2	0.30	-0.03*	**0.793**	0.66	236.0	143.4	0.31	-0.00	0.74	0.64
In-Domain Evaluation	**Story Trained**						**Review Trained**					
Instruction Tune												
MODERNBERT-D-28M	82.2	402.7	.43	-0.04*	0.43	.49	285.7	140.7	0.35	-0.02	0.62	0.59
LLAMMA3.2-3B	80.7	75.4	0.50	-0.03*	0.23	0.44	219.5	81.6	0.45	0.04*	0.41	0.50
GEMMA2-2B	77.9	75.5	0.43	-0.03*	0.44	0.49	276.1	88.5	0.43	-0.01	0.45	0.53
Token Instruction Tune												
MODERNBERT-D-28M	79.1	318.0	0.35	-0.17*	0.57	0.54	266.3	207.8	0.43	-0.07*	0.42	0.41
LLAMMA3.2-3B	80.3	77.6	0.28	**-0.00**	0.71	0.66	217.8	87.2	0.32	0.04*	0.64	0.62
GEMMA2-2B	77.9	72.1	0.21	-0.03*	0.74	0.75	285.5	108.4	0.30	-0.01	0.64	0.67
Steering												
MODERNBERT-D-28M	82.6	108.2	0.55	-0.30*	0.04	0.31	241.8	186.2	0.54	-0.00*	0.095	0.34
LLAMMA3.2-3B	78.3	221.3	0.54	-0.10*	0.08	0.32	255.1	186.2	0.54	-0.00*	0.095	0.34
GEMMA2-2B	85.3	135.0	0.56	-0.01	0.18	0.39	229.6	124.0	0.54	0.03*	0.17	0.36
SENSESHIFT												
MODERNBERT-E-0.15B	78.3	91.6	0.26	-0.00	0.77	0.69	260.3	55.3	0.29	0.06*	0.71	0.66
MODERNBERT-E-0.4B	81.7	**53.6**	**0.20**	-0.00	0.790	**0.78**	261.5	57.5	0.26	0.04*	0.71	0.69
Out of Domain Evaluation	**Review Trained**						**Story Trained**					
Instruction Tune												
MODERNBERT-D-28M	81.0	366.0	0.35	-0.24*	0.61	0.57	251.5	614.1	0.49	-0.03*	0.30	0.42
LLAMMA3.2-3B	81.2	86.6	0.48	-0.04*	0.35	0.46	281.5	80.2	0.49	0.07*	0.27	0.43
GEMMA2-2B	80.0	113.2	0.42	-0.06*	0.45	0.49	269.5	75.5	0.43	-0.03*	0.44	0.49
Token Instruction Tune												
MODERNBERT-D-28M	88.0	186.1	0.39	-0.27*	0.52	0.49	277.8	667.6	0.39	-0.08*	0.52	0.49
LLAMMA3.2-3B	78.1	82.8	0.40	-0.06*	0.50	0.51	276.6	62.8	0.31	**0.08***	0.66	0.64
GEMMA2-2B	76.3	89.8	0.34	-0.08*	0.59	0.60	229.3	64.9	0.23	0.01	0.73	0.73
Steering												
MODERNBERT-D-28M	83.4	133.3	0.56	-0.30*	-0.00	0.30	237.4	127.7	0.55	-0.15	0.01	0.28
LLAMMA3.2-3B	84.2	153.1	0.53	-0.09*	0.12	0.34	235.1	183.6	0.53	-0.00	0.125	0.36
GEMMA2-2B	82.1	138.9	0.55	-0.01	0.17	0.39	261.9	125.0	0.56	0.03*	0.12	0.35
SENSESHIFT												
MODERNBERT-E-0.15B	87.1	65.6	0.38	-0.00	0.58	0.57	293.6	63.1	0.28	0.07*	0.73	0.69
MODERNBERT-E-0.4B	77.2	83.0	0.34	-0.00	0.62	0.61	260.3	**31.4**	**0.15**	0.07*	**0.85**	**0.85**

ation (Δ_s of 0.20 on Story), and the highest in-domain accuracy on Story (0.78). Crucially, MODERNBERT-E-0.4B maintains Δ_f at or near zero across both domains advocating contextual fitness of the generated sentences. MODERNBERT-E-0.15B already delivers strong results relative to all baselines, indicating that SENSESHIFT scales favorably with model capacity. In the out-of-domain setting, SENSESHIFT remains competitive with, and on most metrics outperforms, the baselines trained on the matching domain. More importantly, SENSESHIFT is the only method to maintain near-zero Δ_f in both the in-domain and out-of-domain blocks, suggesting that encoder conditioning preserves contextual fit while controlling sentiment intensity robust to change of training source.

Sentiment Following Analysis. Next we analyze sentiment following capabilities of competing baselines by examining how Δ_s is influenced by: (1) target sentiment specified by the user, (2) relative sentence position $\left(\frac{\text{sentence index}}{\text{total number of sentences}}\right)$, and (3) generation length as words: short (< 5), normal ($5 \leq l < 10$), long ($10 \leq l < 20$), and very-long (≥ 20). We compare MODERNBERT-E-0.4B with competitive baselines namely prompting OSS-120B and token instruction tuning of LLAMMA3.2-3B, and GEMMA2-2B.

Figure 2 visualizes these analysis. On left, all models demonstrate higher Δ_s when targeting strongly

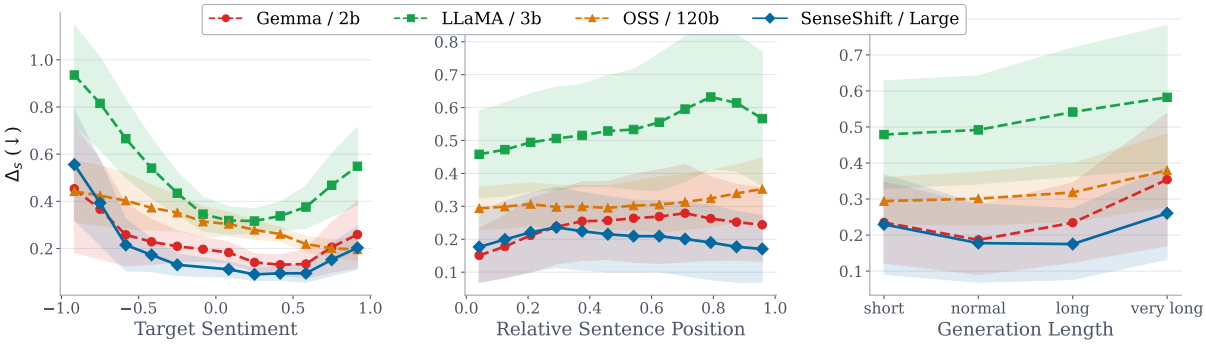


Figure 2: Δ_s across three evaluation dimensions. Shaded regions indicate variance. *Left:* Δ_s vs. target sentiment. *Center:* Δ_s vs. relative sentence position. *Right:* Δ_s vs. number of masked tokens.

negative sentiment values, whereas performance improves then slightly deteriorates toward highly positive targets. Exceptionally, OSS-120B demonstrates a monotonic improvement as the target sentiment becomes more positive while performing slightly better than MODERNBERT-E-0.4B at extreme sentiments. LLAMMA3.2-3B, GEMMA2-2B and MODERNBERT-E-0.4B all exhibit their lowest Δ_s around neutral zone, showing difficulty in producing highly polarized outputs despite strong overall controllability. We attribute this behavior to sentiment distribution of the training corpus for trained models. The story dataset sentence sentiments are heavily skewed toward neutral and positive sentiment, which likely biases models toward conservative sentiment shifts (see Table 3).

With respect to sentence position, all decoder models exhibit a mild increase in Δ_s when the target sentence occurs later in the story. This suggests that models benefit less from richer preceding context when performing sentiment-controlled generation. This behavior is particularly different for MODERNBERT-E-0.4B, which maintain the least Δ_s, however exhibits higher Δ_s on earlier sentences but improves steadily toward later positions. We hypothesize that encoder-based architectures rely more heavily on contextual buildup, making early-sentence editing inherently more challenging due to limited narrative grounding.

Finally, we analyze the effect of generation length. LLAMMA3.2-3B and OSS-120B show increased Δ_s as generation length grows, while MODERNBERT-E-0.4B overall has the least Δ_s and again demonstrates a different trend: Δ_s decreases for normal and long generations before slightly increasing again. GEMMA2-2B also shows improved control for normal length generations before degrading on longer sentences.[8] In summary, our analysis suggests that SENSESHIFT maintains a robust Δ_s advantage over baselines independent of target sentiment, sentence position, and generation length, indicating that the gap reflects a genuine difficulty in maintaining sentiment coherence rather than an artifact of any single evaluation condition.

Table 2: Users preference (%) of sentences generated by MODERNBERT-E-0.4B-story vs GEMMA2-2B trained with Token-Instruction-Tuning. Rows correspond to evaluation criteria: contextual fitness, grammatical correctness, and sentiment following.

Criterion	GEMMA2-2B	SENSESHIFT	Both	Neither
Fitness	22.1	23.0	46.9	8.0
Fluency	6.2	6.2	85.0	2.7
Sentiment	28.3	31.0	28.3	12.4
Overall	18.9	20.1	53.4	7.7

Human Evaluation. Table 2 summarizes human preference results across three criteria: contextual fit, grammatical fluency, and sentiment following. Notably, MODERNBERT-E-0.4B achieves these results with only 395M parameters, compared to GEMMA2-2B (2B parameters) a 5× reduction in parameters (described in Section4.2).

On fluency and contextual fit, both models perform comparably. Annotators rated the outputs as equally fluent in 85.0% of samples and equally well-fitting to the surrounding context in 46.9% of cases, indicating that sentiment control does not substantially degrade linguistic quality. Among the remaining contextual-fit judgments, MODERNBERT-E-0.4B is preferred at a slightly higher rate to GEMMA2-2B (23.0% vs. 22.1%), showing that the smaller encoder-based model matches a substantially larger generative model on contextual

[8] Joint analysis of SENSESHIFT is provided in Section A.5.

integration.

Differences are more pronounced for sentiment following. Annotators judged MODERNBERT-E-0.4B sentiment satisfactory in 31.0% of examples versus 28.3% for GEMMA2-2B, selected both models in 28.3%, and preferred neither model in 12.4% of cases. This indicates that MODERNBERT-E-0.4B achieves stronger sentiment alignment while preserving contextual coherence.

Overall, annotators rated MODERNBERT-E-0.4B as equivalent to or better than GEMMA2-2B in 73.5% of evaluated samples. Achieving this with a fraction of the parameters suggests that encoder-centric approaches are a strong and competitive alternative for fine-grained sentiment-aware controllable text generation, particularly in settings requiring simultaneous preservation of context and precise sentiment manipulation.

6 Conclusion

This paper introduced SENSESHIFT, a lightweight generative framework base on encoder language models for fine-grained sentiment-aware in-text generation. SENSESHIFT utilizes quantized sentence level sentiment token, to condition mask prediction of encoder models. Using our modfied sequential mask prediction strategy SENSESHIFT is able to produce coherent text following the fine-grained sentiment signal. Through extensive in-domain and out-of-domain evaluation on story and review generation of two encoder models in two sizes we demonstrate that SENSESHIFT can generate sentences at various positions using fine-grained sentiment control signals, while preserving overall fluency, contextual fitness, and sentiment adherency compare to fine-tuned and prompting autoregressive models. Furthermore, our results indicate that MODERNBERT-E-0.4B performs robust in out-of-domain scenarios, allowing stable performance for in-text generation task.

7 Acknowledgments

This research was funded in whole or in part by the Austrian Science Fund (FWF): COE12, DFH23, P36413; and by the State of Upper Austria and the Federal Ministry of Education, Science, and Research: LIT-2024-13-SEE-111.

8 Limitation and Ethical Considerations

Sentiment Analysis. Our framework relies on VADER (Hutto and Gilbert, 2014) for all sentiment scoring, both during training signal construction and at inference time, where sentiment must be evaluated repeatedly over the full text to guide and assess each rewrite. This design choice scopes our work and priorities speed. Naturally, more accurate neural sentiment models (e.g., RoBERTa (Barbieri et al., 2020)) would substantially increase latency per generation. VADER's deterministic nature applies a uniform limitation on all baselines both decoder and SENSESHIFT where, contextually nuanced sentiment might not be captures accurately. Hence, reported Δ_s values should be interpreted with this constraint in mind. We assume that having access to human level sentence sentiment or replacing VADER with a lightweight but more accurate sentiment scorer can boost all baselines including SENSESHIFT performance and is a promising direction for future work.

Furthermore, sentiment is an inherently subjective construct: human annotators frequently disagree on the polarity and intensity of the same sentence, which places an upper bound on how precisely any model including SENSESHIFT can be evaluated against a "true" sentiment label.

Training Corpora and Sentiment Distribution. Our models are trained on TinyStories and a review corpus, both of which are narrow in domain. The choice of dataset stems from the task which supposedly requires changing dynamics of the sentiment (partly negative, partly positive) which can be found in long text generation. These constraints prevent us from making strong claims about true generalization to unseen domains such as news, social media, or formal writing. A further challenge is the sentiment distribution of both corpora: Extreme sentiment values appear very infrequently, meaning the model sees comparatively few training examples of strongly positive or strongly negative targets, with negative ones being significantly less observed. This imbalance makes fine-grained control at the tails of the sentiment range harder, and likely contributes to the elevated Δ_s we observe at $s = -1.0$. We expect that curating higher-quality, sentiment-balanced training data, particularly with more uniform coverage of extreme values would meaningfully improve performance, and we leave this exploration to future work.

During editing we observed that the errors of the model with respect to target sentiment also is effected by the sentiment of the sentences surrounding it where the more positive the context surrounding , the more edits models required to change the sentiment to opposite polarity. Due to the constrained scope we did not delve into this effect and leave this analysis to future works.

Baseline Selection and Compute Constraints. Baselines were selected to enable quantifiable, reproducible comparison across dominnant controlling methods namely, prompting, instruction tuning, and activation steering paradigms with a special version of instruction tuning with our sentence level quantized tokenization of sentiment. The choice of baselines was based on compatibility with our method, and comparison between decoder and encoder model architectures. The specific models chosen reflect resource constraints: All experiments were conducted under a fixed compute budget, which precluded evaluation of the largest available open-weight models. Results for prompting-based baselines should therefore be read as representative rather than exhaustive, and stronger frontier models may narrow the observed performance gap.

Our model is designed to edit text toward an arbitrary target sentiment intensity, including strongly negative values. This capability carries clear misuse potential. Because edits are designed to be fluent and contextually coherent, modified outputs may be difficult to distinguish from originals without provenance tracking. We strongly recommend that any deployment of sentiment controllable systems be accompanied by appropriate safeguards, including use-case restrictions, output disclosure to end users, and, where feasible, watermarking or audit logging. Responsible use of this technology is the obligation of both developers and practitioners who build upon it.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Jiyu Chen, Sarvnaz Karimi, Diego Molla, and Cecile Paris. 2025. To labor is not to suffer: Exploration of polarity association bias in LLMs for sentiment analysis. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 70–78, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. MacLaSa: Multi-aspect controllable text generation via efficient sampling from compact latent space. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4424–4436, Singapore. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1587–1596. JMLR.org.

C. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Parag Jain, Abhijit Mishra, Amar Prakash Azad, and Karthik Sankaranarayanan. 2019. Unsupervised Controllable Text Formalization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6554–6561.

Sara Kangaslahti and David Alvarez-Melis. 2024. Continuous Language Model Interpolation for Dynamic and Controllable Text Generation. *arXiv preprint*. ArXiv:2404.07117 [cs].

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. 2024. Style Vectors for Steering Generative Large Language Models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 782–802, St. Julian's, Malta. Association for Computational Linguistics.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. 2025. Bitune: Leveraging bidirectional attention to improve decoder-only LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9510–9536, Suzhou, China. Association for Computational Linguistics.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. LLMs Get Lost In Multi-Turn Conversation. *arXiv preprint*. ArXiv:2505.06120 [cs].

Arya Labroo, Ivaxi Sheth, Vyas Raina, Amaani Ahmed, and Mario Fritz. 2026. Funny or persuasive, but not both: Evaluating fine-grained multi-concept control in LLMs. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 522–554, Rabat, Morocco. Association for Computational Linguistics.

Xun Liang, Hanyu Wang, Shichao Song, Mengting Hu, Xunzhi Wang, Zhiyu Li, Feiyu Xiong, and Bo Tang. 2024. Controlled Text Generation for Large Language Model with Dynamic Attribute Graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5797–5814, Bangkok, Thailand. Association for Computational Linguistics.

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)*, pages 6691–6706, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Michela Lorandi and Anya Belz. 2023. How to control sentiment in text generation: A survey of the state-of-the-art in sentiment-control techniques. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 341–353, Toronto, Canada. Association for Computational Linguistics.

Albert Lu, Hongxin Zhang, Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2023. Bounding the Capabilities of Large Language Models in Open Text Generation with Prompt Constraints. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1982–2008, Dubrovnik, Croatia. Association for Computational Linguistics.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. QUARK: Controllable Text Generation with Reinforced Unlearning. *Advances in Neural Information Processing Systems*, 35:27591–27609.

Fuli Luo, Damai Dai, Pengcheng Yang, Tianyu Liu, Baobao Chang, Zhifang Sui, and Xu Sun. 2019a. Learning to control the fine-grained sentiment for story ending generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6020–6026, Florence, Italy. Association for Computational Linguistics.

Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019b. Towards Fine-grained Text Sentiment Transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2013–2022, Florence, Italy. Association for Computational Linguistics.

Shahed Masoudian, Cornelia Volaucnik, Markus Schedl, and Navid Rekabsaz. 2024. Effective controllable bias mitigation for classification and retrieval using gate adapters. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2434–2453, St. Julian's, Malta. Association for Computational Linguistics.

Gilles Nawezi, Lucie Flek, and Charles Welch. 2023. Style Locality for Controllable Generation with kNN Language Models. In *Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!*, pages 68–75, Prague, Czech Republic. Association for Computational Linguistics.

Damian Pascual, Beni Egressy, Clara Meister, Ryan Cotterell, and Roger Wattenhofer. 2021. A plug-and-play method for controlled text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3973–3997, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards Controllable Story Generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49, New Orleans, Louisiana. Association for Computational Linguistics.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable Natural Language Generation with Contrastive Prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics.

Lin Qiao, Jianhao Yan, Fandong Meng, Zhendong Yang, and Jie Zhou. 2020. A Sentiment-Controllable Topic-to-Essay Generator with Topic Knowledge Graph. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3336–3344, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-

networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Vinay Samuel, Harshita Diddee, Yiming Zhang, and Daphne Ippolito. 2025. CIE: Controlling Language Model Text Generations Using Continuous Signals. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3815–3825, Suzhou, China. Association for Computational Linguistics.

Yizhan Shao, Tong Shao, Minghao Wang, Peng Wang, and Jie Gao. 2021. A Sentiment and Style Controllable Approach for Chinese Poetry Generation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pages 4784–4788, New York, NY, USA. Association for Computing Machinery.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023. Evaluating Large Language Models on Controlled Generation Tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168, Singapore. Association for Computational Linguistics.

Yu-Siou Tang and Chung-Hsien Wu. 2021. Latent Attribute Control for Story Generation. In *2021 International Conference on Asian Language Processing (IALP)*, pages 148–153.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasaath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Xinpeng Wang, Han Jiang, Zhihua Wei, and Shanlin Zhou. 2022. CHAE: Fine-Grained Controllable Story Generation with Characters, Actions and Emotions. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6426–6435, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, and 1 others. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

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*, pages 3511–3535, Online. Association for Computational Linguistics.

Nai-Chi Yang, Wei-Yun Ma, and Pu-Jen Cheng. 2024. Plug-in Language Model: Controlling Text Generation with a Simple Regression Model. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2165–2181, Mexico City, Mexico. Association for Computational Linguistics.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2022. Hierarchical template transformer for fine-grained sentiment controllable generation. *Inf. Process. Manage.*, 59(5).

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2023. A survey of controllable text generation using transformer-based pre-trained language models. *ACM Comput. Surv.*, 56(3).

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Chujie Zheng, Pei Ke, Zheng Zhang, and Minlie Huang. 2023. Click: Controllable text generation with sequence likelihood contrastive learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1022–1040, Toronto, Canada. Association for Computational Linguistics.

Paulina Aleksandra Żal, Guang Lu, and Nianlong Gu. 2024. Sentiment- and Keyword-Controllable Text Generation in German with Pre-trained Language Models. In *Proceedings of the 9th edition of the Swiss Text Analytics Conference*, pages 68–88, Chur, Switzerland. Association for Computational Linguistics.

A Appendix

A.1 Problem Statement and Motivation

Further to related research supporting LLMs challenges of controllability through prompting we also did a small experiment to further show that models are incapable of following prompt-based

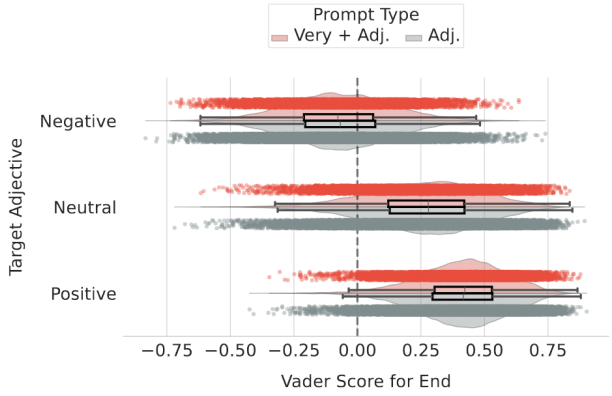


Figure 3: Sentiment of story ending written by GPT4o-MINI for Negative, Positive, and Neutral ending with and without intensifying adverb "Very" in the prompt.

controlling strategies. We directed GPT4o-MINI model to provide stories with three distinct endings namely Positive, Negative, and Neutral with-/without intensifying adverb "Very". We passed the endings provided by GPT4o-MINI to VADER sentiment analyzer to score the sentiment of the stories. Figure 3 shows the result of this experiment. As it can be seen from the figure there are three major problems with the provided endings: (1) weak separation between adjective and intensified-adjective conditions, indicating limited intensity control (Samuel et al., 2025); (2) a distributional shift toward more positive sentiment, even under negative targets (Chen et al., 2025); and (3) compressed score ranges that rarely reach sentiment extremes. These findings are consistent with earlier reports that LLMs struggle with fine-grained controllability and strict prompt constraints (Sun et al., 2023; Lu et al., 2023). Together, this motivates moving beyond prompt-only control and focus on architectural changes and controllable frameworks.

A.2 Baselines

In This section we provide more technical detail on our baselines namely, Prompting, Instruction Tuning, Token Instruction Tuning and Steering Activation.

A.2.1 Prompts

When evaluating decoder models the choice of prompt is of great importance, slight changes in the prompt might lead to variations in the outcome of the model which further motivates our work. However, In order to have a robust evaluation we modified and re-evaluated the decoder models iteratively to come-up with the best instructions that led to the most accurate outcome in both response of the model and accuracy in following the sentiment. In this section we provide specific prompts used for each of the evaluation. Note that prompting variation is only important when evaluating off-the-shelf models and for instruction tuning, token instruction tuning the choice of prompt is not strongly related as the model will be evaluated with the same prompt to ensure a fair comparison with SENSESHIFT.

Prompting: For prompting we observed that masking the target sentence and asking to generate a sentence that fits best to that location is not performing accurately hence the prompt provides the full text while asking model to replace the existing sentence in text with a new sentence of another sentiment. Below we can find the specific prompt for every setup.

Prompting

Consider that every sentence has a sentiment where -1 is the most negative and 1 is the most positive sentiment. Given this text:
`{FULL_TEXT}`
Rewrite TARGET_SENTENCE to have a sentiment value of TARGET_SENTIMENT but still make sense to overall text. Write only the sentence and nothing else.

Instruction Tuning: For instruction tuning to work the model requires the sentence to be predicted hence having the sentence in the text was not desirable and considered as leakage of information, Hence we modified the instruction and included [MISSING_SENTENCE] as masked version of the sentence and asked the models to write the missing sentence to have the specific value.

Instruction Tuning

Consider that every sentence has a sentiment where -1 is the most negative and 1 is the most positive sentiment. Given this text:
`{MASKED_TEXT}`
Rewrite [MISSING_SENTENCE] to have a sentiment value of TARGET_SENTIMENT but still make sense to overall text. Write only the sentence and nothing else.

Token Instruction Tuning: The difference between this variation and the normal instruction tuning is the introduction of new controlling tokens to

the model just as SENSESHIFT is trained on. We equiped decoder models with SENSESHIFT sentiment tokens $[\sigma_i]$ as special untrainable tokens as well as added [MISSING_SENTENCE] as a new special token totalling 22 new tokens. in this version of the prompt each sentence has the control signals attached at the beginning of every sentence to give them context just like SENSESHIFT:

Token Instruction Tuning

Consider that every sentence has a sentiment where -1 is the most negative and 1 is the most positive sentiment. Given this text:
`{MASKED_TEXT_WITH_SENTIMENT_TOKENS}`
Rewrite `{[MISSING_SENTENCE]}` to have a sentiment value of $\{[\sigma_i]\}$ but still make sense to overall text. Write only the sentence and nothing else. $\{[\sigma_i]\}$

A.2.2 Activation Steering

Over the past few years, activation steering has emerged as an effective technique for controlling generated text using contrastive samples. These samples are typically constructed from the original dataset as minimally differing pairs, where small, targeted changes (e.g., sentiment polarity) induce measurable shifts in model activations. Although this approach does not require additional model training, its effectiveness depends critically on the quality and construction of the contrastive pairs.

In our setup, we construct contrastive pairs by selecting sentences with extreme sentiment values (i.e., $|\sigma_{s_i}| > 0.9$). We then use the steering vector library [9] to extract activation directions from these pairs. Here, "training" refers not to updating model parameters, but to computing steering vectors that capture the latent direction corresponding to the desired attribute shift. For each dataset, we sample $10k$ contrastive pairs from the pool of possible combinations to estimate these directions.

At inference time, the steering strength is controlled via a scaling factor (the multiplier), which determines the extent to which the model output is shifted toward the target attribute. While the original method is primarily designed for categorical control, it also suggests the potential for continuous modulation. Following this insight, we adapt the multiplier dynamically based on the target sentiment value, enabling more fine-grained, continuous control over the generated text.

[9] https://github.com/steering-vectors/steering-vectors

Table 3: Dataset statistics for the train splits. Sentiment scores are in $[-1, 1]$.

Statistics	TinyStories	Yelp
Num Samples	4439352	252085
Positive (>0.2)	53.7%	41.0%
Negative (<-0.2)	0.3%	4.3%
Neutral	46.0%	54.8%
Text		
Avg Sentences	14.2 ± 7.77	6.4 ± 4.86
Median Sentences	13.	5
Min/Max Sentences	2 / 195	2 / 86
Avg Words	152.90 ± 53.52	90.28 ± 80.34
Median Words	143.00	68.00
Min/Max Words	9 / 902	1 / 1004
Avg VADER	0.21 ± 0.14	0.15 ± 0.20
Median VADER	0.21	0.14
Min/Max VADER	-0.68 / 0.92	-0.91 / 0.96
Avg RoBERTa	0.27 ± 0.22	-0.0206 ± 0.64
Sentences		
Avg Words	10.7 ± 5.36	13.9 ± 9.61
Median Words	10	12
Min/Max Words	1 / 702	1 / 550
Avg VADER sent.	0.1978 ± 0.38	0.13 ± 0.37
Min/Max VADER	-1.0 / 1.0	-0.99 / 0.99
Avg RoBERTa sent.	0.25 / 0.52	-0.078 / 0.94
Min/Max RoBERTa	-1.0 / 1.0	-0.99 / 0.99

A.2.3 Dataset Statistics and Training Hyper-Parameters

During pre-processing we removed major samples in the dataset which were not suitable for our task (e.g., reviews with one sentence). In Table 4 we report the statistics of the training data and hyperparameters used for model training. For our models we have not used any PEFT method and all models' full number of parameters are tuned for the task to ensure that none of the results are effected by the number of parameters that are trained.

A.3 Iterative Infilling Versus One Shot Prediction

Mask prediction was not used for long text generation due to their one-shot prediction of all masks which result in several repetitive high probability tokens appearing in different parts of the text without rational connection. Here we present an ablation study of removing infilling mask prediction from the model to show the effectiveness of the technique.

Table 5 compares our iterative mask infilling strat-

Table 4: Default hyperparameters used across different training paradigms for controllability. * value for Qwen is 1024 to ensure finalizing thinking process.

State	Hyperparameter	Prompt/Steer	Inst-Tune	Token-Inst-Tune	SENSESHIFT
Training	Learning rate (lr)	–	1e-5	1e-5	1e-5
	Batch size (bs)	–	32	32	50
	Dropout (dp)	–	0.0	0.0	0.0
	Warmup steps	–	500	500	500
	Epochs	–	30	30	50
	Weight decay	–	0.01	0.01	0.01
	Scheduler	–	Linear	Linear	Linear
	Regime	–	Causal	Causal	MLM
Generation	Temperature (T)	0.7	0.7	0.7	0.9
	Top-p	0.9	0.9	0.9	0.9
	Repetition penalty (rp)	1.1	1.1	1.1	-
	Max tokens	128*	128	128	30
	Num Beams	2	2	2	2

Table 5: Comparison between Iterative Mask Infilling and Whole Mask prediction

Model	Train	Story Generation					Review Generation				
		Ppl. ↓	Δ_s ↓	Δ_f ↑	Corr. ↑	Acc. ↑	Ppl. ↓	Δ_s ↓	Δ_f ↑	Corr. ↑	Acc. ↑
Whole Mask Prediction											
MODERNBERT-E-0.15B	Story	596.8	0.42	-0.01	0.54	0.44	303.1	0.44	-0.02*	0.48	0.38
	Review	482.9	0.50	-0.04*	0.27	0.31	325.2	0.43	-0.01	0.55	0.44
MODERNBERT-E-0.4B	Story	869.9	0.4	-0.039*	0.55	0.5	571.3	0.43	0.01	0.48	0.39
	Review	426.3	0.50	-0.041*	0.29	0.32	288.8	0.44	-0.013	0.51	0.41
Iterative Mask infilling											
MODERNBERT-E-0.15B	Story	91.6	0.26	-0.00	0.77	0.69	63.1	0.28	**0.07***	0.73	0.69
	Review	65.6	0.38	-0.00	0.58	0.57	55.3	0.29	0.06*	0.71	0.66
MODERNBERT-E-0.4B	Story	**53.6**	**0.20**	-0.00	**0.79**	**0.78**	**31.4**	**0.15**	0.07*	**0.85**	**0.85**
	Review	83.0	0.34	**-0.00**	0.62	0.61	57.5	0.26	0.04*	0.71	0.69

egy against a whole mask prediction using SENSESHIFT, in which all masked tokens are predicted in a single forward pass. The results strongly favour the iterative approach across every metric and configuration. Perplexity improvements are the most striking: iterative infilling reduces PPL by a factor of 5–10× across all model sizes and training domains (e.g., MODERNBERT-E-0.4B Story drops from 869.9 to 53.6), indicating that generating masked spans token-by-token produces substantially more fluent text than predicting the entire span at once. Sentiment control follows the same pattern: Δ_s is consistently lower under iterative infilling, with reductions of 0.15–0.25 across configurations, while Δ_f values remain near zero compared to the small but consistent negative drift observed under whole mask prediction. These results suggest that whole mask prediction suffers from exposure bias where the model must generate a coherent span without access to its own intermediate outputs whereas iterative infilling conditions each token on the previously generated context, allowing sentiment and fluency constraints to be satisfied progressively rather than in a single unconstrained step giving advantage of decoder models to encoder ones.

A.4 Human Evaluation Protocol

To ensure that our results are comparable with human understanding of fitness and sentiment we compared SENSESHIFT rewrites with parallel rewrites of closest comparable competing decoder-based model namely GEMMA2-2B. Both models are trained on story and GEMMA2-2B is selected for human evaluation because of its competetive performance in automatic evaluation. Furthermore GEMMA2-2B is the closest comparable baseline when it comes to evaluation while it goes through the same sentiment token training as MODERNBERT-E-0.4B and in contrast to prompting method which the original sentence is visible to the model GEMMA2-2B does not have access to the original sentence.

Our human evaluation is a user preference study where users are asked to select which model they think generated a higher quality sentence given a document. We developed a simple annotation plat-

form and selected 100 samples from stories (n=55) and reviews (n=45) where models both models rewrite for the exact same sentence and the exact same sentiment. We shared the annotation link with 8 annotators volunteered to give their preference on the rewrites. We did not collect their demographic information nor collected any other sensitive information rather than labels from the annotators. We provided them full document context (review or story), the original sentence which was targeted for replacement, and two anonymized candidate generated sentences which one produced by GEMMA2-9B trained with token instruction tuning and one by MODERNBERT-E-0.4B. We display the sentences in randomized order to control for position bias of annotators and also to protect model identities. For each pair, annotators respond to three questions:

- **Contextual Fit.** Which rewrite fits better into the surrounding text? Judge whether it matches the tone, vocabulary, register, and stylistic quirks of the full passage. If the original writing is formal, poetic, blunt, or otherwise distinctive, a good rewrite preserves those features and sits well inside the position to produce a coherent text.
- **Grammar and Fluency.** Ignoring the surrounding text, which sentence is more grammatically correct and naturally written as a standalone sentence? Annotators focus purely on syntax, word choice, and fluency.
- **Sentiment Match.** Which rewrite better achieves the target sentiment, specified on a scale from -1 (very negative) to $+1$ (very positive)? Annotators judge how well each sentence's emotional tone matches the indicated target value.

For each criterion, annotators select one of four options: **A**, **B**, **Both** (both outputs are equally good), or **Neither** (neither output is satisfactory) where A and B represent GEMMA2-2B and MODERNBERT-E-0.4B-story in random order. We report average preference of annotators in Table 2 across the three quality criteria. Tables 6 and 7 report the results in detail for each domain.

A.5 Joint Analysis of Δ_s

Figure 4 extends the univariate analysis of Section 5 by examining Δ_s across pairs of evaluation dimensions for the *large-story* model.

Table 6: Human preference evaluation on the **Story** domain (n=55 sentence pairs). The specific model used for evaluation is MODERNBERT-E-0.4B trained on stories. Similarity to Original is reported as a diagnostic metric and excluded from the Overall score.

Criterion	GEMMA2-2B	SENSESHIFT	Both	Neither
Fitness	17.2	23.4	54.7	4.7
Grammar	4.7	4.7	89.1	1.6
Sentiment	29.7	28.1	31.2	10.9
Overall	17.2	18.8	58.3	5.7

Table 7: Human preference evaluation on the **Review** domain (OOD n=45 sentence pairs). The specific model used for evaluation is MODERNBERT-E-0.4B trained on stories. Similarity to Original is reported as a diagnostic metric and excluded from the Overall score.

Criterion	GEMMA2-2B	SENSESHIFT	Both	Neither
Fitness	27.7	23.4	36.2	12.8
Grammar	8.5	8.5	78.7	4.3
Sentiment	25.5	36.2	23.4	14.9
Overall	20.6	22.7	46.1	10.6

Strongly negative target sentiment ($s{=}-1.0$) dominates error across all positions, peaking at $\Delta_s{=}0.66$ at late document positions where accumulated context compounds the difficulty. Across non-extreme sentiment values, sentence position has comparatively little effect, suggesting position-dependent degradation is largely confined to the sentiment extremes.

The interaction between large mask counts and strongly negative targets is the most pronounced pattern across all three heatmaps, reaching a global maximum of $\Delta_s{=}0.72$. At neutral-to-positive targets, increasing mask size has only a modest effect, indicating that additional generation freedom is primarily problematic when the target register is already difficult to maintain.

Error increases moderately toward late positions and large mask counts, though the interaction between the two is approximately additive. Notably, mid-document positions (0.44–0.69) yield the lowest Δ_s, suggesting that a moderate amount of preceding context aids sentiment following more than editing at the document start.

A.6 Qualitative Examples

To demonstrate the model's capabilities, we provide a demo program that allows users to inspect model behavior on arbitrary input text. Figure 5 visualizes the interactive demo. We present qualita-

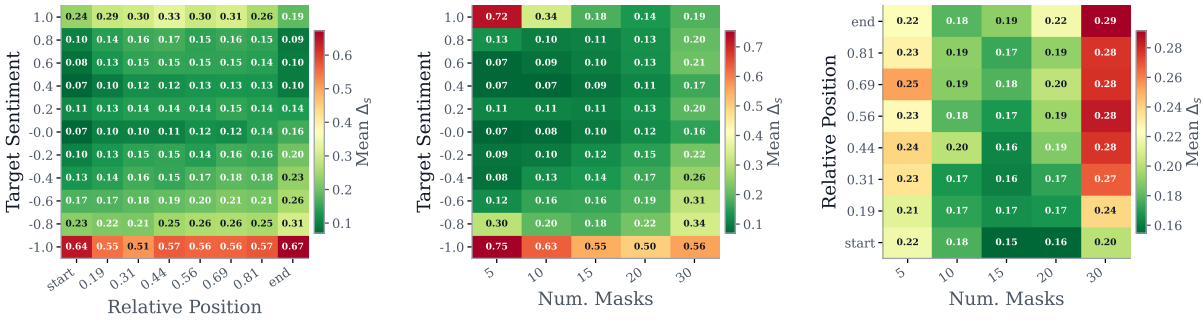


Figure 4: Mean Δ_s as a joint function for the MODERNBERT-E-0.4B-story. *Left:* Target sentiment vs. relative sentence position. *Center:* Target sentiment vs. number of masked tokens. *Right:* Relative sentence position vs. number of masked tokens. Lower values (green) indicate better sentiment control.

tive examples of iterative sentiment editing applied to a story (in-distribution) and a hotel review (out-of-distribution), both edited using MODERNBERT-E-0.4B trained on the story dataset.

Figure 6 illustrates the story editing process using our custom visualization tool. Figure 6a shows the original sentiment trajectory, revealing a mix of positive, neutral, and negative tones that makes fine-grained control challenging. In Figure 6b, we shift the story toward a more negative tone without altering its core theme where a boy, a fish, and his mother contribute while transforming a harmless event into one involving a dead fish and an angry mother. This required 20 edits across all sentences. We observed stronger negative transformations sometimes necessitate modifying surrounding context, as locally extreme changes can conflict with the overall tone of prior or posterior sentences. Figure 6c shows the opposite transformation toward a more positive tone, requiring only 12 edits across 7 of 8 sentences, suggesting that positive sentiment shifts are more efficient when the original context already leans positive.

Figure 7 presents the same editing process applied to a hotel review as an out-of-distribution test. Figures 7 (a), (b), and (c) show the original review, negative arc shift (8 edits), and positive arc shift (6 edits) respectively. MODERNBERT-E-0.4B demonstrate strong OOD capability in review domain despite being trained exclusively on stories, maintaining coherent and fluent edits across both sentiment directions.

Overall, these examples demonstrate that MODERNBERT-E-0.4B achieves fine-grained controllable sentiment editing while preserving narrative coherence, with the number of required edits reflecting the asymmetry between negative and positive sentiment shifts observed in the quantitative results.

A.7 Usage of AI

We used Claude to assist with plotting and writing the HTML code for the user-study interface. We also used OpenAI models and Claude to revise the authors' original writing for grammar and clarity.

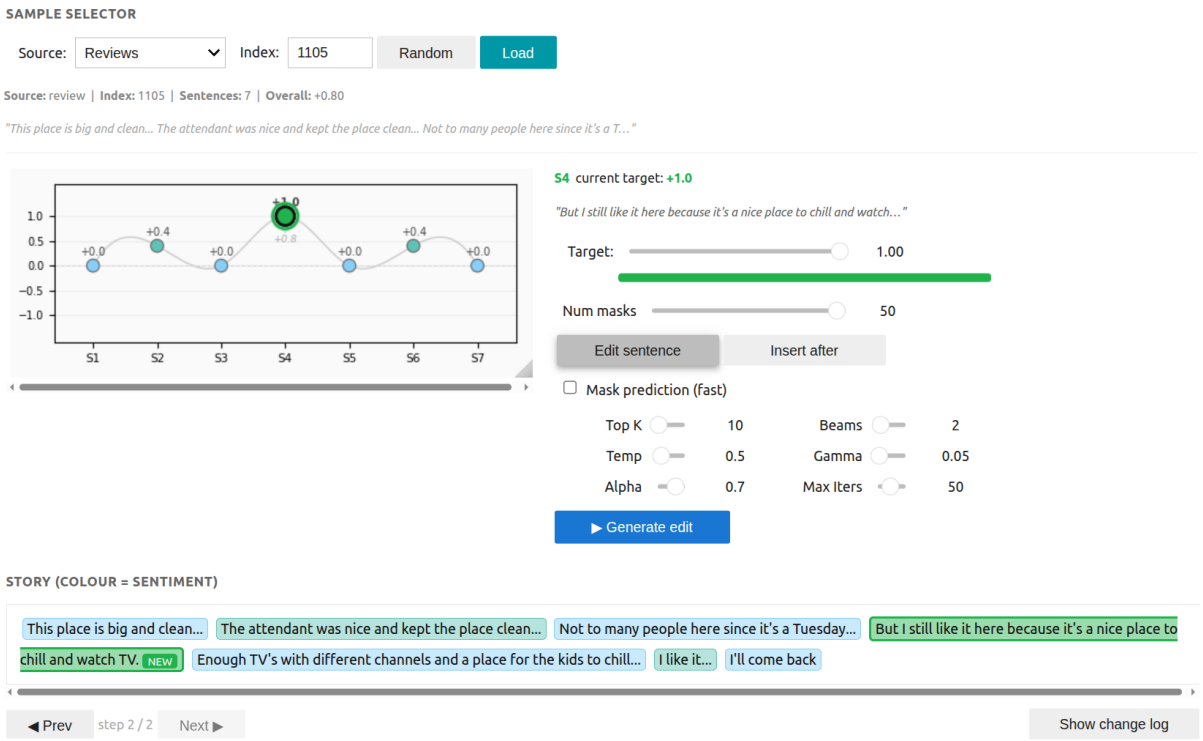
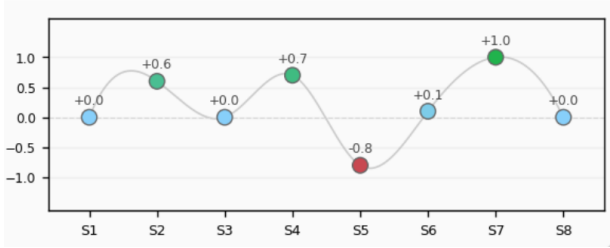
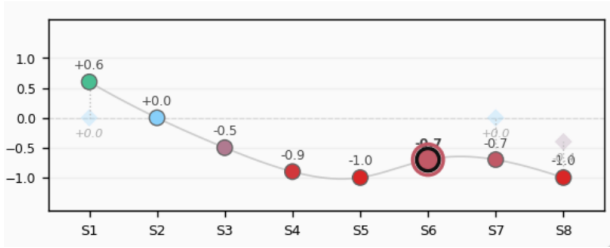


Figure 5: Interactive demo of SENSESHIFT. Users can choose or randomly select data from both dataset and edit different sentences with their desired sentiment intensity



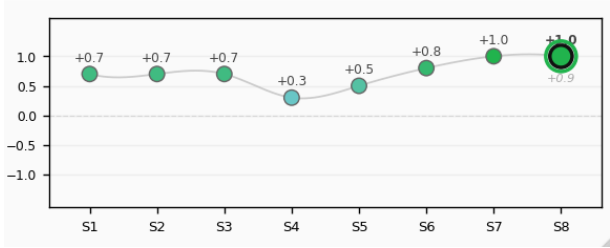
Once upon a time, there was a little boy named Timmy. | Timmy loved to go fishing with his grandpa. | One day, they went to the lake to catch some fish. | Timmy caught a big fish and was so happy. | But when they got home, Timmy's mom got mad because they made a big mess in the kitchen while cleaning the fish. | "Timmy, you need to clean up this messy kitchen!" she said. | Timmy replied, "But Mom, we had to perform surgery on the fish to get it ready for dinner!" His mom laughed and said, "Okay, but next time, let's do it outside." And they all had a good laugh together. | The end.

(a) Original story and its sentiment arc.



Once upon a time, there was a little name was Timmy and he lived with his family. | His dad and mom were always busy with work. | So one day, Timmy decided to do something bad. | He decided to kill his pet fish and he did it. | But then, Timmy's mom came home and saw the dead fish lying on the floor and got angry. | Why did you kill the fish? she asked. | Timmy cried. | Timmy felt sad and the fish was dead.

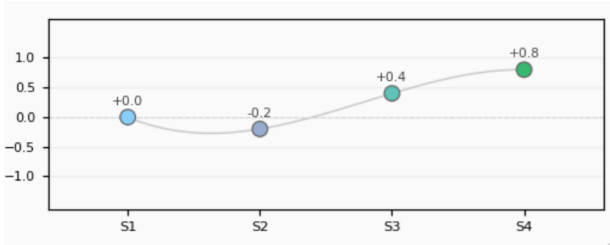
(b) Shifting sentiment arc toward negative with 20 edit. Less-hued numbers show actual post-edit sentiment; larger gaps indicate lower faithfulness.



Once there was a happy boy named Timmy. NEW | They loved playing outside. | One sunny day, they went fishing with their friends. | They caught a big fish and shared it. | After they ate, Timmy's mom came in and saw the yummy fish. | She was happy and told them how good it tasted. | Timmy replied, "But Mom, we had to perform surgery on the fish to get it ready for dinner!" His mom laughed and said, "Okay, but next time, let's do it outside." And they all had a good laugh together. | They were happy and enjoyed the yummy treat!

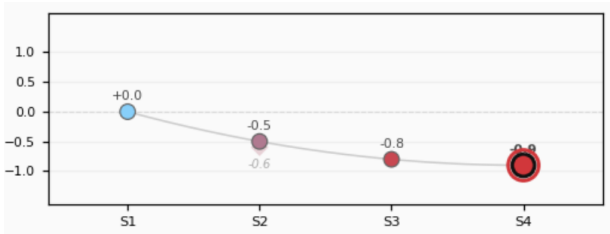
(c) Shift Arc toward positive with 12 edits. Positive shifts required fewer edits reflecting the sentiment asymmetry in the quantitative results.

Figure 6: Qualitative examples of iterative sentiment editing on a story (in-distribution). Each subfigure shows the sentiment arc (top) and annotated text (bottom).



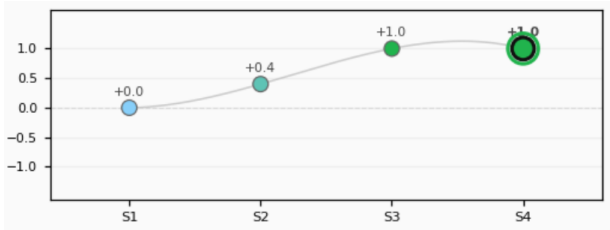
This is decent place for the money if you just need a place to sleep. | The room is a little small, no microwave, mini fridge or coffee maker. | I have stayed in nicer places that have all of these things for a little less. | I'd recommend looking around for the best deal with the amenities you want.

(a) Original review and its sentiment arc.



This is decent place for the money if you just need a place to sleep. | But if you need a room, a TV, and a pool, it's not so good. | The price is too high, and it's a bad place if you don't have good luck. | Poor people have to suffer in this bad place. NEW

(b) Shifting sentiment arc toward negative with 8 edits. Less-hued numbers show actual post-edit sentiment; larger gaps indicate lower faithfulness.



This is decent place for the money if you just need a place to sleep. | I recommend staying here because it has a cheap price and it has all the things you might need. | The place is nice and comfortable, and the people who help you are friendly and love to help. | I recommend this place to all my friends and they love it too, so it's super popular! NEW

(c) Shifting sentiment arc toward positive with 6 edits.

Figure 7: Qualitative examples of iterative sentiment editing on a hotel review (out-of-distribution). Each subfigure shows the sentiment arc (top) and annotated text (bottom).