CS590.06 Project Report – Sketch

Chuck Xiong & Qilin Ye

April 27, 2025

TODOs:

- *Add a TL;DR / opening phrase for each subsection in §3 — the whats, whys, and intuitions*
- *References*
- *Concluding remarks, if any*
- *And beamer of course (will do on Sunday)*

1 Introduction

Online review platforms (e.g. Yelp, Google Reviews, OpenTable) play an increasing central role in shaping consumer choice and business reputation. Machine-learning models trained naïvely on star ratings and raw text exhibit over-sensitivity to highly polarized words, amplifying certain voices while muting more nuanced perspectives. This not only undermines an ML model's predictive accuracy but also perpetuates unfair biases, as users who choose subtler language or more balanced phrasing are systematically under-valued. Addressing this gap is critical both for creating more reliable sentiment analysis systems and for ensuring that recommendation engines reflect a fair cross-section of user opinions.

To illustrate, consider the following two versions of the same café review:

(Subtly negative) The pasta **arrived lukewarm and a bit undercooked**. Our server **barely checked in after serving**. Overall, it was **disappointing and not worth a return**.

(Strongly worded) The pasta **was cold and almost raw, completely inedible**. Our server **ignored us**. Overall, it was a **dreadful experience I won't repeat**.

In the yelp_review_full dataset, each entry consists of a textual review similar to the one shown above, and an integer-valued star rating in $\{1, 2, 3, 4, 5\}$. The goal is to train a model that predicts a star-rating based on a review. A naïvely trained model might score the first review as a 2-star while the second as 1-star, solely due to the difference in the reviewers' writing styles, and in doing so a bias is induced.

In this project, we first quantify two concrete distortions under a standard training regime—an **asymmetric effect** (negative words pulling ratings down more than positives push them up) and a **saturation effect** (additional extreme descriptors ceasing to move the needle). From a causal standpoint, we treat highly polarizing tokens as potential confounders in the review-to-star process. To "adjust" for their influence, we augment the cross-entropy

loss with a custom loss that penalizes model's sensitivity to polarizing tokens. We demonstrate that the adjusted model is statistically more robust and less biased.

2 Experimental Setup

2.1 Dataset

We use the yelp_review_full dataset from Huggingface, which contains $\approx 650,000$ English reviews paired with integer star ratings in $\{1, 2, 3, 4, 5\}$. We adopt the standard split, reserving 10% of the training portion for validation and using the official test split for final evaluation.

For pre-processing, we apply the uncased BERT WordPiece tokenizer (`google/bert_uncased_L-4_H-256_A-4`) with a maximum length of 512 (truncating longer reviews and padding shorter ones).

2.2 Model & Training Dynamics

Our sentiment model, `BertSentiment`, consists of a light BERT backbone (`google/bert_uncased_L-4_H-256_A-4`, i.e., 4 Transformer layers, hidden size 256, 4 attention heads), followed by dropout (rate 0.1), a LayerNorm, and a single linear layer mapping the pooled `[CLS]` embedding to five logits. We initialize the classifier weights with Xavier uniform and biases to zero.

We construct PyTorch `DataLoaders` over the cached tensors with batch size 32, shuffling the 80% training split and holding out 20% for validation each epoch. Training proceeds for up to 10 epochs with early stopping on the validation loss, using AdamW (learning rate 2×10^{-5}, weight decay 0.01) and gradient clipping at norm 1.0. We aim to minimize cross-entropy loss.

2.3 Identifying the Most Sentimental Words

To discover which tokens most strongly signal positive or negative sentiment, we split our training set into two corpora: the positive corpus of all reviews with stars ≥ 4, and the negative corpus of all reviews with stars ≤ 2. After lemmatizing every review with SpaCy, we count the total occurrences $n^+(w)$ and $n^-(w)$ of each lemma w, and let $N^+ = \sum_w n^+(w)$ and $N^- = \sum_w n^-(w)$ be the respective corpus sizes. For example, the review "I loved the food but the service was terrible" is tokenized and lemmatized into

$$[\ \texttt{i},\ \texttt{love},\ \texttt{the},\ \texttt{food},\ \texttt{but},\ \texttt{the},\ \texttt{service},\ \texttt{be},\ \texttt{terrible}\ ],$$

so that "loved" becomes `love` and "was" becomes `be`.

Following Monroe et al., we impose a symmetric Dirichlet prior of mass α to stabilize rare counts, then compute the smoothed log-odds difference

$$\Delta_w = \log \frac{n^+(w) + \alpha}{N^+ + \alpha K - \left(n^+(w) + \alpha\right)} \; - \; \log \frac{n^-(w) + \alpha}{N^- + \alpha K - \left(n^-(w) + \alpha\right)},$$

and standardize by its estimated standard deviation:

$$z_w = \frac{\Delta_w}{[1/(n^+(w) + \alpha) + 1/(n^-(w) + \alpha)]^{1/2}}.$$

Intuitively, z_w measures the difference in usage rates of w between high-star and low-star reviews, normalized by sampling variability, so that a large positive z_w (e.g. "delicious") indicates strong association with positive feedback, whereas a large negative z_w (e.g. "awful") indicates negative sentiment.

Before ranking by z_w, we apply three filters to focus on reliable evaluative language: we require total frequency $n^+(w) + n^-(w) \geqslant 50$, exclude SpaCy's built-in stop-words (e.g. `the` & `and`), and retain only lemmas tagged as adjectives or adverbs. Examples of most positively and negatively sentimental words:

Word	great	best	delicious	friendly	good	bad	rude	worst	horrible	terrible
Score	108	61.6	60.7	60.3	59.4	−71.6	−68.6	−64.5	−63.9	−60.5

3 Experiments, Results, & Analyses

In this Section we present a series of experiments that we have conducted, as well as their results and some relevant discussions.

3.1 Asymmetric Effects of Polarizing Words

To quantify how the mere presence of an extreme token shifts the star rating, we define two binary treatments for each review i: T_i^- the indicator that the review contains at least one strongly negative word, and likewise T_i^+ positive. A single covariate, we let X_i be the total review length. (Note we assume no unobserved confounding beyond X_i, and we later verify the overlap in propensity scores to back up our claim. See §3.3 for a different perspective.) We then fit a logistic regression

$$\hat{e}_i \;=\; \Pr(T_i = 1 \mid X_{\text{base},i}) \;=\; \text{LogisticRegression}\big(\text{StandardScaler}(X_{\text{base}})\big),$$

and compute overlap-weight ATE

$$\hat{\Delta} = \frac{\sum_i w_i T_i y_i}{\sum_i w_i T_i} \;-\; \frac{\sum_i w_i (1 - T_i) y_i}{\sum_i w_i (1 - T_i)}, \quad w_i = \begin{cases} 1 - \hat{e}_i, & T_i = 1 \\ \hat{e}_i, & T_i = 0. \end{cases}$$

Applied to `has_neg` and `has_pos`, this yields

$$\Delta^- = -1.327, \qquad \Delta^+ = +1.392, \qquad \Gamma = \Delta^- - \Delta^+ = -2.719.$$

We then bootstrap (500 resamples) to obtain a 95% confidence interval $[-1.352, -1.303]$ for Δ^- and a 95%-CI $[1.365, 1.419]$ for Δ^+, confirming both effects are highly significant and that positive tokens exert a slightly stronger lift than the drag of negative tokens, but the difference is almost minimal.

What follow are a few sanity checks. First, to verify covariate balance, we examine the standardized mean difference

$$\text{SMD} = \frac{\overline{X}_{T=1} - \overline{X}_{T=0}}{\sqrt{\frac{1}{2}\big(\text{Var}(X_{T=1}) + \text{Var}(X_{T=0})\big)}},$$

and its weighted version by w_i, on review length. For negative-word treatment the raw SMD is $+0.762$ and the weighted SMD reduces to $+0.001$; for positive words the raw SMD is $+1.438$ and the weighted SMD reduces to $+0.001$.

Further, over 90% of our fitted propensity scores $e_i = \Pr(T_i = 1 \mid X_i)$ fall under $[0.1, 0.9]$. These diagnostics confirm that overlap holds and our overlap-weighted ATE estimates are reliable.

Finally, a placebo test using 100 near-zero-z words (e.g. “upfront,” “stairs,” “regional,” etc.) yields an ATE of -0.038 stars, effectively indistinguishable from zero. Taken together, these results establish that highly polarized adjectives and adverbs causally shift human star ratings significantly and in roughly equal magnitude either direction, with confounder adjustment and no spurious effects taken care of.

3.2 Nonlinear and Saturation Effects of Polarizing Words

To investigate whether repeating extreme adjectives leads to linearly increasing effects or instead exhibits saturation (or diminishing returns), we treat the count of polar words as a continuous “dosage.” Concretely, for each review i we let

$$T_i^- = \text{number of negative polar lemmas}, \quad T_i^+ = \text{number of positive polar lemmas},$$

and use review length as a baseline covariate X_i. We first fit a Poisson generalized propensity score $\hat{\lambda}_i = \mathbb{E}[T_i \mid X_i]$ via a pipeline of standard scaling plus Poisson regression. The generalized propensity score for the observed dosage T_i is then

$$R_i = \Pr(T_i \mid X_i) = \mathrm{PoisPMF}\big(T_i; \hat{\lambda}_i\big).$$

To flexibly model the outcome surface $Y_i \mid T_i, R_i, X_i$, we regress the star rating on the features $(T_i,\ T_i^2,\ R_i,\ T_i R_i,\ X_i)$ using ordinary least squares. Finally, for each integer dosage $t = 0, 1, \ldots, 5$ we compute

$$\hat{m}(t) = \frac{1}{N} \sum_{i=1}^{N} \hat{Y}_i\big(t,\ R_i(t),\ X_i\big), \quad R_i(t) = \mathrm{PoisPMF}\big(t; \hat{\lambda}_i\big),$$

to trace out the dose–response curve.

The estimated curves display clear saturation and diminishing-return patterns. For negative polar words,

$$\hat{m}^-(0) \approx 3.39, \quad \hat{m}^-(1) \approx 2.34, \quad \hat{m}^-(2) \approx 1.61, \quad \hat{m}^-(3) \approx 1.08, \quad \hat{m}^-(4) \approx 0.60, \quad \hat{m}^-(5) \approx 0.14,$$

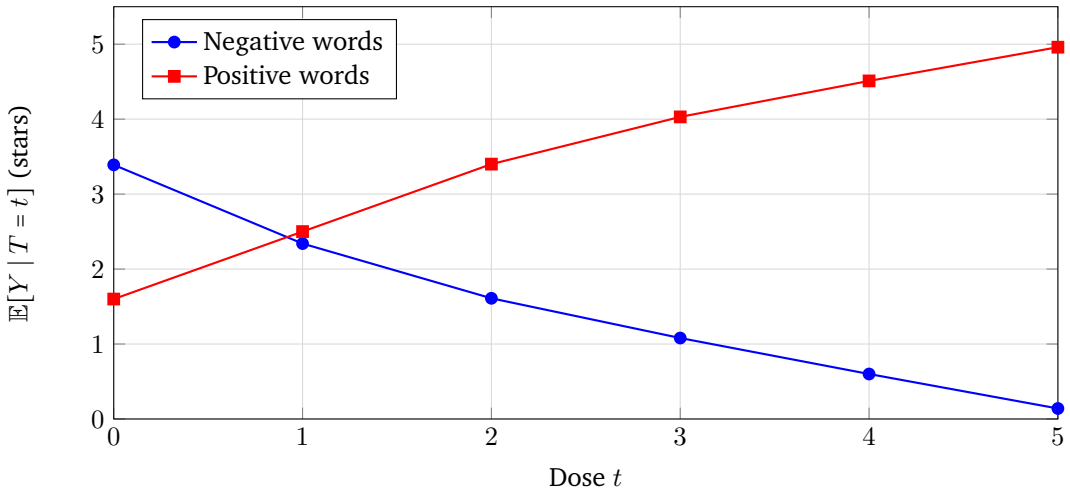
indicating that the first negative descriptor inflicts the largest drop in rating, with each additional descriptor doing progressively less harm. Conversely, for positive words,

$$\hat{m}^+(0) \approx 1.60, \quad \hat{m}^+(1) \approx 2.50, \quad \hat{m}^+(2) \approx 3.40, \quad \hat{m}^+(3) \approx 4.03, \quad \hat{m}^+(4) \approx 4.51, \quad \hat{m}^+(5) \approx 4.96,$$

showing strong initial gains that plateau as the number of positive descriptors increases. These nonlinear patterns confirm that while a few highly subjective words can substantially sway star ratings, piling on more extreme adjectives yields limited additional effect.

3.3 Causal Impact of Subjective Language and Model Debiasing

To quantify how the presence of strongly subjective tokens affects human ratings, we first estimate the average treatment effect using Double Machine Learning (DML) with rich text-based confounders. For each review i, we define the binary treatment $T_i = 1$ if the text contains at least one lexicon adjective or adverb, and $T_i = 0$ otherwise, and let $Y_i \in \{1, \ldots, 5\}$ be the star rating. As covariates we include the raw review length (number of polar tokens) and a low-dimensional embedding derived from Sentence-BERT. Concretely, we encode each review—after masking



Dose–response curves for negative (blue) and positive (red) polar words.

out subjective tokens—using the `all-MiniLM-L6-v2` model (384 output dimensions), then apply PCA with 16 components to obtain $\mathrm{SBERT}_i \in \mathbb{R}^{16}$. The confounder vector is $X_i = (\mathrm{length}_i, \mathrm{SBERT}_i)$.

Following Chernozhukov et al., we fit two nuisance functions on 5-fold stratified splits: a gradient-boosted regressor for $\hat{g}(X) = \mathbb{E}[Y \mid X]$ and a classifier for $\hat{m}(X) = \Pr(T = 1 \mid X)$. We then compute residuals $\tilde{Y}_i = Y_i - \hat{g}(X_i)$ and $\tilde{T}_i = T_i - \hat{m}(X_i)$, and estimate the ATE by ordinary least squares:

$$\hat{\Delta}_{\mathrm{subj}} = \frac{\sum_i \tilde{Y}_i \tilde{T}_i}{\sum_i \tilde{T}_i^2} \approx 0.43.$$

This indicates that, all else equal, inserting one subjective descriptor causally raises the human star rating by roughly 0.43 points.

Having identified the causal effects of subjective languages on human ratings, next, we ask whether our vanilla BERT predictor repeats this bias. For each review, we construct a counterfactual text $\tilde{x}_i$ by masking all lexicon words; see below for an example (assuming "cold" and "raw" are strongly subjective — the example below is purely illustrative).

"The pasta was incredibly raw." $\longrightarrow$ "The pasta was `<mask>` raw."

Note we also assume no-spillover—that deleting subjective tokens does not change the true underlying sentiment—so that the masked text remains a valid counterfactual. We find that

$$\Delta_{\mathrm{BERT}} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i^{\mathrm{orig}} - \hat{y}_i^{\mathrm{masked}} \right) \approx 0.30,$$

which shows removing a handful of subjective tokens is enough to significantly alter the model's outcome: the predictor is quite sensitive to those tokens under this controlled deletion.

This is bad! It implies that the model is also succumbing to the bias induced by these stylistic choices. To mitigate this, we introduce a counterfactual invariance regularizer alongside the usual cross-entropy loss. During training, for each (x_i, y_i) we also feed the masked text $\tilde{x}_i$ through the same model, and we penalize large differences between the original and the counterfactual input. Formally we optimize

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \Big[\ell\big(f(x_i), y_i\big) + \ell\big(f(\tilde{x}_i), y_i\big) \Big] + \lambda \frac{1}{N} \sum_{i=1}^{N} \big(f(x_i) - f(\tilde{x}_i) \big)^2,$$

where ℓ is cross-entropy and λ weights the invariance term. After training with $\lambda = 1$, the model's sensitivity Δ_{BERT} drops to nearo zero (0.04) while other performance metrics (accuracy, MAE) on the held-out test set remain unchanged. This demonstrates that simple counterfactual augmentation can eliminate the predictor's undue dependence on superficial subjective wording, yielding a more robust and fair sentiment model.