Sentiment Analysis of Customer Support Conversations Using nanoGPT and GPT-2

Esra Şekerci
Department of Information Systems
Middle East Technical University
Ankara, Turkey
esra.sekerci@metu.edu.tr

***Abstract*— This study conducts a comparative evaluation of transformer-based architectures for sentiment classification in customer-agent dialogues. A structured preprocessing pipeline is applied to preserve conversational structure and eliminate linguistic noise. Two modeling strategies are investigated: (i) a compact transformer architecture inspired by NanoGPT, trained from scratch, and (ii) a fine-tuned GPT-2 model leveraging pretrained weights. Both models are evaluated on a stratified and balanced dataset, with performance assessed on a held-out test set. Results underscore the advantage of transfer learning, with GPT-2 demonstrating superior generalization in dialogue-based sentiment prediction.**

Keywords—Sentiment classification, transformer models, GPT-2, NanoGPT, customer-agent dialogue, transfer learning.

I. Introduction

Sentiment analysis, one of the fundamental tasks in Natural Language Processing (NLP), aims to classify the emotional tone of textual data. While traditional approaches focus on short texts, applications in customer service demand models that can interpret multi-turn dialogues, where sentiment is dispersed across conversational turns and shaped by speaker dynamics.

With the emergence of transformer-based language models the paradigm has shifted toward end-to-end architectures capable of learning semantic and contextual representations from data. Models like GPT-2, originally designed for text generation, have shown strong transfer learning capabilities especially when fine-tuned for specified tasks. Alternatively, smaller models like NanoGPT offer a lightweight solution for training from scratch, especially in resource-constrained settings.

This study explores the comparative evaluation of a fine-tuned GPT-2 model and a custom-trained NanoGPT-based transformer for sentiment classification on customer service dialogues. The objective is to examine the effectiveness of each approach under a consistent preprocessing, training, and evaluation framework.

II. Dataset

A. Dataset Overview

The dataset comprises annotated transcripts of customer support interactions between clients and service agents. Each instance includes a multi-turn conversation and an associated sentiment label (positive, neutral, or negative), along with metadata such as issue type, agent experience level, and product category. The initial dataset contains 3,074 samples, which form the basis for training, validation, and evaluation.

B. Data Cleaning and Label Encoding

To enhance data quality, ensure consistency and remove noise we applied a structured preprocessing pipeline. Text normalization involved lowercasing and whitespace trimming, while non-linguistic noise (e.g., punctuation, emails, URLs, and phone numbers) was removed to improve tokenization. Speaker tags (customer: and agent:) were preserved to maintain dialogue structure. Generic agent phrases were filtered to reduce sentiment bias, and frequent misspellings were corrected using a curated dictionary.

Sentiment labels were encoded as ordinal integers (negative = 0, neutral = 1, positive = 2). To address class imbalance, the underrepresented positive class was oversampled prior to the stratified train-validation split.

C. Exploratory Data Analysis

A series of exploratory analyses were performed to assess structural and statistical properties of the dataset prior to model development.

a) Sentiment Distribution: Initial inspection revealed substantial class imbalance: neutral instances comprised 56.3% of the data, negative 41.7%, and positive only 2.0% (Figure 1). This imbalance necessitated targeted oversampling of the positive class during preprocessing to support robust learning.

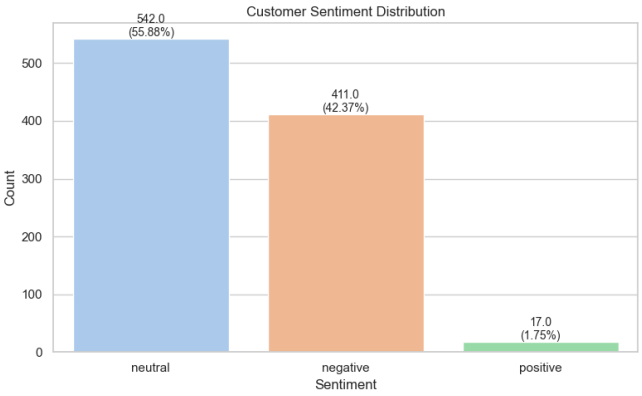

Figure 1 - Sentiment distribution before preprocessing

b) Conversation Length: Dialogue length was analyzed in both character and token counts to assess verbosity across sentiment classes. The data followed a long-tailed distribution, with most conversations being short but some extending significantly (Figure 2). Stratified statistics (Table I) show that negative conversations are consistently longer than their neutral and positive counterparts, suggesting an association between dissatisfaction and verbosity. ANOVA testing confirmed a significant difference across sentiment groups ($F = 99.28$, $p < 0.0001$).

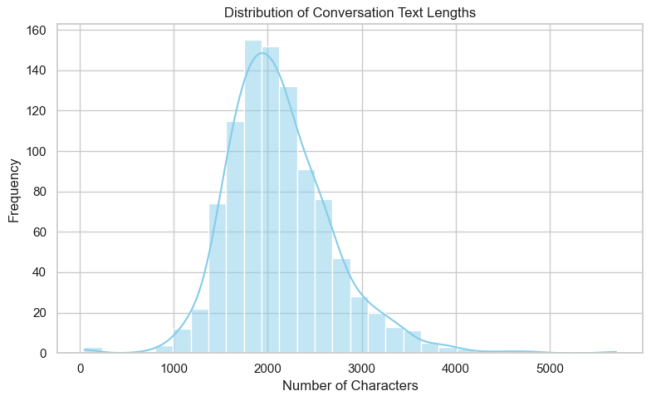

Figure 2 - Distribution of conversation lengths in characters

	count	mean	std	min	25%	50%	75%	max
negative	411.0	417.67	101.35	8.0	351.5	406.0	471.5	992.0
neutral	542.0	337.78	80.60	129.0	285.0	329.0	373.0	802.0
positive	17.0	294.00	48.18	213.0	253.0	286.0	329.0	378.0

Table 1 - Summary Statistics for Character Count by Sentiment Label

c) Lexical Patterns: A word cloud constructed from the cleaned corpus (Figure 3) illustrates common lexical items, with frequent terms like "order," "issue," and "help" reflecting the transactional nature of customer-agent interactions.

Figure 3 - Word cloud from preprocessed conversations.

In summary, EDA revealed three main insights: (i) significant class imbalance, addressed via oversampling; (ii) statistically relevant length-sentiment correlation; and (iii) moderate lexical association with product categories. Combined with systematic preprocessing, these findings informed subsequent modeling choices and contributed to improved data quality and task-specific representation learning.

III. Modeling

The first transformer-based architectures is a custom implementation based on NanoGPT, restructured from an autoregressive generator to a classification model. The original causal language modeling head was replaced with a fully connected output layer, mapping pooled token embeddings to sentiment classes. To facilitate bidirectional context aggregation, decoder-specific constraints were eliminated. The model comprises four transformer layers with 256-dimensional embeddings and sinusoidal positional encodings, offering a compact and interpretable configuration suited for training from scratch.

The second model employs GPT-2, a pretrained decoder-only transformer with rich linguistic priors. Using the GPT2ForSequenceClassification wrapper, we appended a linear classification head to the final transformer block. All transformer layers were unfrozen to support full backpropagation and adaptation to the downstream task. Tokenization relied on GPT-2's subword vocabulary, with EOS-based padding for classification alignment. This architecture leverages pretraining knowledge to improve generalization in low-resource settings.

IV. Evaluation

Models were evaluated using accuracy, macro-averaged precision, recall, and F1-score to ensure fair comparison across imbalanced sentiment classes. Macro F1 was prioritized as it balances sensitivity and precision independently of label distribution.

V. Results

The table below reports test set performance for both models on 30 balanced instances (10 per class). All evaluations were conducted using the models' best checkpoints as determined by validation macro-F1.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
NanoGPT	53.33%	74.75%	53.33%	50.69%
GPT-2	60.00%	81.82%	60.00%	54.34%

Table 2 - Test Set Performance Comparison

VI. Discussion

Fine-tuning a pre-trained model leverages prior linguistic knowledge embedded during large-scale pretraining, enabling faster convergence and improved generalization. GPT-2, initialized with such priors, consistently outperformed the randomly initialized NanoGPT variant across all evaluation metrics. While the architecture of NanoGPT is structurally sufficient for sequence modeling, the absence of pretraining limits its representational capacity and increases its susceptibility to variance and overfitting. This disparity highlights the importance of model initialization, particularly in low-resource settings, and aligns with findings in the transfer learning literature that emphasize pretraining as a regularization mechanism.

Nevertheless, several dimensions of the pipeline remain open to optimization. Improvements in preprocessing—such as incorporating syntactic parsing, discourse-level cues, or speaker turn segmentation—could yield more informative inputs. Additionally, leveraging domain-adaptive pretraining or contrastive learning strategies may benefit both pretrained and scratch-based architectures by enhancing task-specific discrimination.

VII. Conclusion

This analysis highlights the effectiveness of pretrained transformer models for sentiment classification in customer-agent dialogues. The fine-tuned GPT-2 outperformed NanoGPT trained from scratch, confirming the benefits of transfer learning in low-resource settings. Nonetheless, further improvements remain feasible.

Future work may enhance preprocessing strategies, incorporate structural features from dialogues, and integrate auxiliary metadata. Exploring compact pretrained variants and broader evaluation benchmarks could also support more efficient and generalizable models.

REFERENCES

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3

[2] V. B. Parthasarathy, A. Zafar, A. Khan, and A. Shahid, "The ultimate guide to fine-tuning LLMs from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities," *CeADAR Connect Group, University College Dublin*, Version 1.0, 2024. Available: https://doi.org/10.48550/arXiv.2408.13296

[3] T. Xiao and J. Zhu, "Foundations of large language models," *arXiv preprint arXiv:2501.09223*, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.09223