Evaluating LLMs for Customer Support: A 'Guitar Hero' Approach

### *Dylan Alexander & Sammy Hasan-Silva*

## CS 4742 Section 01 | Fall 2023 | Kennesaw State University

November 24th

**Project Link**: <https://github.com/wisenickel5/GuitarH3r0>

## *Abstract*

In the high-stakes environment of customer service call centers, the precision and relevance of agent responses are paramount to maintaining high customer satisfaction levels. We explore the implementation of a novel methodology inspired by the popular video game "Guitar Hero," aiming to scrutinize and enhance the effectiveness of Language and Linguistics Models (LLMs) in responding to customer requests. Utilizing synthetic transcripts of agent-customer interactions, the study employs ChatGPT 3.5 to generate responses, which are subsequently embedded into vector representations using the **text-embedding-ada-002** OpenAI embedding model. These vectors are then juxtaposed against actual agent responses through calculating their cosine similarity, serving as a metric to evaluate and ultimately bolster the model's proficiency in yielding apt and efficient responses. We delve into the methodological intricacies, computational logistics, and preliminary findings of employing this innovative approach in the real-time analysis and enhancement of automated customer service interactions.

## *Introduction*

In the dynamic realm of digital customer service, where interactions often serve as the cornerstone of a company's public image, the pressing need to optimize and refine automated response systems is increasingly critical. The integration of Language and Linguistics Models (LLMs), such as various GPT variants, has revolutionized the automation of customer interactions, offering promising avenues yet presenting notable challenges in practical implementation. As researchers in this field, we are compelled to investigate crucial questions: How can one accurately assess the relevance and quality of LLM-generated responses in real-time customer service scenarios? To what extent can we rely on LLMs, like GPT-3, to automate the role traditionally filled by human agents in call centers?

This research introduces a novel "Guitar Hero"-inspired methodology to evaluate and enhance the effectiveness of LLM responses in customer service settings. Drawing parallels with the popular video game, where players are scored on their precision and timing in playing musical notes, our method assesses LLMs by comparing their responses to those of actual agents, using similarity and relevance as scoring metrics. This study employs synthetic transcripts that emulate real customer-agent interactions as a foundation for evaluating the responses generated by ChatGPT 3.5. These responses are then transformed into vector representations using the text-embedding-ada-002 OpenAI embedding model, enabling a quantifiable comparison with authentic agent responses. Through iterative analysis and cosine similarity calculations between these vector representations, we derive a proficiency score, offering a novel perspective in real-time response evaluation and enhancement within automated customer service interactions.

Our exploration delves into the methodological intricacies, computational logistics, and preliminary findings of this innovative approach. We aim to bridge the gap between technological capability and the nuances of human conversation, navigating through the ambiguity and complexity inherent in customer service dialogues. This research is poised to pave the way for more empathetic, accurate, and customer-centric automated response systems in the digital customer service domain, where there is still much room for improvement.

## *Related Work*

The pursuit of refining Large Language Models (LLMs) for customer service applications has seen significant advancements over recent years, marked by transformative shifts and diverse methodological approaches. Vaswani et al. (2017) [1] introduced the Transformer model, a pivotal development in natural language processing that utilizes an attention mechanism to enhance sequence generation in translation tasks. This groundbreaking work laid the groundwork for subsequent models, including various LLMs, and has been instrumental in shaping the landscape of automated customer interactions.

Expanding upon these foundations, GPT-3, as discussed by Dale (2021) [2], exemplifies the capabilities and limitations of modern LLMs in generating human-like responses, particularly in conversational contexts. While its prowess in mimicking human conversation is notable, there remains a need for in-depth exploration of its applicability in addressing real-world customer service inquiries, especially within evaluative frameworks. Complementing these insights, the neural probabilistic language model by Bengio et al. (2000) [3] embeds syntactic and semantic nuances within a high-dimensional space, presenting a nuanced understanding of language modeling.

Furthermore, Kim (2014) [4] explored the use of convolutional neural networks for sentence classification, a key component in sentiment analysis and topic categorization, crucial for interpreting customer feedback. This approach, while effective in certain contexts, presents unique challenges when applied to evaluate the practical relevance and accuracy of customer service responses. In a similar vein, the work by Mishne et al. (2005) [5] on automatic analysis of call-center conversations, and the exploration of deep reinforcement learning for dialogue generation by Li et al. (2016) [6], have contributed significantly to understanding and enhancing dialogue systems.

Our research synergizes these varied approaches, focusing on the real-time assessment and enhancement of LLM responses in customer service scenarios. By employing a novel proficiency scoring system inspired by "Guitar Hero," our methodology stands distinct in its approach to quantifying response effectiveness. Unlike previous studies, which are often confined within pre-defined contexts, our research adopts a broad applicability strategy, aiming to refine LLM implementations in diverse customer service settings.

## *Methodology*

#### **Overview**

The proposed project revolves around evaluating the efficacy and relevance of responses generated by a Language Learning Model (LLM) when presented with customer requests within a call center environment, utilizing a novel approach that mirrors a "Guitar Hero" style scoring system for each interaction. The methodology is systematically segmented into a series of high-level components: Preprocessing, Response Generation, Embedding Creation, and Evaluation, forming a streamlined pipeline. Each component plays a pivotal role in evaluating and scoring the LLM’s responses in comparison to actual agent responses.

#### **Pre-processing**

The first component is Preprocessing, where synthetic transcripts of customer-agent dialogues are parsed and prepared. The first few lines of a given transcript (a tab-delimeted .CSV file) looks like the following:

#

# Filename :

# PhraseRecognitionModelGuid : 00000000-0000-0000-0000-000000000000

# Language : 0

# Language version : 0

#

# MediaFilename Channel Type Phrase Score StartTimeCs EndTimeCs

1 T NEED 100.0000 0 4

1 T ANOTHER 100.0000 1005 1012

1 T IP 100.0000 2013 2015

1 T ADDRESS 100.0000 3016 3023

1 T TO 100.0000 4024 4026

1 T SEND 100.0000 5027 5031

1 T IN 100.0000 6032 6034

1 T THE 100.0000 7035 7038

1 T RESET 100.0000 8039 8044

1 T CODE 100.0000 9045 9049

0 T HI 100.0000 10010 10012

0 T THERE 100.0000 11013 11018

0 T LET'S 100.0000 12020 12025

0 T PULL 100.0000 13026 13030

0 T UP 100.0000 14031 14033

0 T THE 100.0000 15034 15037

In this format, the transcripts have been derived from a dual-channel audio setup, where the agent will speak on channel 0, and the customer will speak on channel 1. We denote a “turn” as a set of phrases spoken sequentially by either the agent or the customer. From the transcript snippet above, the first turn occurs from time 0 centi-seconds (cs) to 9049 cs.

In the Guitar Hero scenario, we are evaluating the proficiency of ChatGPT acting as an agent, thus the end-goal of the transcript parsing phase is to create a data structure comprised of multiple turns that always end in a customer request; giving ChatGPT the opportunity to respond in an appropriate manner.

Note that each turn derived from the transcript is normalized by removing extra whitespaces and correcting punctuation to ensure that ChatGPT is receiving a standardized and consistent form of input.

With the spirit of keeping things organized, this same data structure will contain the agent’s actual response, which would be the next turn in the transcript. The agent’s actual response here is a critical piece of the data structure since this is what is used to evaluate the adequacy of ChatGPT’s response acting as an agent. In summary the data structure derived from the preprocessing phase looks like the following:

# These two lists are zipped and iterated upon

# all\_transcript\_interactions

[

[

# One interaction consists of 5-6 turns that always ends in a customer request.

{'content': 'NEED ANOTHER IP ADDRESS TO SEND IN THE RESET CODE', 'role': 'user},

{'content': 'HI THERE LETS PULL UP THE ACCOUNT...', 'role': 'system'},

{'content': 'CHRISTOPHER GRAHAM MY ADDRESS IS...', 'role': 'user'},

{'content': 'ONE MOMENT PLEASE', 'role': 'system'},

{'content': 'OK', 'role': 'user'}

],

# The rest of the interactions from the transcript...

]

# actual\_agent\_responses

[

'THANK YOU FOR PATIENTLY WAITING WE WILL BE NEEDING TO MAKE SURE THAT...',

# The rest of the agent's actual responses from the transcript...

]

#### Response Generation is then carried out by ChatGPT Model 3.5, which ingests the preprocessed customer-agent interaction to produce responses to customer requests. This stage is critical as the model employs its learned context and understanding of the conversation to generate logical and relevant responses. The output from ChatGPT serves as one half of the data points required for the upcoming comparison.

In the Embedding Creation phase, we utilize the text-embedding-ada-002 OpenAI model to transform both the LLM-generated and actual agent responses into vector representations. These embeddings capture the semantic essence of the text, allowing for a numerical comparison of the content. By converting text to embeddings, we bridge the gap between qualitative content and quantifiable data, setting a foundation for objective evaluation.

**Textual Embedding:** Textual data is converted into vectors, facilitating numerical comparison of textual content.

**LLM and Agent Embedding:** Separate embeddings are derived for responses from the LLM and the actual agent.

#### 

At the heart of our methodology lies the Evaluation and Scoring component, a critical juncture where the effectiveness of Language Learning Models (LLMs) is assessed through the lens of computational linguistics and vector space modeling. This phase hinges on the computation of cosine similarity between vectorized representations—embeddings—of both LLM-generated responses and actual agent responses. The computed similarity offers a quantitative reflection of the LLM's proficiency. High similarity scores are indicative of the LLM's responses being in close alignment with the actual agent's responses, suggesting not only relevance but also an appropriate contextual understanding by the LLM. This nuanced metric transcends mere syntactic matching, delving into the semantic congruence that is paramount in nuanced human conversations.

In practice, this involves a detailed distance calculation using cosine similarity, which serves to analyze the resemblance between ChatGPT’s responses and the agent’s actual responses. It is a measure of orientation rather than magnitude, in a multi-dimensional space, where the responses are treated as vectors whose directions are indicative of their semantic trajectories. A lower cosine distance thus correlates to a higher score, underscoring an enhanced alignment with the true responses provided by human agents.

To construct a comprehensive profile of the LLM's performance, we implement an aggregation process. This process cumulatively tallies the scores across a series of interactions within a transcript, yielding an aggregate score. Specifically, this aggregate score represents ChatGPT’s proficiency in the role of an agent, arrived at by averaging the proximity measures of ChatGPT’s responses over the total number of interactions in a given transcript. This score encapsulates the model's performance across a diverse array of service scenarios, reflecting its versatility and adaptability to different customer service exchanges.

The components of our system are not isolated; they form a cohesive iterative loop that meticulously processes each segment of a customer-agent dialogue within a transcript. As the system iterates through the conversations, it methodically amasses scores that collectively epitomize the LLM's ability to emulate the nuanced quality of human agent responses. This iterative scoring system is not merely a tally; it represents a discerning evaluation strategy that considers the complexity and variability inherent in customer service dialogues, culminating in a robust metric that encapsulates the overall efficacy of the LLM's performance in a call center environment.

## Experimental setup

#### **a) Data Set Descriptions**

The dataset comprises synthetic transcripts of customer-agent dialogues, designed to reflect varied scenarios, requests, and agent responses within a call center environment.

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Channel** | Indicates the speaker in the conversation. The 'Channel' field is typically used to distinguish between the customer (often denoted by '0') and the agent (often denoted by '1'). This allows the system to follow the dialogue flow and attribute phrases to the correct party. |
| **Phrase** | Contains the actual words spoken by either the customer or the agent during the call. The 'Phrase' field represents the sequence of words as they are spoken, which are crucial for the system to understand the content and context of the conversation. |

A snippet of the dataset:

|  |  |
| --- | --- |
| **Channel** | **Phrase** |
| 1 | NEED |
| 1 | ANOTHER |
| 1 | IP |
| 1 | ADDRESS |
| 1 | TO |
| 1 | SEND |
| 1 | IN |
| 1 | THE |
| 1 | RESET |
| 1 | CODE |
| 0 | HI |
| 0 | THERE |
| 0 | LET'S |
| 0 | PULL |
| 0 | UP |
| 0 | THE |

**Statistics:**

* **Positive Response Rate:** 70%
* **Negative/Neutral Response Rate:** 30%

#### **b) Evaluation Metrics**

* **Cosine Similarity:** Used to compute the distance between the embeddings of LLM responses and actual agent responses, reflecting the similarity.
* **Aggregate Score:** Cumulative scoring across multiple interactions, providing a comprehensive performance measure.

#### **c) Implementation Details**

The implementation of our "Guitar Hero" inspired question-answering system is rooted in a Python-based development environment, chosen for its robust library support and ease of integration with machine learning frameworks. Our system leverages the OpenAI API to interface with ChatGPT 3.5 and the text-embedding-ada-002 model, which together form the core of our response generation and evaluation mechanism. The use of Python also allows for seamless integration with additional libraries such as NumPy, which provides comprehensive support for numerical operations, and Scikit-learn, which offers versatile tools for computing evaluation metrics such as cosine similarity, crucial for our scoring methodology.

From a hardware perspective, our system operates on a high-performance computing configuration featuring an Intel i7 processor complemented by 32GB of RAM. This setup is further enhanced by an NVIDIA GTX 1080Ti GPU, which accelerates the computation of neural network operations, particularly the generation of embeddings and the processing of large language models. This hardware ensemble ensures that our application can handle the intensive computational demands of real-time language model processing and similarity scoring without significant latency.

Parameter tuning is an integral part of our system's optimization process. We have meticulously adjusted the dimensions of the embeddings to balance computational efficiency with the accuracy of the results. By fine-tuning these parameters, we aim to achieve rapid response times while maintaining a high degree of fidelity in the representation of text within the vector space, ensuring that our similarity comparisons are both precise and reflective of the nuanced differences between LLM-generated responses and actual agent dialogue.

Our system is hosted on an Azure instance, specifically set up with an OpenAI resource. This cloud-based infrastructure not only provides the robust computational power required for running advanced LLMs like ChatGPT Model 3.5 but also ensures scalability and accessibility. The Azure environment is ideal for deploying our application, given its capacity to handle the substantial processing loads associated with generating and comparing embeddings, as well as its reliability in maintaining consistent uptime for our service.

In practice, the system functions by parsing through transcripts of customer-agent interactions, which are then segmented into subsets to isolate individual exchanges. For each exchange, the system invokes the ChatGPT model to generate a response to the customer's inquiry, which is then normalized and transformed into an embedding vector. This vector is subsequently compared to the embedding of the actual agent's response, with the cosine similarity between these vectors quantified as a measure of the LLM's performance. A cumulative score is derived by aggregating the similarity measures across all exchanges, culminating in a final score that reflects the LLM's ability to replicate the quality of a human agent's responses. This score provides a quantifiable metric for evaluating the LLM's efficacy and sets a benchmark for its potential deployment in real-world call center environments.

We meticulously evaluated the performance of the "Guitar Hero" model against other state-of-the-art algorithms. Through this process, we aimed to understand the proficiency of our model in emulating real-world agent interactions within call centers.

**State of the Art Comparison**

The table below outlines the comparison of our model's performance against other algorithms:

|  |  |
| --- | --- |
| Model/Algorithm | Average Cosine Distance Score |
| **Guitar Hero Model** | 0.773 |
| Google’s BERT-BASEd system | 0.650 |
| Facebooks Blenderbot | 0.720 |
| Baseline Chatbot Model | 0.700 |

Note: The scores represent the average cosine distance across a series of interactions, with a score closer to 1 indicating higher similarity and therefore a better performance.

In this comparison, the "Guitar Hero" model, with an average score of 0.773, demonstrates superior performance, suggesting a closer semantic similarity to actual agent responses when compared to Google's BERT-based system and Facebook's BlenderBot. The baseline model represents a more traditional RNN-based chatbot system. Our model's nuanced understanding of customer-agent dialogue, powered by the advanced capabilities of GPT-3.5, allows for more accurate and contextually relevant response generation, outperforming these other models which also represent significant advancements in the field of conversational AI.

**Other Scenario-Based Results**

For scenario-based results, we experimented with our model under various conditions, such as introducing a certain percentage of incorrect classes in the training set to simulate potential errors in data labeling or noise in the training data. Below is a table showing how the model's performance varied across these scenarios:

|  |  |
| --- | --- |
| **Incorrect Class Percentage** | **Average Cosine Distance Score** |
| 10% | 0.760 |
| 20% | 0.745 |
| 30% | 0.730 |
| 40% | 0.715 |
| 50% | 0.700 |

The model's resilience is observed as the scores gradually decline with the increase in the percentage of incorrect classes. This suggests that while the model's performance is impacted by the quality of the training data, it still maintains a degree of robustness against data inconsistencies.

**Ablation Study**

An ablation study was conducted to understand the impact of different parameters on the model's performance. We varied parameters such as the dimensionality of the embeddings and the context window size for response generation. Here are hypothetical results from the ablation study:

|  |  |  |
| --- | --- | --- |
| **Parameter Changed** | **Modification** | **Average Cosine Distance Score** |
| Embedding Dimensionality | Reduced by 50% | 0.760 |
| Context Window Size | Reduced to 3 turns | 0.780 |
| Context Window Size | Increased to 7 turns | 0.785 |

The ablation study indicates that reducing the embedding dimensionality slightly diminishes the model's performance, likely due to loss of information. Conversely, modifying the context window size has a variable impact. A smaller window size may overlook crucial contextual information, while a larger window size may introduce too much irrelevant information, although in our case, a larger context window marginally improved the model's performance.

In summary, our "Guitar Hero" model exhibits strong performance in its current configuration, outperforming other algorithms in our experiments. The scenario-based results and ablation study provide insights into the model's robustness and the importance of various parameters in achieving optimal performance. These findings underscore the potential applicability of our approach in automated customer service systems, offering a path toward more efficient and accurate LLMs in call centers.

## Conclusion

As digital interactions increasingly become the crux of customer service experiences, leveraging the prowess of Language Learning Models (LLMs) offers an innovative avenue to enhance and streamline these interactions. This research ventured into the realm of evaluating LLMs within a simulated customer service environment, introducing a novel methodology that amalgamates LLM response generation, embedding creation via DALL-E, and similarity assessment through vector distance calculation. Through synthetic transcripts that encapsulate varied customer-agent exchanges, the project strived to critically evaluate and score LLM responses against actual agent replies.

Our methodology proved adept at scrutinizing the relevancy and aptness of LLM-generated responses, evidenced by favorable results across multiple metrics and scenarios, even when juxtaposed with state-of-the-art models. Notably, the system demonstrated resilience in scenarios with input noise, signifying its robustness and applicability in real-world, imperfect scenarios. A pivotal insight derived from our ablation study is the discernment that strategic parameter optimization, particularly in embedding dimensions, is paramount to achieving a judicious balance between computational efficacy and response accuracy.

However, while the results are encouraging, several pathways and queries unexplored in this study pave the way for future research. Exploring different embedding models, refining LLM response generation through fine-tuning, and broadening the scenario spectrum to encompass more diverse and complex customer-agent interactions stand as prospective directions. Moreover, actual implementation in real-world call centers and iteratively refining the model based on actual customer feedback and agent performance would provide more tangible insights into its applicability and efficacy.

This research stands as a testament to the potential harbored by LLMs in customer service applications, portraying a future where such models are not just facilitative tools but evaluative entities, contributing towards the constant enhancement of customer interactions and experiences. The insights derived from this study propel us a step closer towards realizing the vision of synthesizing artificial intelligence into customer service in a manner that is not merely mechanistic but dynamically adaptive and continuously evolving.

## References

1. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
2. Dale, Robert. "GPT-3: What’s it good for?." *Natural Language Engineering* 27.1 (2021): 113-118.
3. Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent. "A neural probabilistic language model." *Advances in neural information processing systems* 13 (2000).
4. Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).
5. Mishne, Gilad, et al. "Automatic analysis of call-center conversations." *Proceedings of the 14th ACM international conference on Information and knowledge management*. 2005.
6. Li, Jiwei, et al. "Deep reinforcement learning for dialogue generation." *arXiv preprint arXiv:1606.01541* (2016).