Self-Aware Feedback-Based Self-Learning in Large-Scale Conversational AI

Pragaash Ponnusamy* **Clint Solomon Mathialagan***
Gustavo Aguilar **Chengyuan Ma** **Chenlei Guo**
Amazon Alexa
{ponnup,matclint,gustalas,mchengyu,guochenl}@amazon.com

Abstract

Self-learning paradigms in large-scale conversational AI agents tend to leverage user feedback in bridging between what they say and what they mean. However, such learning, particularly in Markov-based query rewriting systems have far from addressed the impact of these models on future training where successive feedback is inevitably contingent on the rewrite itself, especially in a continually updating environment. In this paper, we explore the consequences of this inherent lack of self-awareness towards impairing the model performance, ultimately resulting in both Type I and II errors over time. To that end, we propose augmenting the Markov Graph construction with a superposition-based adjacency matrix. Here, our method leverages an induced stochasticity to reactively learn a locally-adaptive decision boundary based on the performance of the individual rewrites in a bi-variate beta setting. We also surface a data augmentation strategy that leverages template-based generation in abridging complex conversation hierarchies of dialogs so as to simplify the learning process. All in all, we demonstrate that our self-aware model improves the overall PR-AUC by 27.45%, achieves a relative defect reduction of up to 31.22%, and is able to adapt quicker to changes in global preferences across a large number of customers.

1 Introduction

Large-scale conversational AI systems such as Alexa, Google, Siri etc. serve millions of users daily all over the planet, who speak diverse languages and have a myriad of regional preferences. These models need to be constantly updated with new data to adapt to changing customer behavior and trends. Data curation processes that rely solely on human annotations cannot possibly scale to sustain the rapid update pace of these systems. Therefore, quite naturally, these AI agents have increased their reliance on explicit and implicit feedback from customer interactions to automate the learning process while limiting manual annotation efforts selectively only to auditing and quality control purposes.

In such feedback-based self-learning systems where new streams of data are being funneled in to continually update the system, the mere presence of the ML model itself inevitably impacts future training data. This is rather evident with query rewriting models where the reformulated query becomes intertwined with the original utterance to the extent where the successive feedback in the customer-system interaction paths become contingent on the rewrite. Here, we show that as these models continue to be updated without accounting for this unintended interference, they tend to learn false equivalencies between the original requests and rewrites, thereby impeding their own self-learning capabilities.

In this work, we build upon an absorbing Markov Chain model to make the model self-aware i.e. it can distinguish between customer requests and system rewrites, and adapt its decision boundary based on the quality of the rewrites. Note that the system can also be an ensemble of heterogeneous agents proposing different reformulations for the same query. The self-learning Markov model does not require any agent specific information and rather treats them all as a single entity. Thus, this work can be integrated into any conversational AI system to enable self-learning at a system-level without major changes to the rest of the architecture.

2 Related Work

Query rewriting techniques, particularly in the form of suggestive disambiguation have been extensively employed in online search systems (Jansen et al., 2009; Antonellis et al., 2008; He et al., 2016; Riezler and Liu, 2010), so as to increase recall and im-

* Equal contribution

Proceedings of NAACL-HLT 2022: Industry Track Papers, pages 324 - 333

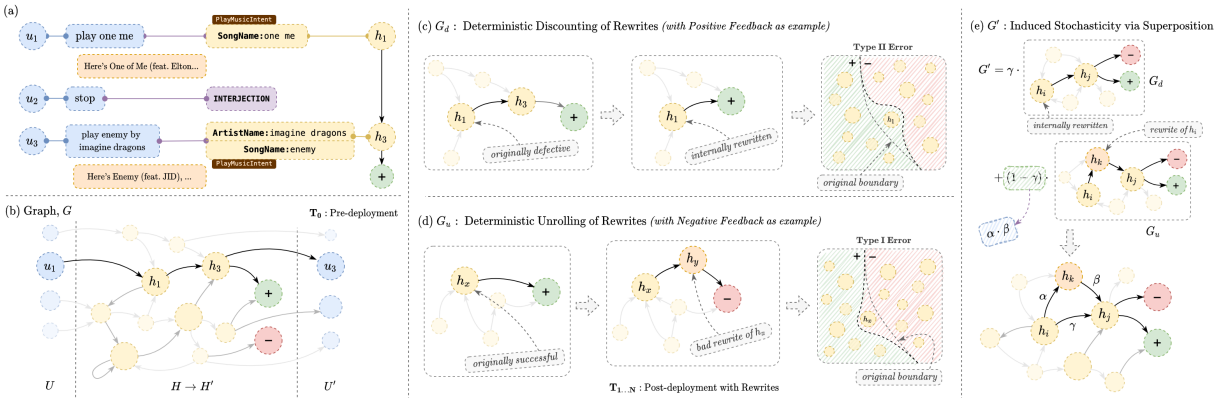

Figure 1: A general walk-through for motivating a meta-state augmented Graph: Beginning with the original construction of chains in **(a)** where utterances, U are projected into the hypothesis space, H before being encoded into the absorbing Markov model in **(b)** showing how a target rewrite in U' is resolved given a source in U. Thereafter, upon deployment, the effect of continuing to model the Graph as before i.e. by discounting the presence of rewrites, G_d in **(c)** and choosing to always unroll the internal rewrites as an externalized state, G_u in **(d)**, both lead to Type II and I errors respectively. Note that the decision boundaries over discrete spaces here are to illustrate the nature of mis-classifications. Naturally, in attempt to balance these two categories of error, a superposition of G_d and G_u is constructed in **(e)** wherein the rewrites act as meta-states that induce stochasticity within the Graph, G'.

prove click-through rates. Naturally, conversational AI systems have also adopted similar techniques to reduce customer defects (Sodhi et al., 2021; Hao et al., 2020; Su et al., 2019; Rastogi et al., 2019; Roshan-Ghias et al., 2020; Yuan et al., 2021; Fan et al., 2021). To the best of our knowledge, none of them address feedback issues that arise from model-in-the-loop environments.

Previous work has analyzed biases and noises in the feedback loop of machine learning models, particularly in recommendation systems (Chaney et al., 2018; Mansoury et al., 2020; Sun et al., 2019; Mehrabi et al., 2021; Lim et al., 2015; Saito et al., 2020). Khritankov (2021); Sculley et al. (2015); Amodei et al. (2016) delve into the effects of unwanted feedback loops that can lead to AI system instability. These works do not consider misplaced attribution of the feedback itself, which is exacerbated in query-rewriting systems.

In Ponnusamy et al. (2020), customer interactions are modeled as an absorbing chain Markov model, and the candidate that is most likely to result in a successful absorbing state is predicted as the rewrite. This work does not address the equivalence conflation problem that occurs over time in such a setup. We update the Markov formulation to enable self-awareness and resolve the ambiguity in feedback attribution.

In Shi et al. (2021), the Markov model is leveraged as a recall layer that produces candidates which are re-ranked by a self-learning neural model that relies on negative user feedback. While there is not much information on the performance of the recall layer, their neural ranking mechanism is richly augmented with common sense and various user preferences. They do not mention any degradation of the Markov model over time but it is possible that the enriched re-ranker could be compensating for this. In contrast, our work solves the issue within the self-learning Markov model itself as opposed to deferring it to a downstream model. This has the added benefit of accelerating the rate of self-learning.

3 Dataset

To extract the chains of successive customer interactions for the eventual Graph, we first pre-process about 90 days of de-identified time-series utterance data from a representative sample of customers worldwide to construct our dataset of sessions, $\mathcal{D}$. Here, conceptually speaking, each such session represents a time-delimited snapshot of a particular customer's conversation history. To illustrate this, consider the session in Figure 1(a) that encapsulates a series of consecutive utterances which follows a customer interjecting with a "*stop*" and following up with a rephrase of their original request to play the song "*Enemy*". Note that in practice, to maximize the consistency of a conversational goal, the time delay between consecutive turns is heuristically bounded.

Now, while the vast majority of interactions are

indeed stateless, there are those which trigger dialogs so as to solicit the user to disambiguate. This inevitably creates conversational hierarchies that span multiple turns. To ground this, consider the dialog in Figure 2(a) where the system is unable to fulfill the initiating request without first clarifying which playlist to add the song to. To address this complexity and improve the overall intelligibility of the corresponding session, such multi-turn dialogs are abridged by connecting the initiating turn with a synthetic one as shown in Figure 2(c). This is accomplished via template-based DAGs (*the construction of which is explored with greater detail in the Appendix Section 8.1*) wherein the resolved entities towards the end of the corresponding dialog are passed through to generate the synthetic utterance e.g. the DAG in Figure 2(b) is fed with "**SongName:**escape", "**ArtistName:**enrique iglesias", and "**PlaylistName:**kacey's" so as to surface the eventual synthesized utterance, "*add escape by enrique iglesias to kacey's playlist*".

4 Self-Aware Markov Model

Much akin to the original formulation of the Markov model by Ponnusamy et al. (2020), which we henceforth regard as our baseline, our dataset of ordered linear sequence of utterances is first projected into the hypothesis space, H e.g. the utterance "*play one me*" is mapped with the aid of the system's NLU component to the hypothesis, "**Music|PlayMusicIntent|SongName:**one me". Thereafter, they are each terminated with an absorbing state. The union of these disjoint chains tantamount to our Markov Graph, $G = (V, E)$ where $V = H \cup S$ represents the set of all transient and absorbing states respectively, while $E = V \times V$, naturally corresponds to the set of edges. In a more canonical form, the Graph can be represented via the transition matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{0} & \mathbf{I_2} \end{bmatrix} \tag{1}$$

where $\mathbf{Q} \in \mathbb{R}^{|H| \times |H|}$ is the sub-matrix of transition probabilities between transient states such that its (i, j)-th element corresponds to the probability of some source transition state, h_i transitioning to some target transition state, h_j in a single step or mathematically speaking, $q_{i,j} = P(h_j|h_i)$. The sub-matrix $\mathbf{S} \in \mathbb{R}^{2 \times |H|}$ refers to the immediate absorption probabilities of the corresponding transient states i.e. $\mathbf{S} = [\mathbf{s}^+, \mathbf{s}^-]$.

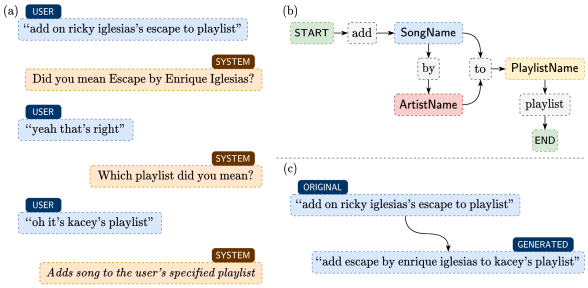

Figure 2: Dialog abridging via template-based DAG with (a) being the original dialog, (b) the extracted template graph, and (c) original with the synthesized utterance.

Now, with $\mathbf{Q}$ being a square matrix[1] whose norm, $\|\mathbf{Q}\| < 1$, the *fundamental matrix* of the Markov model, $\mathbf{N}$ as formulated in Definition 11.3 by Grinstead and Snell (2012) is therefore given by $\mathbf{N} = \sum_{n=0}^{\infty} \mathbf{Q}^n = \left(\mathbf{I}_{|\mathcal{H}|} - \mathbf{Q}\right)^{-1}$ where $\mathbf{Q}^n$ refers to the transition probability sub-matrix $\mathbf{Q}$ after exactly n steps. The *fundamental matrix*, $\mathbf{N}$ is leveraged in resolving the Markov model so as to surface rewrite candidates. Specifically, for a given initial transient state, h_i, a particular target transient state, h_t would be classified as a potential candidate should it be both *reachable* by h_i and conditioned on h_i, it leads to a higher chance of success. Mathematically speaking, this optimization objective can be expressed as $\Phi_\infty(h_t) > \Phi_\infty(h_i)$ where $\Phi_k(h_j)$ refers to the probability of reaching a successful absorbing state, s^+ from h_i via another state h_j that is at most k hops away i.e.:

$$\Phi_k(h_j) = P(s^+|h_j) \cdot \mathbf{N}_{i,j} \tag{2}$$

Here, by identifying the initial transient states that have at least one relatively more successful target transient state and thereby learning a measure of equivalency between states in the hypothesis space, H, the model is effectively able to partition H into those that require reformulation i.e. the defective sub-space, H^- and those that don't i.e. the successful sub-space, H^+. This nature of automatic partitioning leads the model to predict *rewritability* $\hat{y}$ of a given h_i as follows:

$$\hat{y}(h_i) = \mathbb{1}\left\{ \left(\arg\max_{h \in H} \Phi_\infty(h) \right) \neq h_i \right\} \tag{3}$$

[1] As every atomic chain in the Graph is terminated with an absorbing state, these terminal states are guaranteed to always be reachable by any given source transient state, thus ensuring their convergence i.e. $\lim_{n \to \infty} \mathbf{Q}^n = \mathbf{0}$.

4.1 Decision Boundary Degeneracy

Upon deployment however, the very presence of rewrites can significantly destabilize the Graph and impair the integrity of its learned partitioning. To ground this, consider, in the absence of any rewrite, a commonly misrecognized utterance, *"play theme"* (u_1) is followed up with rephrases of *"play team"*, *"play the song team by lorde"*, etc. Now, when the first Markov model $G_d^{(0)}$ is trained initially at T_0 (Figure 1b), it learns to rewrite u_1 to *"play team by lorde"* (r_1). Once deployed, as the Markov model continually learns from customer feedback, u_1 becomes more and more successful than it actually is, since r_1 is not explicitly modeled. Conceptually, this **deterministic discounting** deforms the decision boundary around u_1, resulting in a Type II error (Figure 1c). Such a misclassification will eventually shed the rewrite, forcing the graph to revert to $G_d^{(0)}$. This increases the rephrases to u_1 as previously observed at T_0 and as it gathers sufficient defect statistics, the pattern would repeat, resulting in an unstable oscillatory system that struggles to maintain a consistent decision boundary.

One way of solving the above problem, is to account for rewrites by always including them in the original interaction chain. While this might alleviate the Type II error described above, we show that this limits the system's capability to handle defective rewrites. Imagine a case where a successful utterance, say *"play la da dee"* is followed up by a defective system rewrite *"play lady"* (Figure 1d). This may arise due to a number of reasons such as epistemic or systemic errors, multi-agent interaction, etc. as it is the nature of any statistical model. This process of **deterministic unrolling**, which presumes rewrites to have some degree of latent intent equivalency with the original utterance, would cause the original hypothesis to become more and more defective than it actually is, resulting in a Type I error. To recover the original intent, the customers would need to rephrase following the defective rewrite e.g. *"play la da dee by cody simpson"* or some external guardrail mechanism would need to intervene. Yet again, the Graph will be slow to adapt the decision boundary in response to a Type I error or even worse, may completely fail to recover.

4.2 Meta-State Augmentation

A natural way to balance out these Type I and II errors and thereby maximizing the eventual precision and recall of the rewrites would to be to learn to unroll the rewrite should it improve the customer experience and discount it otherwise. This form of adaptive preservation and suppression of rewrites gives rise to a probabilistic decision making process where the rewrites act as a kind of meta-states that induce stochasticity within the Graph. Conceptually speaking, this is equivalent to both G_d and G_u being in a state of superposition as shown in Figure 1(e) where in the event that a particular transient state, h_i is both rewritten to h_k and followed-up by h_j, a meta-state triplet (MST) is formed. In more robust terms, each of these MSTs within the Graph are comprised of a *viability* edge, (h_i, h_k), a *succeeding* edge, (h_k, h_j), and a *discounting* edge, (h_i, h_j) and are uniquely parameterized by their own set of probabilistic values, namely in this case, $\boldsymbol{\alpha}_{ik}$, $\boldsymbol{\beta}_{kj}$, and $\boldsymbol{\gamma}_{ij}$ respectively so as to allow the Graph to truly be locally adaptive in its learning. To that extent, we first construct a superposition-based transition matrix $\mathbf{A}'$ by updating the probabilities as below:

$$\begin{aligned} \mathbf{A}' &= (\boldsymbol{\lambda} \circ \mathbf{C})^\top \mathbf{D}^{-1} \\ \boldsymbol{\lambda} &= \boldsymbol{\alpha} \circ \mathbf{J}^{(\alpha)} + \boldsymbol{\beta} \circ \mathbf{J}^{(\beta)} + \boldsymbol{\gamma} \circ \mathbf{J}^{(\gamma)} + \mathbf{J}^{(\epsilon)} \end{aligned} \tag{4}$$

where $\mathbf{C} \in \mathbb{Z}_{0+}^{|V|\times|V|}$ such that $\mathbf{C}_{xy}$ refers to the co-occurrence count of the directed edge $e_{xy} = (h_x, h_y)$ in the superposition Graph, G' and $\mathbf{D}$ is the diagonal matrix whose entries are row-wise sum of the matrix $\mathbf{C}$ i.e. $\text{diag}(\mathbf{D}) = (\boldsymbol{\lambda} \circ \mathbf{C}) \cdot \mathbf{1}$. The entries $\mathbf{J}_{xy}^{(\alpha)}, \mathbf{J}_{xy}^{(\beta)}$ and $\mathbf{J}_{xy}^{(\gamma)}$ on the other hand, are the ratios of e_{xy} occurring as either a *viability*, *succeeding* or *discounting* edge respectively. $\mathbf{J}_{xy}^{(\epsilon)}$, however, is the complementary ratio of e_{xy} not being a part of any MST. As a matter of completeness, it's worth noting here that $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{J}^{(\cdot)} \in [0, 1]^{|V|\times|V|}$ such that $\mathbf{J}_{xy}^{(\alpha)} + \mathbf{J}_{xy}^{(\beta)} + \mathbf{J}_{xy}^{(\gamma)} + \mathbf{J}_{xy}^{(\epsilon)} = 1$. Consequently, this modified transition matrix is then used in resolving the Markov Graph as before, to generate rewrite candidates.

4.3 Meta-State Triplet Parameters

In order to adaptively preserve or suppress the rewrites, the weights on the *viability* edges, α should reflect the performance of rewriting. As such, for a given *viability* edge e_{xy} we compare the interaction quality (IQ), as scored by a neural dialog model (Gupta et al., 2021) of the population where h_x was not rewritten, X against that where h_x was rewritten to h_y, $Y|X = W$. Now, suppose that the probability of success in each

of these populations follows Beta distributions i.e. $p_X \sim \text{Beta}(a_x, b_x)$ and $p_W \sim \text{Beta}(a_w, b_w)$. Then, leveraging the beta bi-variate hypothesis testing model as formalized by Miller (2015), the probability that rewriting is comparatively better is given by:

$$P(p_W > p_X) = 1 - \int_0^1 f(p_X, p_W) \cdot d_{p_X}$$

$$f(p_X, p_W) = \frac{p_X^{a_x - 1}(1 - p_X)^{b_x - 1}}{B(a_x, b_x)} \cdot I_{p_X}(a_w, b_w)$$

where, B is the beta function and I, the regularized incomplete beta function. Thereafter, $\boldsymbol{\alpha}_{xy}$ is computed as a variant of $P(p_X > p_Y)$ by leveraging different probability arguments depending on support sufficiency for both p_X and p_Y as detailed in the appendix.

Then, while α reflects the rewrite quality via historical statistics, the weights on the *succeeding* edge, $\beta = \alpha^\rho$ are designed to maintain the semantic connectivity between the rewrite and the succeeding states. Here, we rely on Levenshtein ratio to score on both the grapheme and phoneme levels so as to compute a relevance measure, $\rho \in [0, 1]$. Intuitively speaking, it allows the α-β flow to be dampened in the event the rewrite is followed up with a semantically similar rephrase, indicating that it may not have quite achieved the customer's true intent. In a complementary fashion, the weight of the *discounting* edge $\gamma = 1 - \alpha \cdot \beta$ acts as a response whose magnitude correspond to how much the corresponding rewrite in its MST needs to be suppressed. Thus, the locally adaptive Markov model is **self-aware** to be able to tailor the decision boundary so as to surgically maximize the precision and recall over the space of rewrites.

5 Experiments

We build an evaluation dataset of request-rewrite pairs annotated by a cascaded labeling pipeline comprising of an interaction quality model, NLU scores and manual verification. This fundamentally enables us to surface, for a given request, u, both the set of rewrites which significantly improve the customer experience, $\mathbf{r}_u^+$ and the set that significantly worsen, $\mathbf{r}_u^-$ to collectively yield our core evaluation dataset, $\mathcal{D}_e$. Then, for any given request, we further define its *rewritability*, i.e. a binary label which indicates whether a particular request, u, should at all be rewritten, as $y_u = \mathbb{1}(|\mathbf{r}_u^+| > 0)$.

We benchmark our self-aware Markov model variant $\mathcal{M}_s$ against the baseline $\mathcal{M}_b$ (Ponnusamy et al., 2020) [2] and measure the gains introduced by our template-based generation strategy on both model variants, denoted by the subscript $+g$. Specifically, we measure their performance on the evaluation set $\mathcal{D}_e$ over three tasks, namely their ability to **partition** the requests based on their predicted *rewritability*, learn the optimal rewrite for a given request i.e. **equivalence learning**, and react to changing customer preferences i.e **reactivity rate**.

5.1 Partitioning

The automatic partitioning task is a binary classification problem where the ground truth label y_u is compared against the model prediction (Equation 3). We observe that the self-aware models significantly improve precision and recall compared to their baseline counterparts as shown in Table 1. Here, it is worth mentioning that the consistent

Model	$\mathcal{M}_{b+g}$	$\mathcal{M}_s$	$\mathcal{M}_{s+g}$
Precision	+0.0961	$+\mathbf{0.1808}$	+0.1688
Recall	+0.1724	+0.4674	$+\mathbf{0.5110}$
Accuracy	+0.0606	+0.1922	$+\mathbf{0.2047}$
F_1	+0.2555	+0.5547	$+\mathbf{0.5834}$

Table 1: Partitioning metrics measured as improvement over $\mathcal{M}_b$

significant gain in recall with template-based generation enabled is in part due to a strong correlating property between the need for rewriting and the need for disambiguation, which otherwise would have been lost due to the local Markov property.

5.2 Equivalence Learning

Once the requests are partitioned, the performance of the model in selecting rewrites i.e. its ability to optimally learn *equivalencies* for those in $\mathcal{H}^-$ are evaluated. To this end, we compare the score of the models (Φ_∞ from Equation 2) against the ground truth annotations in D_e i.e. whether a given rewrite candidate makes the customer experience significantly better (+1) or worse (-1). The precision-recall curves are then obtained as in Figure 3. The

[2] To the best of our knowledge, this is a novel space where widely peer-reviewed work on continual adaptive self-learning systems are few and far between. As such, this Markov-based baseline which has already shown to outperform a pointer-generator LSTM is chosen given its already established production impact.

self-aware models exhibit much better precision vs. recall trade-offs and have significantly higher areas under the curve. To highlight, the template augmented self-aware model $\mathcal{M}_{s+g}$ improves the PR-AUC by **27.45%** relative to $\mathcal{M}_{b+g}$.

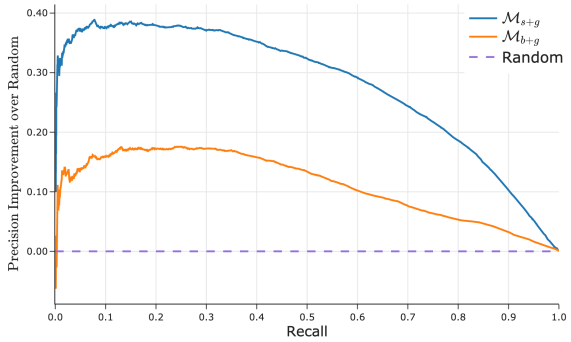

Figure 3: Precision-Recall Characteristics of Equivalence Learning.

5.3 Reactivity Rate

A key paradigm in designing large-scale AI solutions is the adaptability of the system to changing customer preferences. In the query rewriting domain, this quality can be expressed via the rate at which the top rewrite candidate changes over time i.e. the *reactivity rate*. Figure 4 shows the distribution of reactivity rate for common requests across the graph over a 30 day time period. The

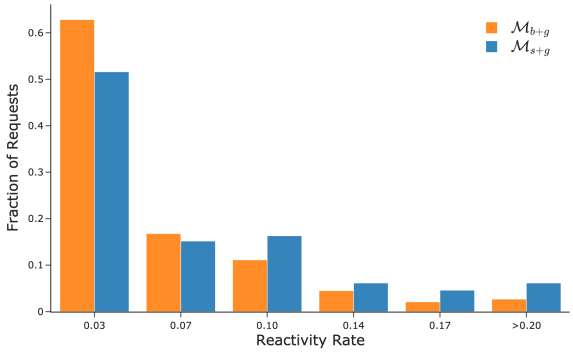

Figure 4: Reactivity Rate Distribution.

self-aware model exhibits higher reactivity as seen by the right shift in the distribution with respect to the baseline. To study the impact on performance over time, we compare the relative change in F_1 scores of the models $\Delta F_1^{(t)} = \frac{F_1^{(t)}}{F_1^{(0)}} - 1$ where, $F_1^{(t)}$ is the F_1 score of the given model at a given timestamp t on the equivalence learning task. It can be seen from Figure 5 that the self-aware model shows relative increase in the score over time, whereas the baseline is subject to a degradation in performance. Thus the higher reactivity rate of self-awareness is correlated to increased self-learning with the models adapting to customer feedback.

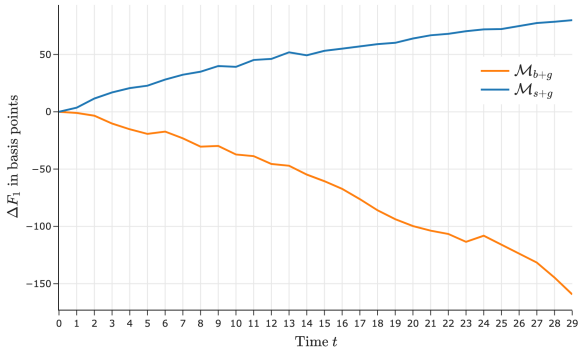

Figure 5: Relative change in F_1 score over time t. Note that for every timestamp, both models were retrained with new customer feedback.

5.4 Online Performance

With our approach for template-based generation being inherently scalable across languages and our self-aware Markov Graph naturally being language agnostic, we successfully deployed the model across 11 locales spanning 6 languages worldwide. To facilitate the models' ability to be continually adaptive, they are refreshed daily with new customer feedback. After nearly 6 weeks of in-depth A/B testing in production, we observed a strongly significant reduction (i.e. achieving a p-value of ≤ 0.0001) in defects experienced by the customers compared to the baseline (see Table 2) with a relative defect reduction of up to **31.22%**.

6 Deployment

In similar fashion to the well-established architecture of modern conversational AI systems (Gao et al., 2018), Alexa follows suit in which the user-spoken audio is first transcribed into an utterance text by an automatic speech recognition (ASR) system and thereafter has its domain, intent and entities inferred by the natural language understanding (NLU) system. However, with the presence of our reformulation engine as shown in Figure 6 below, the utterance text is intercepted so as to vend out a rewrite by means of an online database-backed lookup system before being funneled through to NLU. Thereafter, the resulting interpretation in context of the active dialog is leveraged to execute the corresponding action and respond back to the user.

Language	Defect Reduction	Example Request	Example Rewrite
English	25.78%	play tokyo take out	OLD: play tokyo **takedown** NEW: play **towkyo takeout by michael giacchino**
French	31.22%	mets la chanson le dimanche à bamako	OLD: mets **le** dimanche à bamako NEW: **joue la album dimanche** à bamako **par amadou**
Italian	23.98%	metti campioni del mondo	OLD: **metti la** canzone campion**i** del mondo NEW: **riproduci** canzone **italia** campion**e** del mondo **di gigione**
German	22.73%	spiel sun goes down von lenas x.	OLD: spiel sun goes down von lil nas **you** NEW: spiel sun goes down von lil nas **x**.
Spanish	28.06%	reproducir feliz cumpleaños de alejandro fernández	OLD: **pon las mañanitas con** alejandro fernández NEW: reproduc**e las mañanitas** de alejandro fernández
Portuguese	26.21%	toca mulher chorona	OLD: toca mulher chorona de **corpo e alma** NEW: toca**r** mulher chorona de **trio parada bruta**

Table 2: Online Performance of $\mathcal{M}_{s+g}$ with Qualitative Examples.

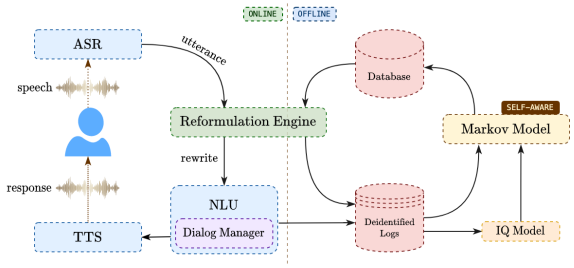

Figure 6: System Architecture

Within the offline data cycle, the de-identified logs are enriched with defect predictor labels by the interaction quality (IQ) model before being collectively used to train the self-aware Markov model. The resulting rewrites surfaced by the Markov model are successively uploaded to the aforementioned online database. It is worth noting here that the offline data cycle in entirely is executed on a daily cadence so as to ensure the overall reactivity of the system. In contrast to the baseline Markov Graph, training the self-aware model incurs a rather moderate ($\sim 8.33\%$) computational overhead due to the additional α computation and the increased amount of edges.

7 Conclusion

In this work, we address one of the key hurdles to the achieving self-learning in continuously updated feedback based systems, namely the deformation of the partitioning decision boundary due to lack of self-awareness. To overcome this degradation in Markov-based query rewriting models, we propose a superposition-based model that continually and reactively learns locally-adaptive decision boundaries, maximizing its precision and recall over time. Our proposed strategies show significant improvements in self-learning tasks and overcome long-term performance degradation. That being said, its dependence on sufficient statistical evidence for rewrite quality renders it subject to volatility with regard to tail or highly personalized rewrites, which we discuss further in the Appendix.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Ioannis Antonellis, Hector Garcia-Molina, and Chi-Chao Chang. 2008. Simrank++: Query rewriting through link analysis of the clickgraph (poster). In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, page 1177–1178, New York, NY, USA. Association for Computing Machinery.

Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 224–232, New York, NY, USA. Association for Computing Machinery.

Xing Fan, Eunah Cho, Xiaojiang Huang, and Chenlei Guo. 2021. Search based self-learning query rewrite system in conversational ai.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai.

Charles Miller Grinstead and James Laurie Snell. 2012. *Introduction to probability*. American Mathematical Soc.

Saurabh Gupta, Xing Fan, Derek Liu, Benjamin Yao, Yuan Ling, Kun Zhou, Tuan-Hung KPham, and Chenlei Guo. 2021. Robertaiq: An efficient framework for

automatic interactionquality estimation of dialogue systems.

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2020. Robust dialogue utterance rewriting as sequence tagging. *arXiv preprint arXiv:2012.14535*.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to rewrite queries. CIKM '16, page 1443–1452, New York, NY, USA. Association for Computing Machinery.

Bernard J Jansen, Danielle L Booth, and Amanda Spink. 2009. Patterns of query reformulation during web searching. *Journal of the american society for information science and technology*, 60(7):1358–1371.

Anton Khritankov. 2021. Hidden feedback loops in machine learning systems: A simulation model and preliminary results. *Lecture Notes in Business Information Processing*, page 54–65.

Daryl Lim, Julian McAuley, and Gert Lanckriet. 2015. Top-n recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 309–312.

Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2020. *Feedback Loop and Bias Amplification in Recommender Systems*, page 2145–2148. Association for Computing Machinery, New York, NY, USA.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6).

Evan Miller. 2015. https://www.evanmiller.org/bayesian-ab-testing.html.

Pragaash Ponnusamy, Alireza Roshan Ghias, Chenlei Guo, and Ruhi Sarikaya. 2020. Feedback-based self-learning in large-scale conversational ai agents. 34:13180–13187.

Pushpendre Rastogi, Arpit Gupta, Tongfei Chen, and Mathias Lambert. 2019. Scaling multi-domain dialogue state tracking via query reformulation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 97–105, Minneapolis, Minnesota. Association for Computational Linguistics.

Stefan Riezler and Yi Liu. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics*, 36(3):569–582.

Alireza Roshan-Ghias, Clint Solomon Mathialagan, Pragaash Ponnusamy, Lambert Mathias, and Chenlei Guo. 2020. Personalized query rewriting in conversational ai agents.

Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. WSDM '20, page 501–509, New York, NY, USA. Association for Computing Machinery.

D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Chen Shi, Yuxiang Hu, Zengming Zhang, Liang Shao, and Feijun Jiang. 2021. *User Feedback and Ranking In-a-Loop: Towards Self-Adaptive Dialogue Systems*, page 2046–2050. Association for Computing Machinery, New York, NY, USA.

Sukhdeep S. Sodhi, Ellie Ka-In Chio, Ambarish Jash, Santiago Ontañón, Ajit Apte, Ankit Kumar, Ayooluwakunmi Jeje, Dima Kuzmin, Harry Fung, Heng-Tze Cheng, Jon Effrat, Tarush Bali, Nitin Jindal, Pei Cao, Sarvjeet Singh, Senqiang Zhou, Tameen Khan, Amol Wankhede, Moustafa Alzantot, Allen Wu, and Tushar Chandra. 2021. Mondegreen: A post-processing solution to speech recognition error correction for voice search queries. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 3569–3575, New York, NY, USA. Association for Computing Machinery.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance rewriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31, Florence, Italy. Association for Computational Linguistics.

Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the human-recommender system feedback loop in collaborative filtering. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 645–651, New York, NY, USA. Association for Computing Machinery.

Siyang Yuan, Saurabh Gupta, Xing Fan, Derek Liu, Yang Liu, and Chenlei Guo. 2021. Graph enhanced query rewriting for spoken language understanding system. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7997–8001.

8 Appendix

8.1 Template-Based Generation

While most interactions are single-turn, i.e. closed-form requests that are information complete, there are nonetheless dialogs that serve to disambiguate

the user's intention. Such multi-turn interactions introduce conversational hierarchies, rendering each subsequent dialog turn contextually and cumulatively dependent on all its preceding turns. To ground this, consider the pair of requests – "*set an alarm for tomorrow*" and "*set an alarm for seven a. m.*". While the latter is informationally sufficient for the system to take the requisite action, the former in contrast remains ambiguous and warrants multiple turns. Under Markov conditions where the conditional distributions are entirely uni-variate, such hierarchies are not simultaneously observed by the model and fundamentally prevent it from providing an optimal rewrite.

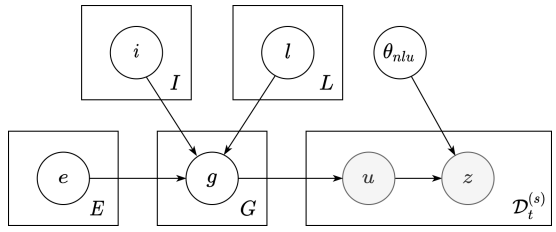

Figure 7: Plate notation summarizing the relationship between intents I, languages L, entity sets E, the corresponding templates G and the consequent utterances and confidences in the single-turn training dataset, $\mathcal{D}_t^{(s)} = \{(u, z)^{(1)}, \ldots, (u, z)^{(k)}\}$.

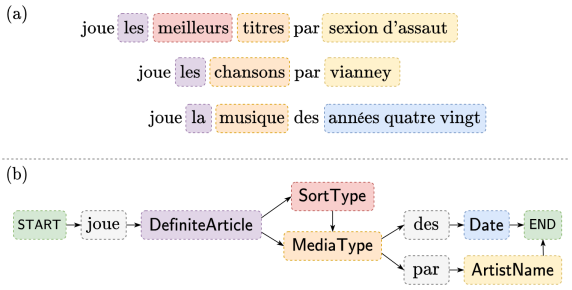

Figure 8: Template DAG extraction via NER and POS tagging with (a) showing multiple utterances with their entities and articles in colored boxes, and (b) representing the DAG for those utterances.

To address the limitation of the local Markov property in multi-turn dialogs, we introduce a synthetic utterance generation strategy that abridges the aforementioned hierarchy into a mere pair of turns. We define the single-turn training dataset $\mathcal{D}_t^{(s)}$ as described in the plate notation in Figure 7. We form the dataset of utterances u by sampling from a distribution of templates that are conditioned on entity sets, languages, and user intents. These templates are obtained by leveraging NER and POS tagging results from NLU, as shown in Figure 8a. Note, however, that a template g leads to utterances that are not enforced to follow a proper grammatical form—potentially reflecting a low NLU confidence z. Thus, for a specific entity set e, an intent i, and a language l, we determine the most plausible template g^* by maximizing the expected value of the NLU confidence z:

$$g^* = \underset{g \sim p_{g|e,i,l}}{\arg\max}\, E[z \mid g] \tag{5}$$

where $p_{g|e,i,l}$ denotes the sampling probability for the template g conditioned on its corresponding entity type, language, and intent. Once we have the set of templates for a given language and intent, we convert each template into a token chain and unify nodes across chains to form a single graph (see Figure 8b). Although this graph is constructed from high-quality templates, it may contain cycles that prevent a proper synthetic utterance generation. Therefore, we factorize the graph into multiple directed acyclic graphs (DAGs). We identify and break cycles using depth-first search to ensure directedness while preserving the syntactic integrity of the original linguistic structures. This process results in multiple DAGs that account for all the original valid paths.

When generating synthetic utterances, we extract the entities from a multi-turn dialog and obtain the template g^* that maximizes the overlap between its entity types e and the DAG nodes N_g:

$$\underset{g^* \in G^*_{(i,l)}}{\arg\max}\ |e \cap N_{g^*}| \tag{6}$$

where $G^*_{(i,l)}$ is the set of optimal templates that defines the DAG and (i, l) denotes a common intent and language across those templates. Once the path has been determined, we replace the entities in template $g^*_{(i,l)}$ with their corresponding values and resolve the entity articles, if applicable. It is possible, however, that the algorithm may not necessarily find a satisfactory path among the DAGs defined from $G^*_{(i,l)}$. In such cases, we abridge the entire dialog to merely retain the first turn of the dialog. Additionally, our algorithm is only executed when the multi-turn dialog has a successful conversion (i.e., the user's request was satisfied). In the event of an unsuccessful dialog or an abrupt end (e.g. "*no*", "*stop*"), we terminate the dialog with an interjectory utterance. Figure 2 describes the high-level process of compressing a multi-turn dialog into a single-turn dialog.

8.2 Meta-State Augmentation

The weight α is chosen in a hierarchical fashion as follows. We select the first α from the successive preference relation, $\alpha_c \succ \alpha_g \succ \alpha_e$ whose confidence interval widths given by Wilson's method for both the utterance and rewrite are lesser than η. Here, the Wilson's score interval is computed with a significance of 89% CI and η was calibrated via cross-validation to an optimal value of 0.588. Each of the α_c, α_g and α_e is defined by the following probability arguments,

$$\alpha_c = P(p_{W|c} > p_{X|c})$$
$$\alpha_g = P(p_W > p_X)$$
$$\alpha_e = P(p_{W_e} > p_{X_e})$$

where α_c relies on the supporting statistics for a given customer, c while α_g extends that statistic globally across all customers in the data. Unlike α_c and α_g, however, we determine α_e by the distributions of entity changes between the utterance and the rewrite. Given the entity set e, along with their corresponding changes between the original and its rewrite (e.g., **ArtistName** added, **SongName** changed, etc.), we compute α_{e_i} for every entity $e_i \in e$ and retrieve the maximum absolute deviation as α_e:

$$\alpha_e = \max_{e_i \in e} |\alpha_{e_i} - 0.5| \tag{7}$$

We choose the maximum absolute deviation because it linearly provides a sense of dispersion without overly weighting values as in other formulations (e.g., standard deviation). More importantly, Equation 7 defines α_e based on a single most-dispersed α_{e_i} value, which can lead to either suppress (i.e. low dispersion) or encourage (i.e. high dispersion) the $\alpha\beta$-path.

8.3 Risks and Limitations

In order to be locally adaptive i.e. decisively unroll or discount a particular rewrite when warranted so, the learning of the Graph hinges on its ability to determine the *viability* i.e. the α value of the said rewrite—the performance of which is squarely correlated with that of the IQ model and thereby inheriting the model's limitations in its overall precision and recall. That being said, the Graph does internally rely on its collaborative filtering ability to regularize the model's decision while external guard-rail mechanisms are also in place to further mitigate the impact of this dependency.

Another matter of concern here would be the requisite for sufficient statistics when computing α, which becomes a limiting factor for highly tail or personalized rewrites, where the Graph would essentially struggle to learn a consistent decision boundary given a high entropy of plausible rewrite alternatives, resulting in its equivalency learning to be entirely contingent on the more prevalent cohort within each learning cycle. In practice however, this is far from being a considerable issue as the over-arching system takes on a multi-stage hierarchical approach that permits other personalized agents to act in lieu of the Graph, while maintaining the Graph's role for its more confident set of customer cohorts.

Conversely speaking, should there be a significantly widespread rewrite that abruptly becomes defective, the Graph would inevitably require a substantial or quite possibly, an equally voluminous source of negative feedback to counter the highly successful prior. This in turn could subject a vast number of customers to a bad experience for a considerable amount of time that ultimately drives down the engagement. As clear of a risk this is in a deployed application setting, a veritable solution here would be to adopt a sense of recency-weighting in constructing the Graph's adjacency matrix, which stands as a worthwhile future effort. In the meantime however, we rely on external gating mechanisms that refresh far more often than the Graph to aid in mitigating the overall severity of such an issue.