Reward Constrained Interactive Recommendation with Natural Language Feedback

IIC3633 - Sistemas Recomendadores

Benjamín Farías, Benjamín Lepe, Juan Romero

Noviembre, 2021

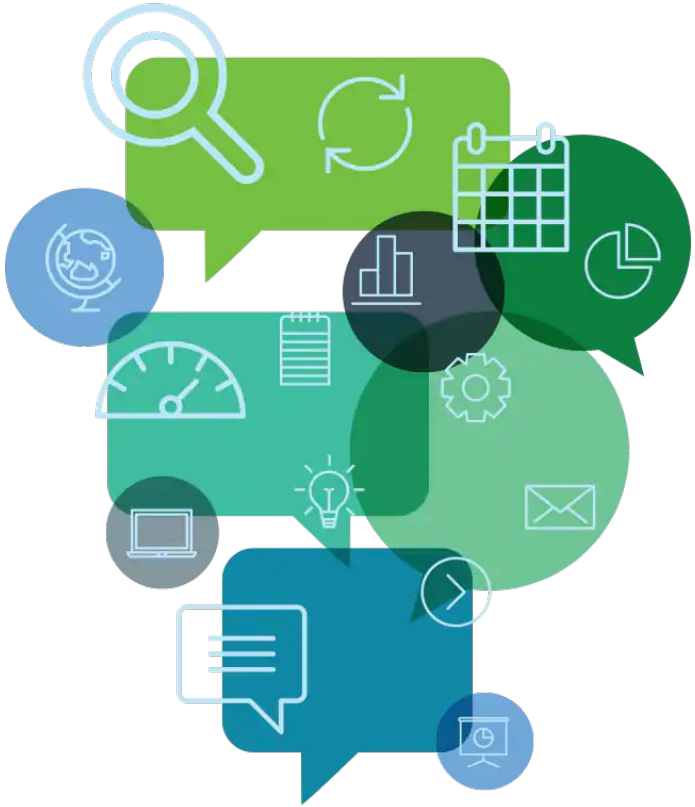
Outline

1. **Contexto, Problema y Contribución**
2. Marco Teórico y Estado del Arte
3. Solución: Reward Constrained Recommendation (RCR)
4. Evaluación y Conclusiones

Contexto

Uno de los desafíos de implementar un sistema recomendador es cómo obtener/interpretar el feedback del usuario respecto a los ítems recomendados.

Entre muchas técnicas, se propone el uso del **lenguaje natural** para poder obtener una valoración de los ítems por parte del usuario.



Problema

Si bien el lenguaje natural es preciso para describir la percepción del usuario sobre el ítem, los sistemas actuales pueden **perder la memoria** al no restringirse por el historial de recomendaciones y por ende, ignorar feedback antiguo del usuario al realizar nuevas recomendaciones.

Contribución

Se propone un modelo que usa el lenguaje natural, basado en **restricciones dinámicas** que actúan como un crítico del sistema recomendador y son capaces de generalizar adecuadamente. Además, se muestran resultados empíricos que evidencian un rendimiento superior a los métodos actuales.

Outline

1. Contexto, Problema y Contribución
2. **Marco Teórico y Estado del Arte**
3. Solución: Reward Constrained Recommendation (RCR)
4. Evaluación y Conclusiones

Reinforcement Learning

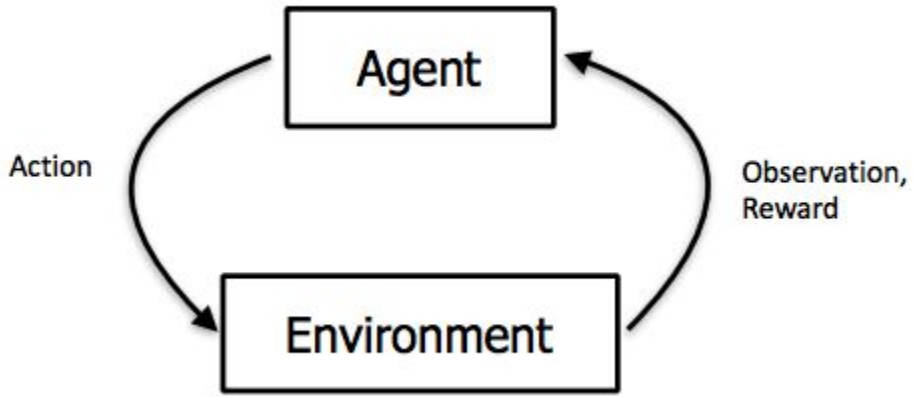
El sistema toma una secuencia de decisiones a lo largo del tiempo, moviéndose entre distintos estados. Por cada decisión el sistema recibe una recompensa según lo beneficioso que sea el estado alcanzado. La idea del modelo es maximizar el valor esperado de estas recompensas.

$$J_R(\pi) = \sum_{t=1}^{\infty} \mathbb{E}_{P,\pi} \left[r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right] \qquad [1]$$

$$\max_{\pi \in \Pi} J_R(\pi), \quad \text{s.t. } J_C(\pi) \leq \alpha \qquad [2]$$

Text-Based Recommendations (RL)

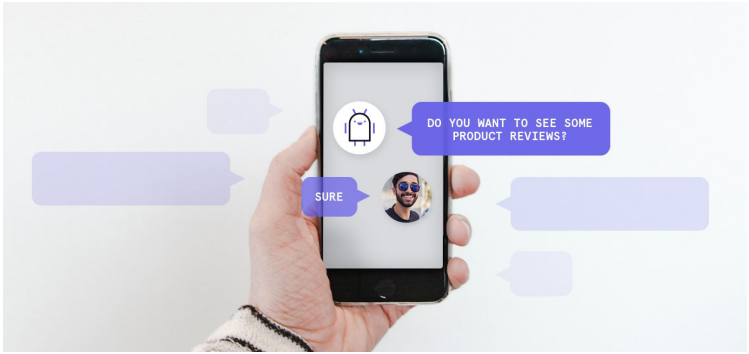
El sistema interpreta el lenguaje natural para poder **usar dicho feedback** al momento de calcular la recompensa para el modelo de aprendizaje reforzado.



Estado del Arte

El uso de *Constrained Policy Optimization* se ha visto en diversas áreas, tales como el análisis de redes eléctricas y la robótica [3].

Se han desarrollado sistemas recomendadores interactivos basados en texto, sin embargo, sus restricciones **no son modeladas de forma explícita.**



Estado del Arte

Existen modelos basados en redes adversarias con aprendizaje reforzado para poder generar texto. Sin embargo, el enfoque ha priorizado la calidad y diversidad del texto, **sin definir restricciones sobre su contenido** directamente [4].

Outline

1. Contexto, Problema y Contribución
2. Marco Teórico y Estado del Arte
3. **Solución: Reward Constrained Recommendation (RCR)**
4. Evaluación y Conclusiones

Reward Constrained Recommender

Objetivo: Recomendar ítems visuales al usuario en base al feedback textual entregado en las recomendaciones pasadas.

Problema Encontrado: Métodos convencionales de RL pueden ignorar el feedback pasado entregado por el usuario **(ver imagen).**

Solución Propuesta: Considerar un modelo de optimización de política de aprendizaje (RL), **sujeto a las restricciones obtenidas desde el feedback textual del usuario (RC).**

Constrained Policy Optimization

F.O. para **Recomendar:**

$$J_R(\pi_\theta) = \sum_{t=1}^{\infty} \mathbb{E}_{P,\pi_\theta} \left[r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right], \text{ s.t. } J_C(\pi_\theta) \leq \alpha$$

F.O. para **Discriminar:**

$$L(\phi) = -\mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim p_f} \left[\log(C_\phi(\boldsymbol{s},\boldsymbol{a})) \right] - \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a})\sim p_r} \left[\log(1 - C_\phi(\boldsymbol{s},\boldsymbol{a})) \right]$$

F.O. para **Recomendar (relajada):**

$$\min_{\lambda \geq 0} \max_{\theta} L(\lambda, \theta, \phi) = \min_{\lambda \geq 0} \max_{\theta} \left[J_R(\pi_\theta) - \lambda \cdot (J_{C_\phi}(\pi_\theta) - \alpha) \right]$$

Constrained Policy Optimization

El **recomendador** busca maximizar la recompensa minimizando las restricciones incumplidas.

El **discriminador** aprende a detectar el cumplimiento de las restricciones en los ítems recomendados.

Los **multiplicadores de Lagrange** indican el grado de impacto de la penalización del discriminador.

Algorithm 1 Reward Constrained Recommendation

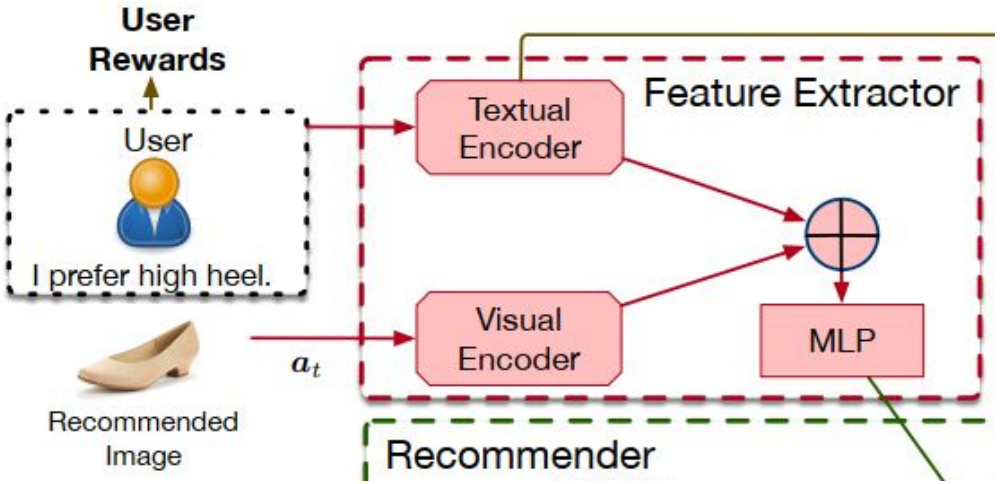
Input: constraint $C(\cdot)$, threshold α, learning rates $\eta_1(k) > \eta_2(k) > \eta_3(k)$
Initialize recommender and discriminator parameters with pretrained ones, Lagrange multipliers $\lambda_0 = 0$
repeat
  for $t = 0, 1, ..., T-1$ **do**
    Sample action $a_t \sim \pi$, observe next state s_{t+1}, reward r_t and penalties c_t
    $\hat{R}_t = r_t - \lambda_k c_t$
    Recommender update with (8)
  end for
  Discriminator update with (9)
  Lagrange multiplier update with (10)
until Model converges
return recommender (policy) parameters θ

Arquitectura RCR

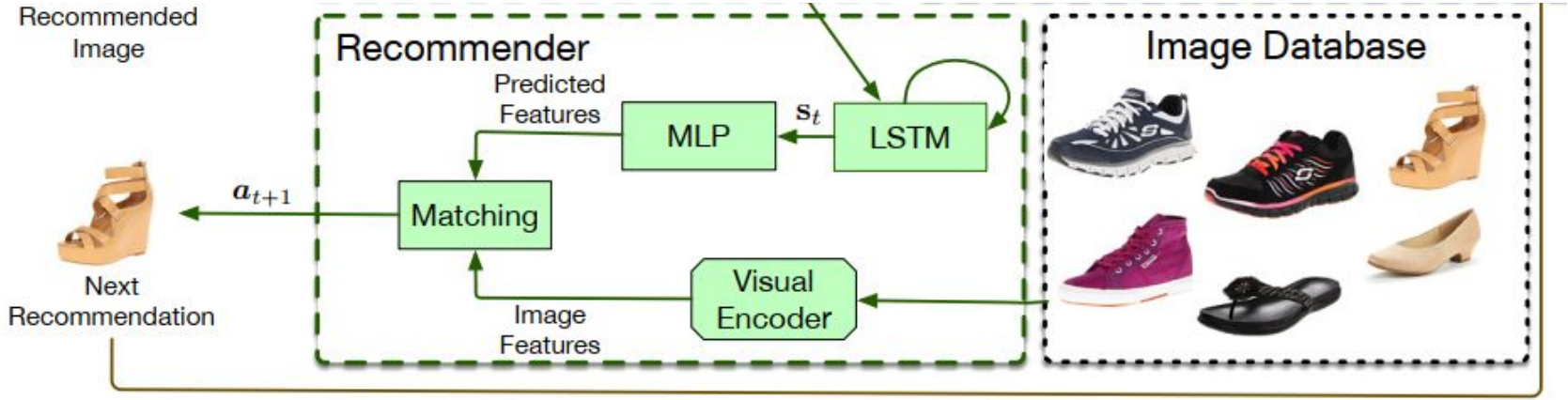
El **extractor de características** se encarga de codificar el ítem recomendado y el feedback del usuario, que corresponden al input.

El **ítem visual** se codifica pasándolo por una CNN y posteriormente una red de atributos.

El **feedback textual** se codifica mediante una capa de embedding y una RNN de tipo LSTM.



Arquitectura RCR

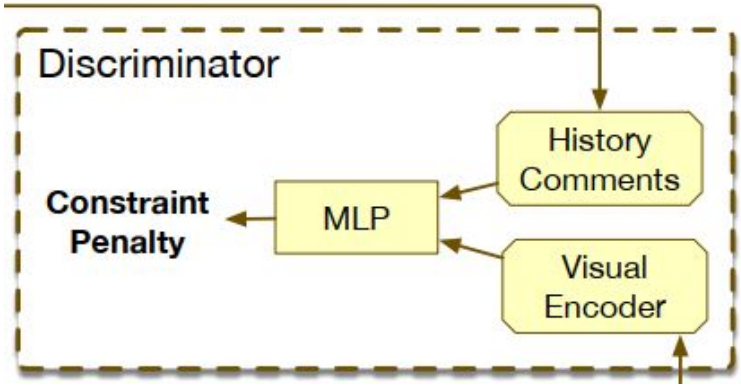


El **recomendador** recibe las codificaciones de entrada y predice las características deseadas por el usuario por medio de una red LSTM + MLP. En base a estas predicciones, entrega los ítems más cercanos a dichas características **(matching).**

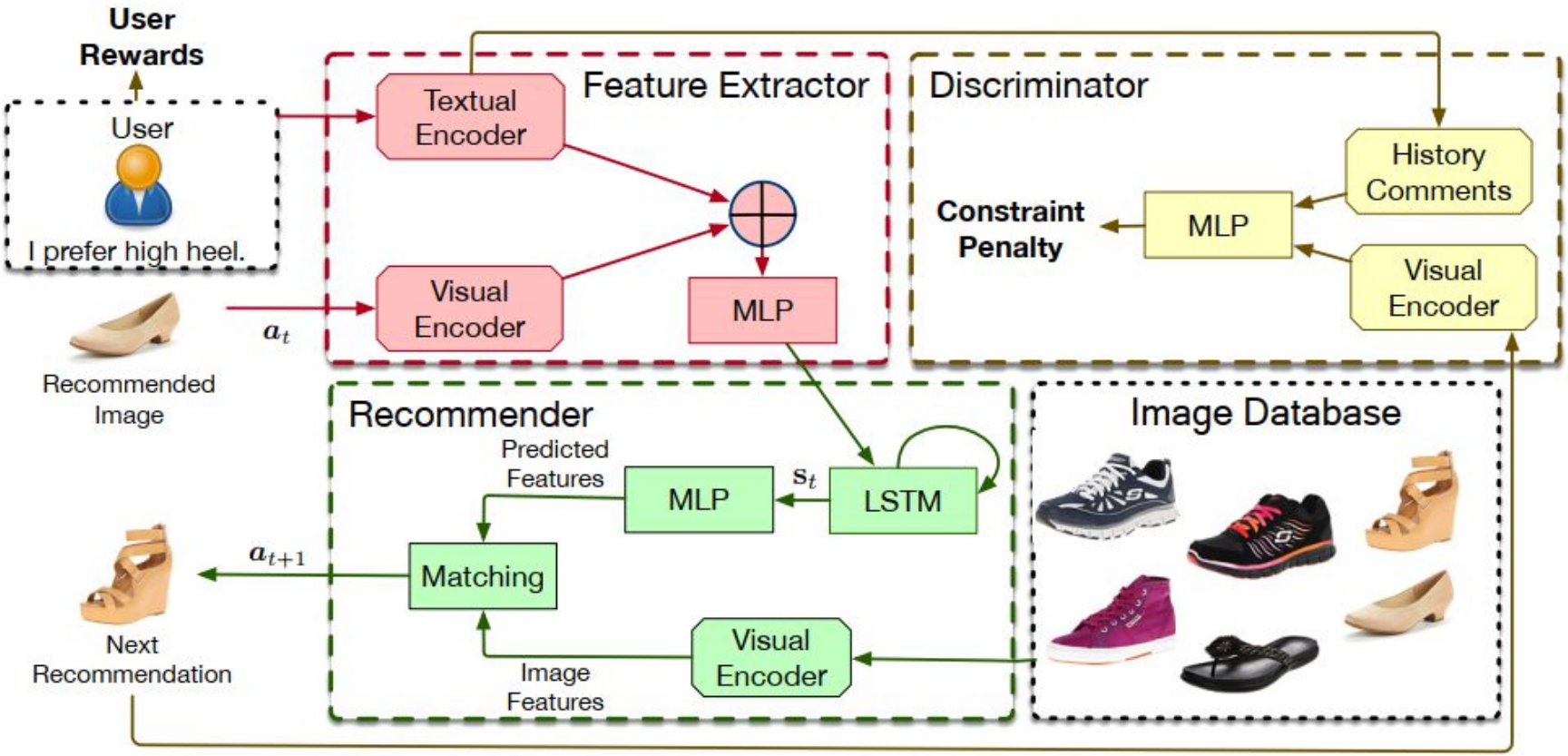
Arquitectura RCR

El **discriminador** se encarga de detectar ítems recomendados actualmente que rompen las restricciones obtenidas a través del feedback pasado del usuario.

Aquellas recomendaciones que no cumplan con las restricciones con una alta probabilidad serán **descartadas.**



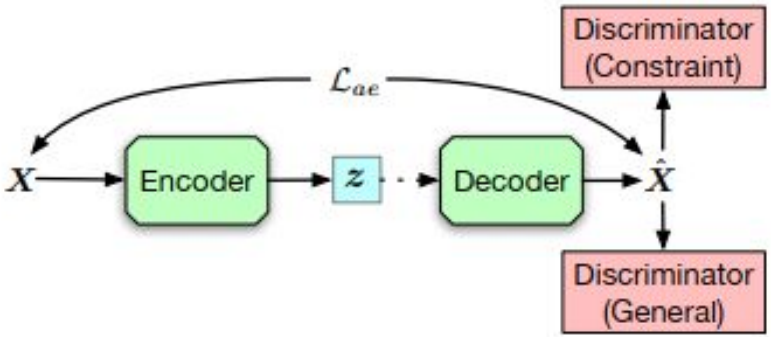
Arquitectura RCR



Extensión: Constrained Text Generation

Se puede extender el modelo RCR propuesto para el problema de **generación de texto sujeto a restricciones.**

Esto permite, por ejemplo, generar nuevas reviews a partir de un dataset entregado, pero satisfaciendo la restricción de que los textos generados sean reviews **estrictamente positivas.**



Outline

1. Contexto, Problema y Contribución
2. Marco Teórico y Estado del Arte
3. Solución: Reward Constrained Recommendation (RCR)
4. **Evaluación y Conclusiones**

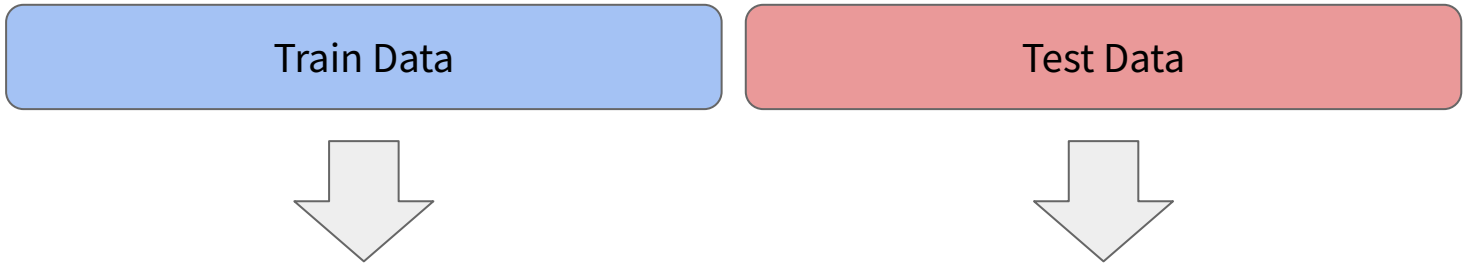
Text-Based Interactive Recommendation

Se utilizó el **dataset** *UT-Zappos50K*, que contiene:

- **50.025** imágenes de zapatos
- **Atributos** sobre cada zapato tales como categoría, altura del taco, cierre, etc.

http://vision.cs.utexas.edu/projects/finegrained/utzap50k/

Text-Based Interactive Recommendation

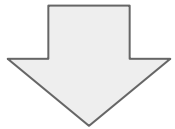


Corresponden a ítems que vienen con todos sus **atributos etiquetados.**

Corresponden a ítems que se asumen como nuevos, y por ende, traen sus **atributos vacíos**.

Text-Based Interactive Recommendation

User comments generator



Se entrenó con **10.000** pares de imágenes, en donde en cada par hay un ítem recomendado y uno deseado.

Text-Based Interactive Recommendation

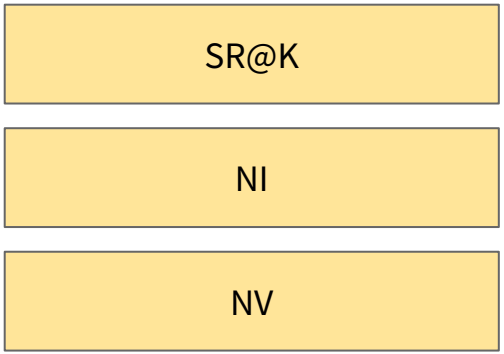
$$r_t = -||\texttt{ResNet}(\boldsymbol{a}_t) - \texttt{ResNet}(\boldsymbol{a}^*)||_2 - \lambda_{att}||\texttt{AttrNet}(\boldsymbol{a}_t) - \texttt{AttrNet}(\boldsymbol{a}^*)||_0$$

Para obtener el **reward** se buscará minimizar la diferencia entre el ítem recomendado a_t y el deseado a*. Por lo tanto, se debe intentar **maximizar** la función anterior.

Text-Based Interactive Recommendation

Para la evaluación se utilizaron 3 **métricas:**

1. **SR@K:** Task success rate después de K interacciones.
2. **NI**: número de interacciones antes de tener éxito.
3. **NV**: número de atributos violados.



Métricas

Text-Based Interactive Recommendation

	SR@10 ↑	SR@20 ↑	SR@30 ↑	NI ↓	NV ↓
RL (Unseen)	19%	44%	63%	26.75 ± 1.67	70.02 ± 6.20
RL + Naive (Unseen)	52%	83%	94%	12.72 ± 0.93	16.47 ± 2.75
RCR (Unseen)	74%	86%	94%	10.91 ± 1.06	11.32 ± 1.98
RCR (Seen)	78%	91%	92%	10.34 ± 1.18	12.25 ± 2.99

Tabla de comparación entre las **evaluaciones** obtenidas por distintos métodos. **RCR (Seen)** corresponde al modelo evaluado en el set de training.

Text-Based Interactive Recommendation

(a) y **(b)** son casos **exitosos** del modelo propuesto **(RCR)**, mientras que **(c)** es un caso de **falla** para el modelo RL tradicional.

Lo anterior se debe a que RCR tiene una mejor generalización que RL en cuanto a las restricciones impuestas por el usuario.

Constrained Text Generation

	Test-BLEU-2	3	4	5	Self-BLEU-2	3	4	VR
RL	0.807	0.622	0.469	0.376	0.658	0.315	0.098	40.36%
RCR (ours)	0.840	0.651	0.492	0.392	0.683	0.348	0.151	10.49%

Usando un dataset de Yelp, se entrenó un **generador de reseñas** con la restricción de que solo fuesen **comentarios positivos**. El modelo propuesto logró un mejor desempeño que los tradicionales, mostrando una mejor calidad en los textos generados.

RCR también mostró una gran disminución en la tasa de violaciones de los atributos **(VR)**, lo que era de esperarse dada su alta efectividad en el manejo de restricciones.

Conclusiones

- RCR es un framework que nos permite obtener las ventajas del **aprendizaje reforzado**, pero aplicadas a la optimización basada en **restricciones dinámicas.**
- El modelo **mejora** notablemente en comparación con los modelos base de RL.
- En un futuro se planea incluir **información histórica** del usuario para mejorar las recomendaciones.
- El framework es **general**, por lo que puede ser aplicado en otras áreas.

Referencias

[1] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. MIT press, 2018.

[2] Eitan Altman. Constrained Markov decision processes. CRC Press, 1999.

[3] Joshua Achiam, David Held, Aviv Tamar and Pieter Abbeel. Constrained policy optimization. In ICML, 2017.

[4] Lantao Yu, Weinan Zhang, Jun Wang and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In AAAI, 2017.