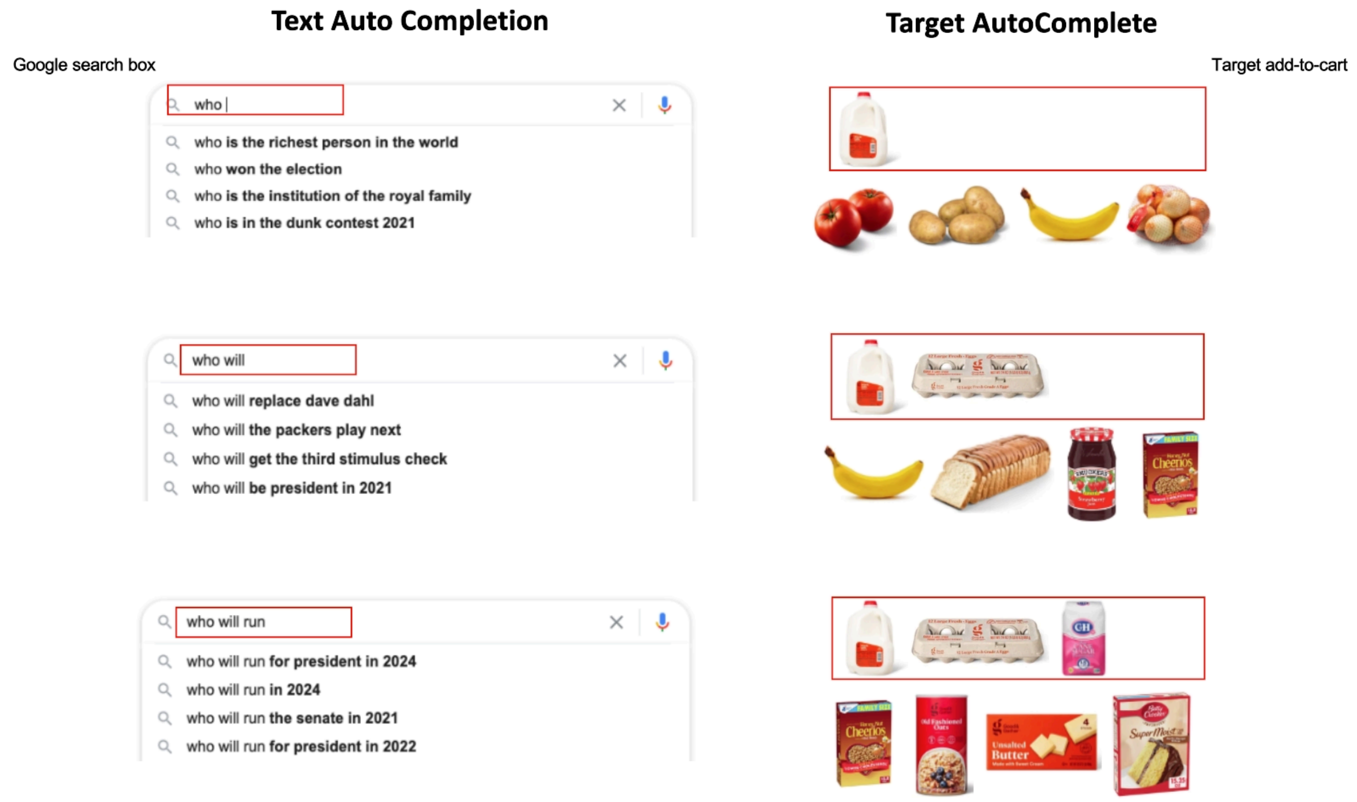
Target AutoComplete: Real Time Item Recommendations at Target

July 25, 2023

Introduction

Our data science team developed a state-of-the-art technology that looks at item(s) already added to a guest's online basket and recommends items they might be interested in buying. Target AutoComplete (TAC) is an AI model that infers recommendations for dynamic sessions in real time. It is inspired by the text auto completion feature found in search engines. Text auto completion recommends complete sentence queries based on the words already entered in the search box. This not only saves time but also makes the search experience effortless and comfortable for users.


Text Auto Completion
Target AutoComplete
Google search box
Target add-to-cart
who
who is the richest person in the world
who won the election
who is the institution of the royal family
who is in the dunk contest 2021
who will
who will replace dave dahl
who will the packers play next
who will get the third stimulus check
who will be president in 2021
who will run
who will run for president in 2024
who will run in 2024
who will run the senate in 2021
who will run for president in 2022

How Text Auto Completion is like Target AutoComplete

TAC falls under the ‘session-based recommendation’ problem, where session implies a list of items added to cart by the guest. When a guest adds an item into their cart, the model provides relevant recommendations based on historical patterns of items purchased together. The model gives importance to the ordering of items and the recency of items: more recent items in the sequence are given more importance.

How is TAC Different from Existing Item Recommendation Models?

There are two novel aspects that make TAC stand out from the existing item recommendation models at Target:

1: TAC recommends based on multiple items *cumulatively*. This is unique at Target because other models make recommendations based on single items.

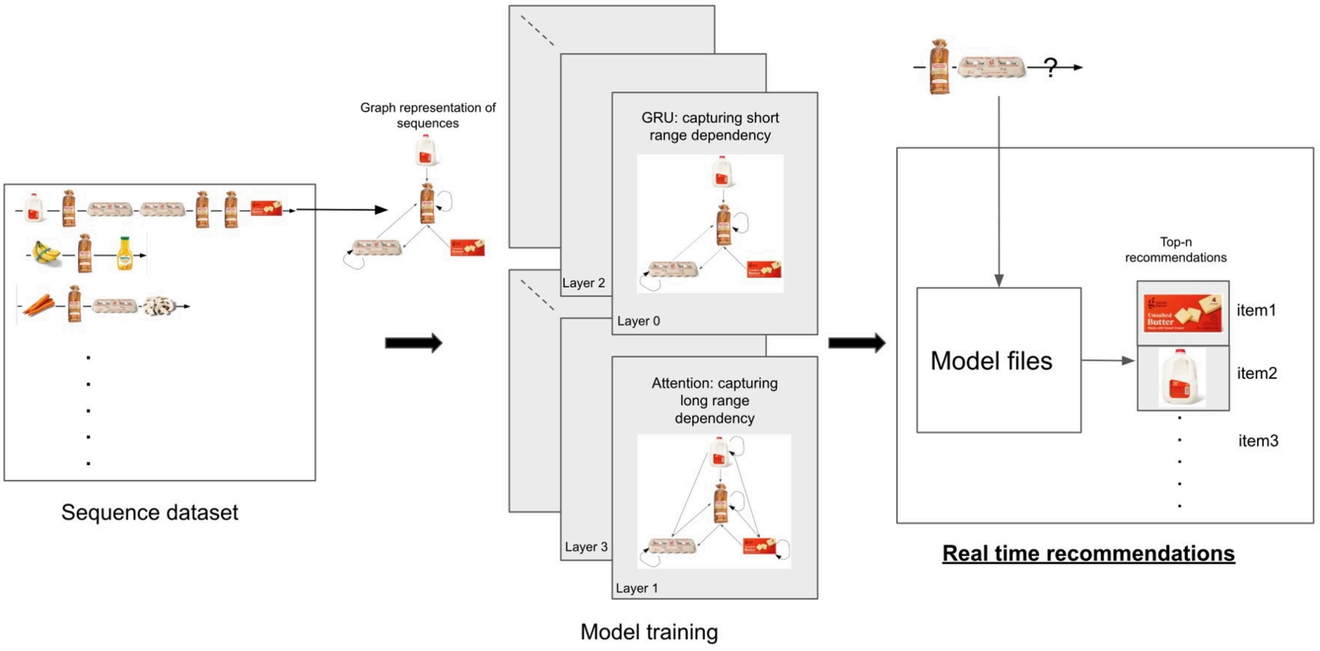
To explain this with an example: Suppose a guest intends to buy 3 items in a session. When milk is added as the first item, recommendations are items *related to milk* - like almond milk, butter, and yogurt. When eggs get added next as the second item, existing models tend to recommend items related to eggs- like egg whites, brown eggs, 12 count eggs, 18 count eggs. However, TAC smartly considers the context of existing cart items (milk and eggs) to recommend *breakfast items* - like bread, cereal, and coffee. When sugar gets added as the third item, existing models would recommend items related to sugar like stevia sweetener, granulated sugar, and organic sugar. TAC would consider the basket context and recommend *baking items* - like cake flour, frosting, and baking powder.

2: TAC is the first machine learning model at Target that does *real time inferencing* where recommendations are predicted in real time, instead of pre-computing recommendations for individual items and looking them up.

To explain this with an example: Existing recommendations are based on single items. For milk, recommendations are related to only milk; for eggs recommendations are related to only eggs and so on. These static recommendations are pre-computed and stored beforehand. They are looked up when a guest adds the item to the basket. TAC, however, tries to understand the context of the basket as it evolves in real time. Consider the two items: milk and eggs. Real time inferencing means one kind of item gets recommended when *eggs are added after milk* (recommendation would be bacon, a breakfast item more related to recently added eggs), which would be different from items recommended when *milk is added after eggs* (recommendation would be coffee, a breakfast item more related to recently added milk). These kinds of dynamic recommendations cannot be pre-computed using existing models because there are trillions of permutations for the entire Target catalog which would take years to pre-compute. The attention-based aggregator ensures that: a) recommendations capture the overall session context, and b) more importance is assigned to recently added items in the session.

TAC - Under the Hood

Let us take a deep dive into the technical aspects of the model. The TAC architecture can be divided into three parts:


Graph representation of sequences
Sequence dataset
GRU: capturing short range dependency
Layer 2
Layer 0
Attention: capturing long range dependency
Layer 3
Layer 1
Model training
?
Top-n recommendations
Model files
item1
item2
item3
Real time recommendations

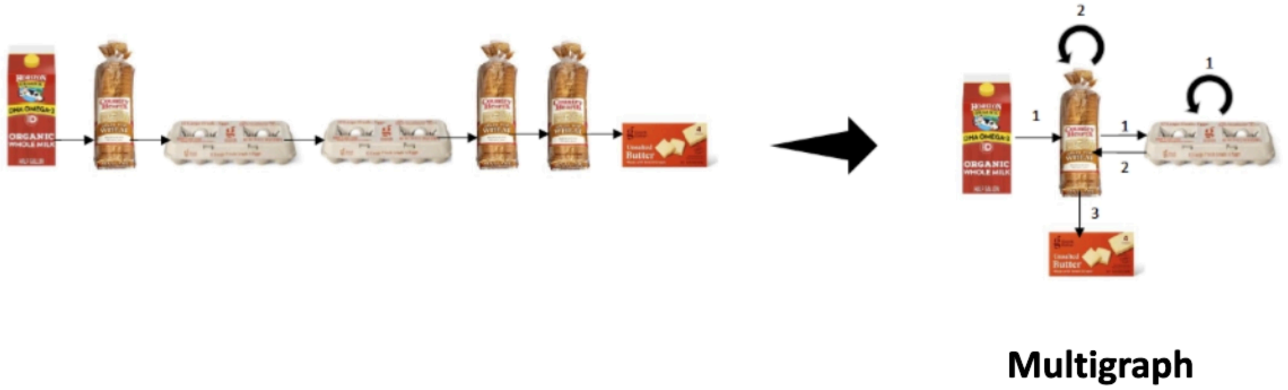
TAC Architecture

Data processing: We use Apache Hive and Spark to generate ordered sequences of items which represent a guest's add-to-cart items.

Model training: The sequences are fed into interleaved graph neural networks (GNN) [1] and attention layers to generate item embeddings and learnable weight parameters as model files. A more in-depth explanation of model training is as follows:

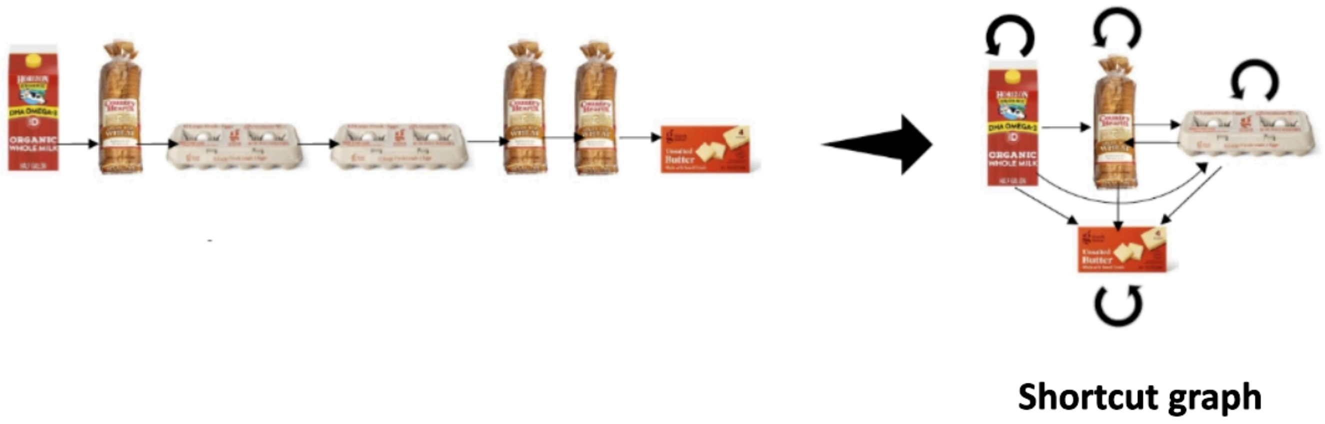
First, ordered sessions are implemented as graphs where nodes are unique items and edges represent how they transition. The model has two graph representations:

-Representing sessions as multigraph for graph neural networks: A multigraph is a graph that allows parallel edges between two nodes. Multigraph preserves the ordering of items which ensures that items closer together in the session have similar embeddings compared to items farther apart, thus capturing short-range dependencies between items. In the example below of a multigraph representation, the outgoing edge (start item to end item) number denotes the order in which the end item occurred in the session after the start item. Ex: Bread occurred 3 times in the sequence with 3 outgoing edges (first item being eggs, second being bread, and third being butter). A Gated Recurrent Unit (GRU) [2] is used to propagate item features to generate item embeddings.



Multigraph representation of a add-to-cart session

-Representing sessions as shortcut graph for Attention model: A shortcut graph is a graph wherein for each ordered pair of items (item_a, item_b) in the session we create an edge from item_a to item_b only if the pair is in the form (item_t1, item_t2) where t1 and t2 denotes position of the item in the session and t1 <= t2. An attention layer propagates item features through the shortcut edges to capture long-range dependencies between items.



Shortcut graph representation of a add-to-cart session

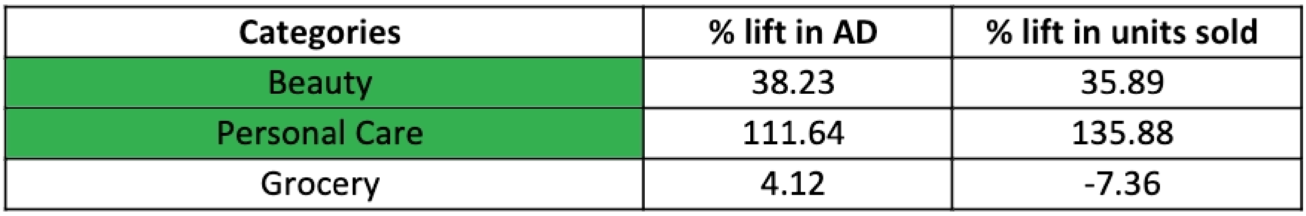
Real time inferencing: Python microservices use the model files to generate session embeddings for the guest adding item(s) to their cart. As the session embedding gets updated with each item, the microservice generates top-recommendations in real time of items whose embeddings are closest to the session embedding.

A/B Test Results and Analysis

TAC went into production in Q4 2022 for our most frequent item categories of grocery, health, and beauty products. The variable saw a lift of 2.05% in click through rate (CTR), 1.38% in conversion rate, and 6.33% in attributable demand (AD). TAC went into production for all other categories in Q1 2023. An improvement of 4.4% in conversion rate and 5.8% in attributable demand was noted. Category-level analysis of the results shows that there are some categories where TAC has scope to improve.

A/B Test Results and Analysis

TAC went into production in Q4 2022 for our most frequent item categories of grocery, health, and beauty products. The variable saw a lift of 2.05% in click through rate (CTR), 1.38% in conversion rate, and 6.33% in attributable demand (AD). TAC went into production for all other categories in Q1 2023. An improvement of 4.4% in conversion rate and 5.8% in attributable demand was noted. Category-level analysis of the results shows that there are some categories where TAC has scope to improve.



Categories	% lift in AD	% lift in units sold
Beauty	38.23	35.89
Personal Care	111.64	135.88
Grocery	4.12	-7.36

Category level analysis of TAC A/B test. Green rows represent categories where TAC succeeded.

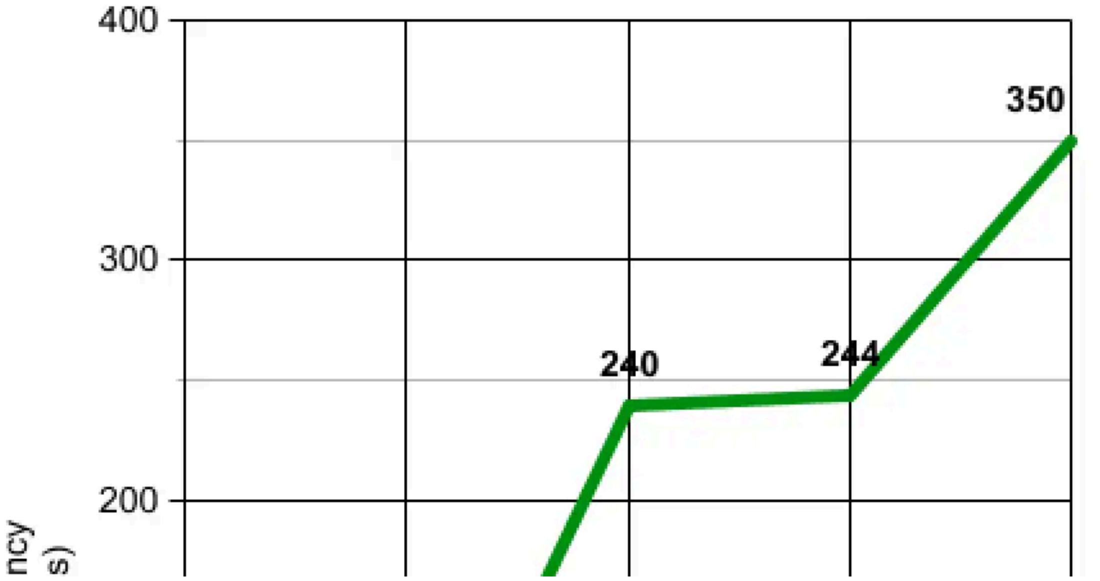
Conclusion

TAC has successfully pushed the frontiers of machine learning and data sciences at Target. It is a step towards Target's efforts to build real time personalization algorithms. TAC will also be used by our automated "Target Run" guest experience team to power Basket Aware recommendations in the to go tab in our mobile app.

As part of future work, our TAC model will be tuned so it out-performs in every category, and it will be tested against strategies in placements that use items viewed data, another form of sessions.

References
[1] Merritt, Rick, “What are Graph Neural Networks?” (2022). (link)
[2] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." *arXiv preprint arXiv:1409.1259* (2014).

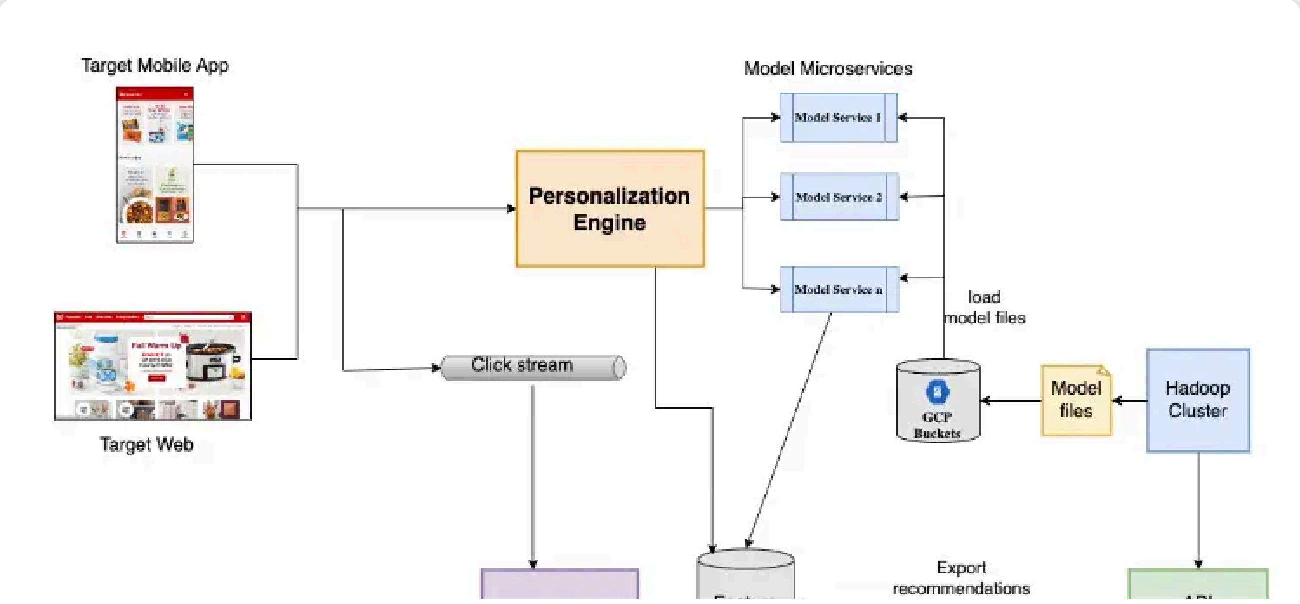
RELATED POSTS



Using BERT Model to Generate Real-time Embeddings

By Pushkar Chennu and Amit Pande, March 23, 2022

How we chose and implemented an effective model to generate embeddings in real-time. Target has been exploring, leveraging, and releasing open source software for…



Real-Time Personalization Using Microservices

By Amit Pande, Prathyusha Kanmanth Reddy, and Pushkar Chennu, May 11, 2023

How Target's Personalization team uses microservices to improve our guest experience

PUBLISHED BY

Bhavtosh Rath

Lead Data Scientist - DS Guest and Digital (Item PRZ)

CATEGORIES

Artificial Intelligence Data Sciences Innovation

SHARE

Privacy policy

Terms & conditions

Team member services

RSS Feed