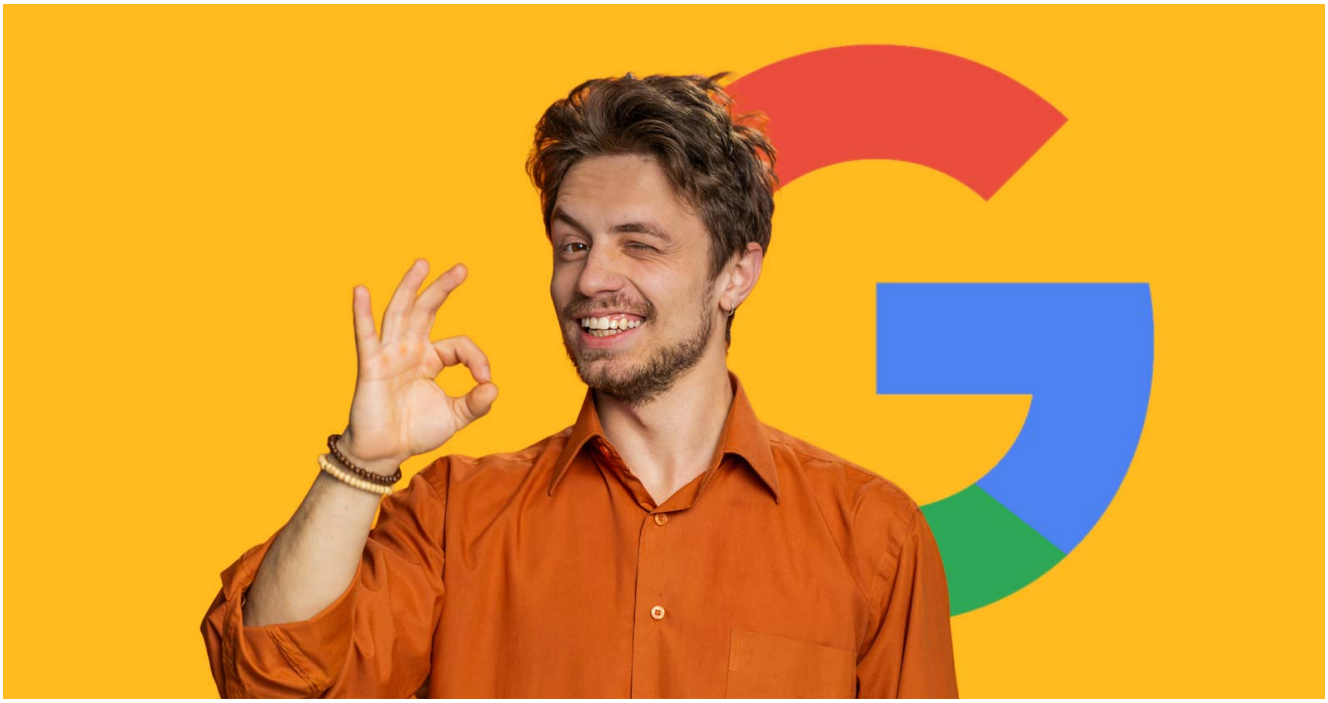
How Recommender Systems Like Google Discover May Work via @sejournal, @martinibuster



Google Discover is largely a mystery to publishers and the search marketing community even though Google has published official guidance about what it is and what they feel publishers should know about it. Nevertheless, it's so mysterious that it's generally not even considered as a recommender system, yet that is what it is. This is a review of a classic research paper that shows how to scale a recommender system. Although it's for YouTube, it's not hard to imagine how this kind of system can be adapted to Google Discover.

Recommender Systems

Google Discover belongs to the class of systems known as a recommender systems. A classic recommender system I remember is the MovieLens system from way back in 1997. It is a university science department project that allowed users to

rate movies and it would use those ratings to recommend movies to watch. The way it worked is like, people who tend to like these kinds of movies tend to also like these other kinds of movies. But these kinds of algorithms have limitations that make them fall short for the scale necessary to personalize recommendations for YouTube or Google Discover.

Two-Tower Recommender System Model

The modern style of recommender systems are sometimes referred to as the Two-Tower architecture or the Two-Tower model. The Two-Tower model came about as a solution for YouTube, even though the original research paper (Deep Neural Networks for YouTube Recommendations) does not use this term.

It may seem counterintuitive to look to YouTube to understand how the Google Discover algorithm works, but the fact is that the system Google developed for YouTube became the foundation for how to scale a recommender system for an environment where massive amounts of content are generated every hour of the day, 24 hours a day.

It’s called the Two-Tower architecture because there are two representations that are matched against each other, like two towers.

In this model, which handles the initial “retrieval” of content from the database, a neural network processes user information to produce a user embedding, while content items are represented by their own embeddings. These two representations are matched using similarity scoring rather than being combined inside a single network.

I’m going to repeat that the research paper does not refer to the architecture as a Two-Tower architecture, it’s a description for this kind of approach that was created later. So, while the research paper doesn’t use the word tower, I’m going to continue using it as it makes it easier to visualize

what’s going on in this kind of recommender system.

User Tower
The User Tower processes things like a user’s watch history, search tokens, location, and basic demographics. It uses this data to create a vector representation that maps the user’s specific interests in a mathematical space.

Item Tower
The Item Tower represents content using learned embedding vectors. In the original YouTube implementation, these were trained alongside the user model and stored for fast retrieval. This allows the system to compare a user’s “coordinates” against millions of video “coordinates” instantly, without having to run a complex analysis on every single video each time you refresh your feed.

The Fresh Content Problem

Google’s research paper offers an interesting take on freshness. The problem of freshness is described as a tradeoff between exploitation and exploration. The YouTube recommendation system has to balance between showing users content that is already known to be popular (exploitation) versus exposing them to new and unproven content (exploration). What motivates Google to show new but unproven content, at least for the context of YouTube, is that users show a strong preference for new and fresh content.

The research paper explains why fresh content is important:

> *“Many hours worth of videos are uploaded each second to YouTube. Recommending this recently uploaded (“fresh”) content is extremely important for YouTube as a product. We consistently observe that users prefer fresh content, though not at the expense of relevance.”*

This tendency to show fresh content seems to hold true for

Google Discover, where Google tends to show fresh content on topics that users are personally trending with. Have you ever noticed how Google Discover tends to favor fresh content? The insights that the researchers had about user preferences probably carry over to the Google Discover recommendation system. The takeaway here is that producing content on a regular basis could be helpful for getting web pages surfaced in Google Discover.

An interesting insight in this research paper, and I don't know if it's still true but it's still interesting, is that the researchers state that machine learning algorithms show an implicit biased toward older existing content because they are trained on historical data.

They explain:

> *"Machine learning systems often exhibit an implicit bias towards the past because they are trained to predict future behavior from historical examples."*

The neural network is trained on past videos and they learn that things from one or two days ago were popular. But this creates a bias for things that happened in the past. The way they solved the freshness issue is when the system is recommending videos to a user (serving), this time-based feature is set to zero days ago (or slightly negative). This signals to the model that it is making a prediction at the very end of the training window, essentially forcing it to predict what is popular right now rather than what was popular on average in the past.

Accuracy Of Click Data

Google's foundational research paper also provides insights about implicit user feedback signals, which is a reference to click data. The researchers say that this kind of data rarely

provides accurate user satisfaction information.

The researchers write:

> *"Noise: Historical user behavior on YouTube is inherently difficult to predict due to sparsity and a variety of unobservable external factors. We rarely obtain the ground truth of user satisfaction and instead model noisy implicit feedback signals. Furthermore, metadata associated with content is poorly structured without a well defined ontology. Our algorithms need*
> *to be robust to these particular characteristics of our training data."*

The researchers conclude the paper by stating that this approach to recommender systems helped increase user watch time and proved to be more effective than other systems.

They write:

> *"We have described our deep neural network architecture for recommending YouTube videos, split into two distinct problems: candidate generation and ranking.*
> *Our deep collaborative filtering model is able to effectively assimilate many signals and model their interaction with layers of depth, outperforming previous matrix factorization approaches used at YouTube.*
>
> *We demonstrated that using the age of the training example as an input feature removes an inherent bias towards the past and allows the model to represent the time-dependent behavior of popular of videos. This improved offline holdout precision results and increased the watch time dramatically on recently uploaded videos in A/B testing.*
>
> *Ranking is a more classical machine learning problem yet our deep learning approach outperformed previous linear and tree-based methods for watch time prediction. Recommendation*

> *systems in particular benefit from specialized features describing past user behavior with items. Deep neural networks require special representations of categorical and continuous features which we transform with embeddings and quantile normalization, respectively.”*

Although this research paper is ten years old, it still offers insights into how recommender systems work and takes a little of the mystery out of recommender systems like Google Discover. Read the original research paper: Deep Neural Networks for YouTube Recommendations

Featured Image by Shutterstock/Andrii Iemelianenko

https://www.searchenginejournal.com/how-recommender-systems-like-google-discover-may-work/565515/