

Technical Disclosure Commons

Defensive Publications Series

December 2020

Entity Suggestions in Custom Audiences

Jane Huang

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

Huang, Jane, "Entity Suggestions in Custom Audiences", Technical Disclosure Commons, (December 08, 2020)

https://www.tdcommons.org/dpubs_series/3854



This work is licensed under a [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/).

This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Entity Suggestions in Custom Audiences

Traditional audience targeting includes broad terms that can be under-inclusive. For instance, a broad term of “male” may miss women interested in the product. Similarly, some advertisers select similar entities (e.g. “pink running shoes” and “blue running shoes”, and “running shoes for women”), but those selections can result in an inefficient overlap between audience groups of each selection. Advertisers should thus be able to specify the closest and most efficient keywords of their audience, such as “running shoes for women.” Similarly, an advertiser can specify keywords like “running shoes” if they want to sell Shoe Brand 1 shoes. However, while advertisers may classify an ad, their classification may not be accurate. It may be difficult for the advertiser to set up the classification before testing the advertisement.

The present disclosure can target specific audiences by generating target keywords from three neural network towers. The first tower is the user tower, which can include data such as watch history, search history, browsing history, browser usage, and behavioral and search history of individual users identified by device ID or user account ID. The user tower can retrieve a user ID, convert it into a 1-hot embedded vector, then shrink the vector to a denser encoding such as a 16-bit float. User browsing, search history, watch history, and device information are separate float values, and are similarly shrunk down to a denser encoding such as a 16-bit float. The output can be a combination of these encodings, such as a 128-bit float. The second tower is the advertiser, which can include an advertiser ID and features such as a landing page, website, or other campaigns. Similar to the user tower, data associated with the advertiser tower can be 1-hot encoded and converted to concatenated vector vertical

categories matched with creatives such as ads (eg. a video about running shoes). The third tower is the entity, which can include interests, keywords, URLs, applications, or locations to classify the audience. Each record, such as a first record for a keyword for shoes and a second record for a keyword for San Francisco, can be associated with one entity rather than a bundle of entities.

A model based on the three towers may offer improved advertising prediction performance over existing models. For instance, some models may predict the probability of converting an advertisement based on the user tower and the advertiser tower. However, adding the entity tower can improve the accuracy of the calculated probability by considering user intent, such as by calculating a probability of conversion between the user and an advertiser for certain service or product. For instance, one target may be able to provide multiple products, and each product can reach different people.

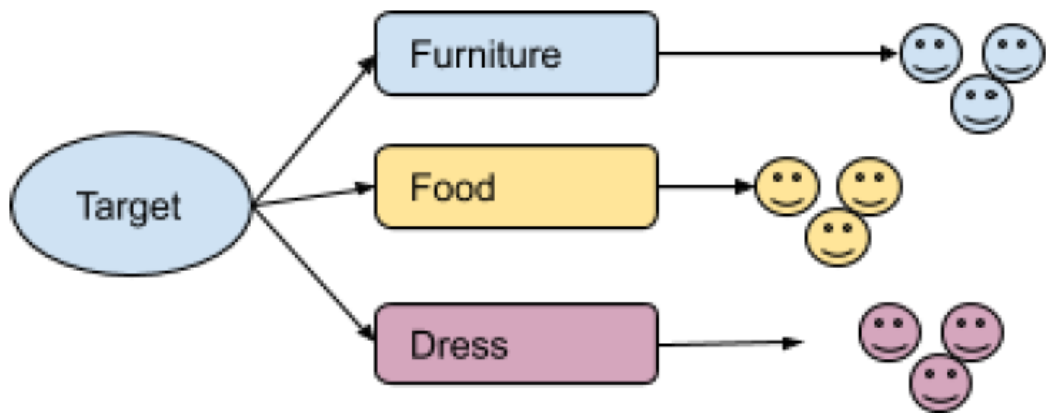


Figure 1: Example of a target providing multiple products reaching different people

Some models may predict the probability of converting an advertisement based on the user tower and the entity tower. However, adding the advertiser tower can improve the accuracy of the calculated probability by additionally having a prediction between an advertiser and the user. For instance, a keyword can represent a different brand affinity for different users.

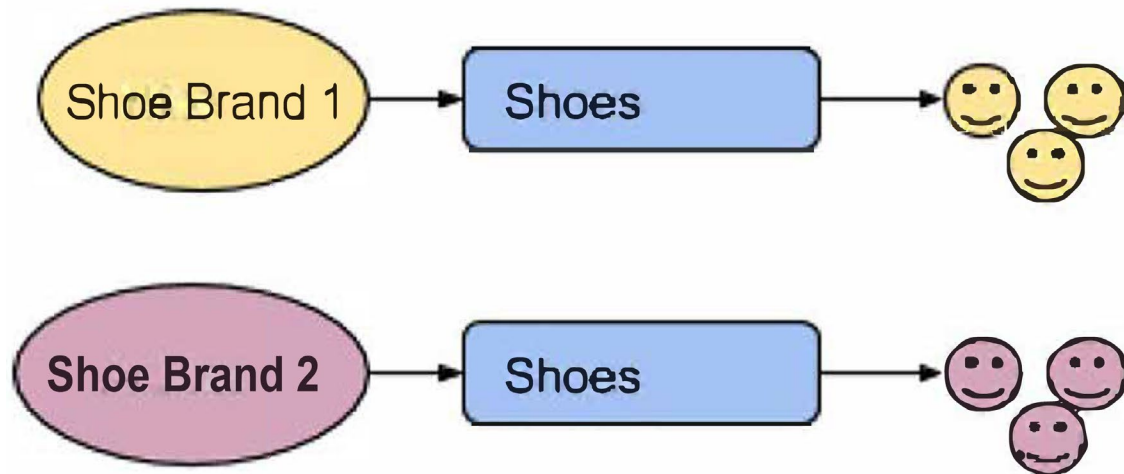


Figure 2: Different brand affinity of different users over the same keyword “shoes”

Each tower can be a deep neural network (DNN) training on the data corresponding to each tower. The DNN can analyze user activities such as whether a user made a purchase with a particular advertiser or past search history. For instance, the DNN can analyze a user’s history of searching for running shoes, Shoe Brand 1 shoes, and horses before making a purchase for Shoe Brand 1 shoes from a particular merchant. The DNN can use semantic filtering to reduce the space and determine which past searches are relevant. The advertiser tower can also be replaced with sub-advertiser towers. Each sub-advertiser tower can represent different towers for campaign or ad groups. For instance, a department store can divide into departments such as “men’s clothing” and “women’s clothing” to treat each department as a different advertiser.

The DNN of each tower can output a float of equal size. For instance, each tower can output a 32-bit float. The equal bit sizes of the float allows for a 3-way join of the towers as a distance function between the three towers.

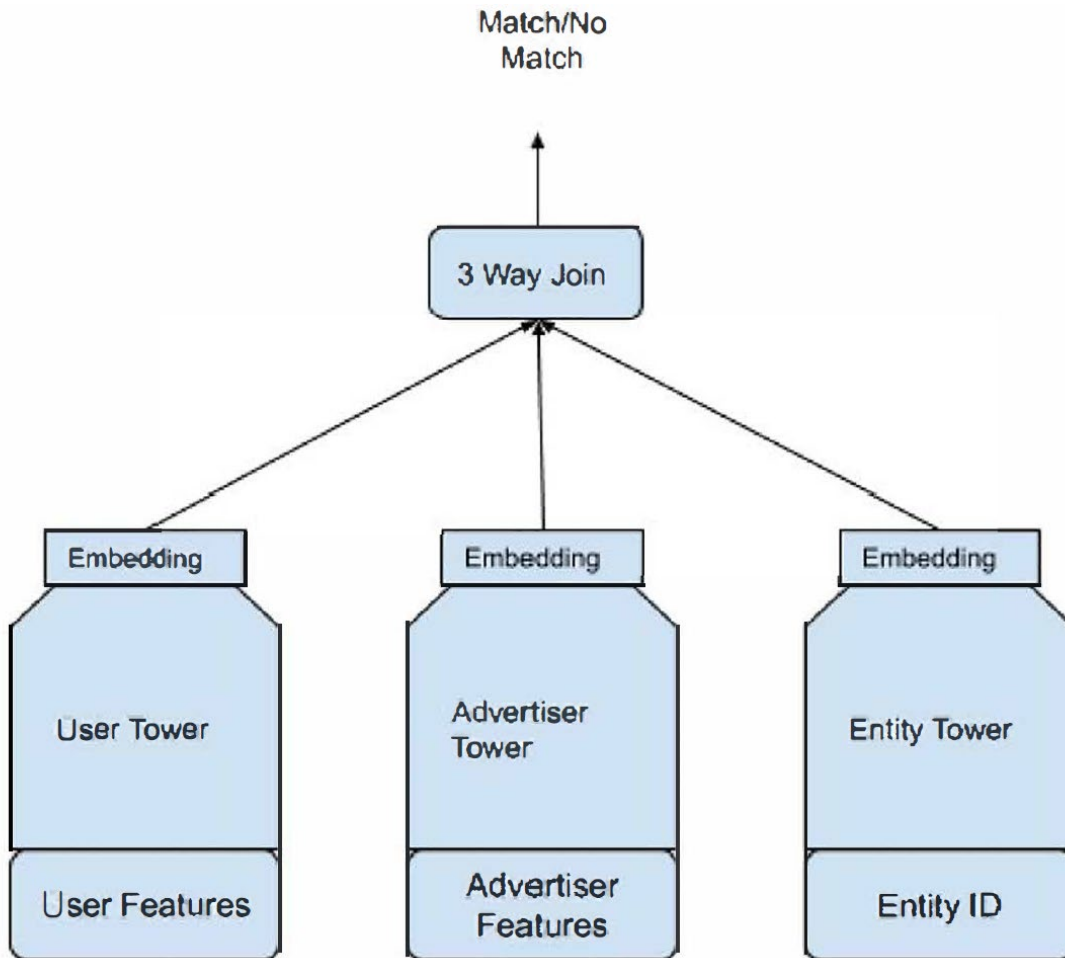


Figure 3: Applying distance function on three towers

Joining three towers can be defined by a distance function, D:

$$D = \frac{U * A * E}{||U|| * ||A|| * ||E||}$$

Figure 4: Joining three towers

The distance function includes U as the user embedding, A as the advertiser embedding, and E as the entity embedding. The distance function can output a distance measurement, which is compared to a threshold matches between users, advertisers, and entities. Based on the distance function, positive labeling can correspond to a conversion event between a user and an advertiser can be associated with an entity. Negative labels can correspond to random sampling. The best match based on the threshold can be aligned in the same

space. Since the output labels can have binary classes, a sigmoid function can be adopted.

In some embodiments, the user and entity embedding already exist, so the DNN can apply the distance function against the advertiser tower.

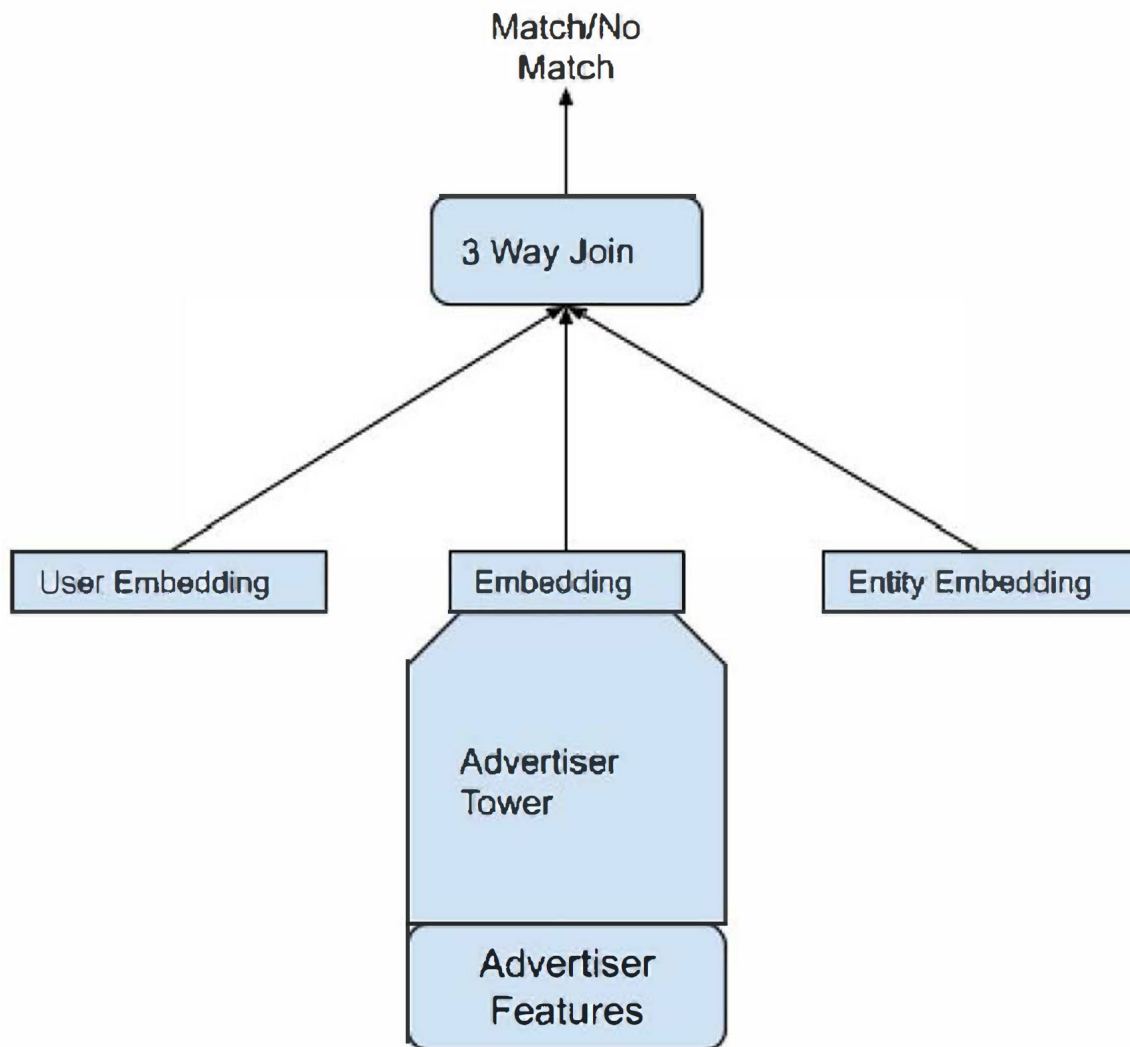


Figure 5: Applying distance function on an advertiser tower given existing User and Entity embedding

If the user and entity embedding already exist, the inputs to the distance function can be pre-trained user and entity embedding, as well as the advertiser tower including advertiser features such as advertiser IDs, advertiser verticals, and campaign information.

The distance function can output a distance measurement, which can be compared to a

threshold to determine matches between users, advertisers, and entities. Based on the distance function, positive labeling can correspond to a conversion event between a user and an advertiser can be associated with an entity. Negative labels can correspond to random sampling. The best match based on the threshold can be aligned in the same space. Since the output labels can have binary classes, a sigmoid function can be adopted.

Based on the matches, the present disclosure can recommend or auto-create campaign keywords or create audience groups based on nearest entities. Advertisers can then retrieve keywords corresponding to top performing entities for their campaign. For instance, an entity based on the embedding representation can be provided to the advertiser based on a k-nearest neighbors (KNN) algorithm. The distance functions used in the KNN algorithm can be a loss function such as a cosine similarity or a Poisson. For instance, the cosine similarity function can be represented as:

$$D(A, E) = \langle A, E \rangle / (\|A\| \times \|E\|)$$

Figure 6: Cosine similarity function

In the cosine similarity function, A is the advertiser tower and E is the entity tower.

One privacy concern is the significant user data required from signed in users, but those users have opted-in by creating an account and signing in. Moreover, the user data cannot be narrowed down to target individual users. For instance, if a single user made a purchase on a strange website, an advertiser cannot identify them through entity selection because of the reduction to the 32-bit float and also because the distance function will rule out very narrow campaigns. The custom audience can also specify a minimum number of users included in an entity to avoid hyper targeting.

Another privacy concern is theoretically identifying a user from conversion type data. The training sample is based on conversions, such as whether a user has made a conversion. Therefore, an attributed conversion, such as when an advertiser had a conversion with a particular user, can be attributed back to a particular ad and click event. Accordingly, attributed conversions are not shared between advertisers. Unattributed conversions, such as when a user made a purchase but it is unknown which ad, if any, they clicked on, cannot be attributed to a user but can be attributed to an advertiser. Accordingly, unattributed conversions are not shared between advertisers.

Abstract

A data processing system can generate associations between user interests and target users based on advertiser features to allow for specific targeting of users with content such as media or advertisements. The target users can be associated with user behaviors such as, for example, client device capabilities, user browsing behavior and actions, or a combination thereof. The advertiser features can be, for example, campaign or ad groups, landing pages, websites, or a combination thereof. The user interests can be keywords, URLs, applications, and locations, or a combination thereof. The user interests, target users, and advertiser features can each be represented by a deep neural network. The data processing system can train the deep neural network to match conversion events between target users and advertiser features through user interests. The data processing system can then retrieve embedding representations of specific campaigns based on advertiser features.