miniproject

Foodahaulics-food recommender system

data:

Food features

Food id	name	tags
1	food1	tag1,tag2,tag3
2	food2	tag4,tag5
3	food3	tag3,tag6,tag7,tag8

User-food interaction

User id	Food id	rating
1	20	4
1	38	2
1	47	5
2	56	2
2	57	4

Food tags: tags are mapped to numbers and substituted to get vector form

{" 'beverages'": 0, " 'quick-breads'": 1, " 'low-calorie'": 2, " 'dips'": 3, " 'yeast'": 4, " 'fish'": 5, " 'broil'": 6, " 'meat'": 7, " 'chocolate'": 8, " 'gumbo'": 9, " 'new-years'": 10, " 'brown-bag'": 11, " 'pork'": 12, " 'low-saturated-fat'": 13, " 'pasta'": 14, " 'bar-cookies'": 15, " 'copycat'": 16, " 'tomatoes'": 17, " 'winter'": 18, " 'preparation'": 19, " 'north-american'": 20, " 'seasonal'": 21, " 'diabetic'": 22, " 'greens'": 23, " 'low-fat'": 24, " 'beans'": 25, " 'to-go'": 26, " 'easy'": 27, " 'high-calcium'": 28, " 'pineapple'": 29, " 'nuts'": 30, " 'valentines-day'": 31, " 'taste-mood'": 32, " 'shellfish'": 33, " 'beef'": 34, " 'peppers'": 35, " 'tempeh'": 36, " 'apples'": 37, " 'occasion'": 38, " 'african'": 39, " 'beef-ribs'": 40, " 'dinner-party'": 41, " 'squid'": 42, " 'indonesian'": 43, " 'pacific-northwest'": 44, " 'spicy'": 45, " 'gluten-free'": 46, " 'tuna'": 47, " 'brunch'": 48, " 'infant-baby-friendly'": 49, " 'romantic'": 50, " 'tropical-fruit'": 51, " 'asian'": 52, " 'healthy'": 53, " 'sandwiches'": 54, " 'vegetarian'": 55, " 'german'": 56, " 'main-dish'": 57, " '15-minutes-or-less'": 58, " 'collard-greens'": 59, " 'comfort-food'": 60, " 'vegetables'": 61, " 'fall'": 62, " 'chili'": 63, " 'st-patricks-day'": 64, " 'weeknight'": 65, " 'salmon'": 66, "'bacon'": 67, " 'mixer'": 68, " 'pies-and-tarts'": 69, " 'crock-pot-slow-cooker'": 70, " 'independence-day'": 71, " 'lasagna'": 72, " 'californian'": 73, " 'potatoes'": 74, " 'polish'": 75, " 'simply-potatoes2'": 76, " 'novelty'": 77, " 'marinades-and-rubs'": 78, " 'lactose'": 79, " 'sweet'": 80, " 'soy-tofu'": 81, " 'refrigerator'": 82, " 'creole'": 83, " 'steam'": 84, " 'eggs'": 85, " 'appetizers'": 86, " 'cuisine'": 87, " 'main-ingredient'": 88, " 'european'": 89, " 'no-cook'": 90, " 'smoothies'": 91, " 'low-sodium'": 92, " 'kwanzaa'": 93, " 'served-hot'": 94, " 'low-in-something'": 95, " '5-ingredients-or-less'": 96, " 'grilling'": 97, " 'wild-game'": 98, " 'italian'": 99, " 'canning'": 100, " 'british-columbian'": 101, " 'low-protein'": 102, " 'drop-cookies'": 103, " 'food-processor-blender'": 104, " 'christmas'": 105, " 'healthy-2'": 106, " 'stews'": 107, " 'very-low-carbs'": 108, " 'low-carb'": 109, " 'mushrooms'": 110, " 'pork-sausage'": 111, " 'free-of-something'": 112, " 'squash'": 113, " 'asparagus'": 114, " 'chinese'": 115, " 'eggs-dairy'": 116, " 'ramadan'": 117, " 'pancakes-and-waffles'": 118, " 'oamc-freezer-make-ahead'": 119, " 'stove-top'": 120, " 'high-protein'": 121, " 'french'": 122, " 'cheese'": 123, " 'pork-chops'": 124, " 'microwave'": 125, " 'beginner-cook'": 126, " 'for-large-groups'": 127, " 'course'": 128, " 'spreads'": 129, " 'gifts'": 130, "'time-to-make'": 131, " 'inexpensive'": 132, " 'pasta-rice-and-grains'": 133, " 'one-dish-meal'": 134, " 'lunch'": 135, " 'northeastern-united-states'": 136, " 'broccoli'": 137, " 'mango'": 138, " 'deep-fry'": 139, " '3-steps-or-less'": 140, " 'vegan'": 141, " 'lettuces'": 142, " 'lentils'": 143, " 'soups-stews'": 144, " 'southern-united-states'": 145, " 'swedish'": 146, " 'turkey'": 147, " 'berries'": 148, " 'spring'": 149, " 'kosher'": 150, " 'rolled-cookies'": 151, " 'cheesecake'": 152, " 'leftovers'": 153, " 'dietary'": 154, " 'pies'": 155, " '30-minutes-or-less'": 156, " 'condiments-etc'": 157, "'lactose'": 158, " 'summer'": 159, " 'super-bowl'": 160, " 'cakes'": 161,....... 269 tags}

Food names:

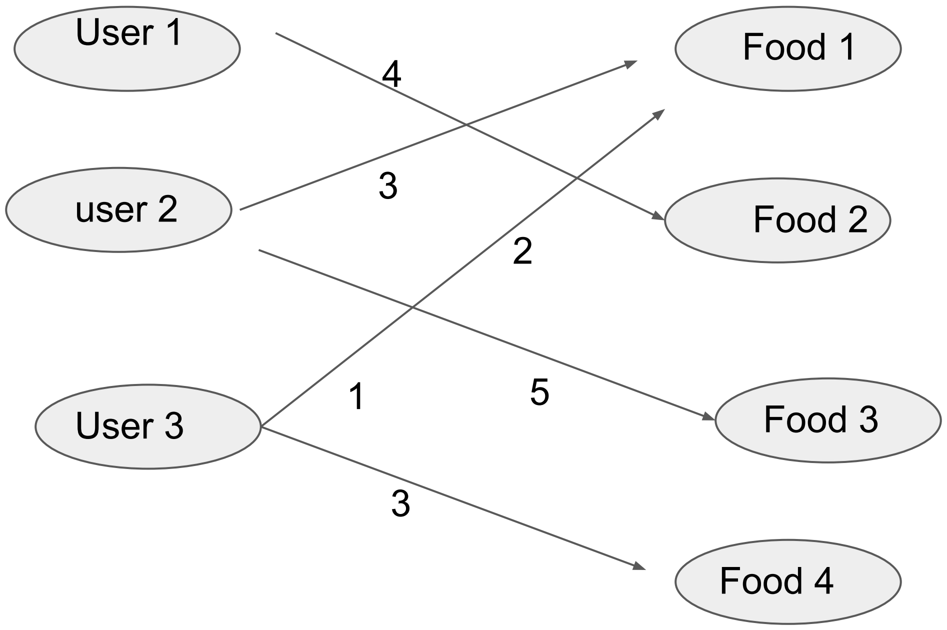
Encoded using a pretrained nlp model.
based on this paper : https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
It maps sentences & paragraphs to a 384 dimensional dense vector space and can be used for tasks like clustering or semantic search.

Eg: Sentence: I like apples

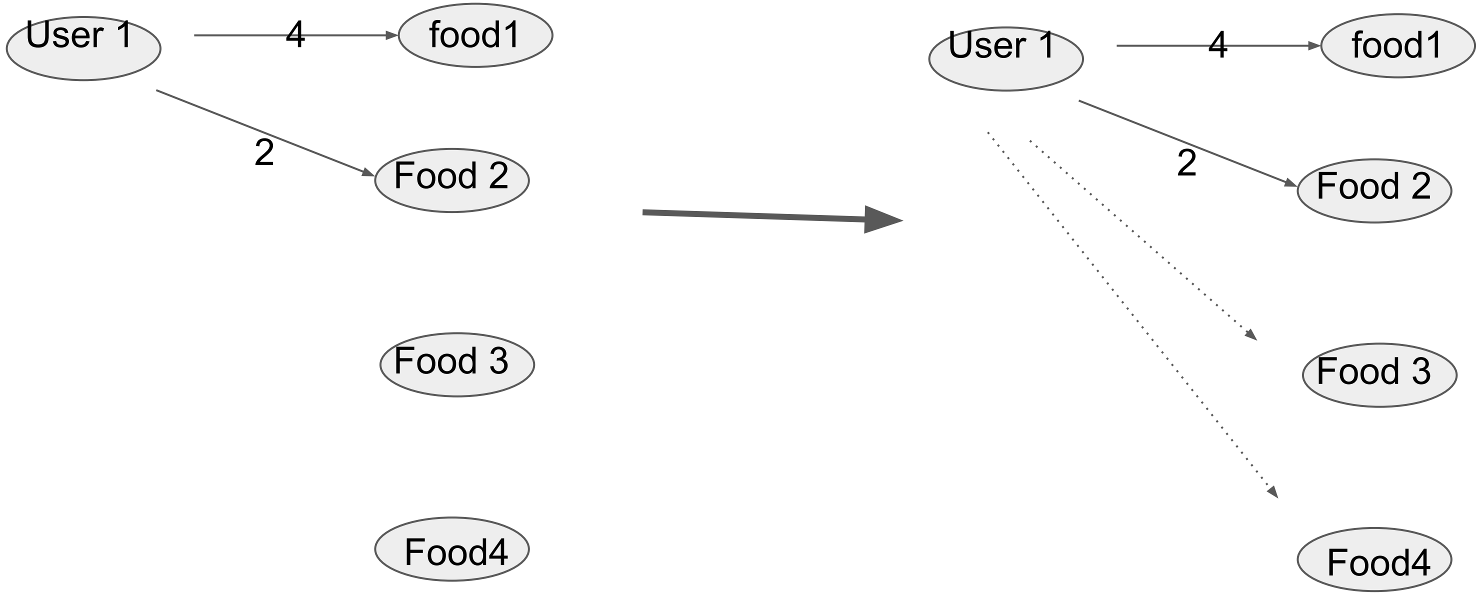
```
Embedding: [-2.88904291e-02  4.64452524e-03 -5.30242780e-03  3.52192707e-02
 -4.06657718e-02 -1.29475649e-02  1.21806540e-01  3.15719889e-03
  4.40227501e-02  3.67142558e-02  3.55952606e-02 -6.54208362e-02
  2.92756706e-02 -1.27371959e-02  3.32819670e-02  6.80304365e-03
  6.32420778e-02 -2.34737415e-02 -7.45559409e-02 -1.54757127e-02
 …..
…..
…..
…..
 -1.88434068e-02  8.92375188e-04 -1.24902343e-02 -7.60383159e-02
 -4.73403074e-02 -4.92182467e-03 -7.11007882e-03 -7.55073801e-02
 -9.50115472e-02 -2.16907579e-02  5.38594984e-02 -1.19841062e-02
 -2.19193678e-02  8.83059353e-02  4.25586626e-02 -2.83873733e-03
  9.04467851e-02  9.53318775e-02  4.97960746e-02 -1.24862483e-02
  3.18680480e-02 -6.74245059e-02  1.08114136e-02  7.36696050e-02
  1.14496395e-01  5.54728061e-02  1.08157434e-01  1.23441033e-02]
```

384 float numbers to represent any sentence

Input to the gnn:



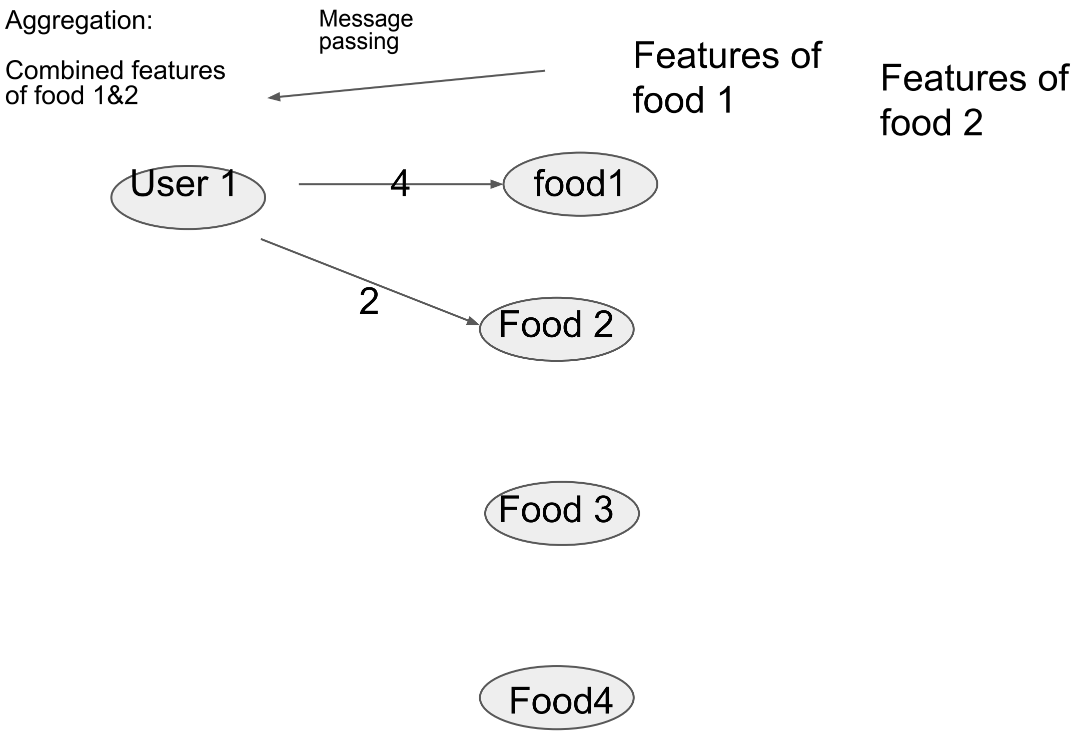
Link prediction:



Initial graph

New links and
their ratings

1. Fetch incoming messages from all neighbors
2. Reduce all those messages into 1 message by doing mean aggregation
3. Matrix multiplication of the neighborhood message with a learnable weight matrix
4. Matrix multiplication of the initial node message with a learnable weight matrix
5. Sum up the results of step 3 and 4
6. Pass the sum through a relu activation function
7. Repeat for as many layers as wished. The result is the output of the last layer.


Aggregation:
Combined features of food 1&2
Message passing
Features of food 1
Features of food 2
User 1
4
food1
2
Food 2
Food 3
Food4

Mathematically, the process can be defined as the following:

$$\mathbf{h}_u^{(k)} = \mathrm{relu}\left(\mathbf{W}_{\mathrm{self}}^{(k)} \mathbf{h}_u^{(k-1)} + \mathbf{W}_{\mathrm{neigh}}^{(k)} \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{(k-1)}\right)$$

$$\mathbf{z}_u = \mathbf{h}_u^{(K)}$$

where $\mathbf{h}_u^{(k)}$ is the embedding of node u at GNN layer k, all neighbors of u are nodes v in the neighborhood $\mathcal{N}(u)$, and $\mathbf{W}$ are trainable parameter matrices.

Designing the model: scoring preferences

The generated embeddings are used to predict the probability that a connection between two nodes exists. The predicted probability of interaction between a user u and an item v is given by the following equation, where f(·) is a cosine similarity function:

$$\hat{y}_{u,v} = f\left(z_u, z_v\right)$$

Cosine similarity **measures the similarity between two vectors of an inner product space**. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.

summary:

In simple terms, we propose an encoder-decoder, or representational learning approach. It can be divided in two steps.

- Generate high quality embeddings for all users and items
- For all users, predict item preferences using the embeddings

Generating embeddings is done through information propagation, also called neural message passing. Predicting preferences is done through simple cosine similarity. Using this approach, we manage to reach promising results and can propose further adjustments to enhance the model.