By *Jure Leskovec*

STANFORD UNIVERSITY

SNAP for C++
SNAP for Python
SNAP Datasets
What's new
People
Papers
Citing SNAP
Links
About
Contact us

Open positions

We have filled all the positions for this quarter. **More info**.

Web data: Reddit Pizza Requests

Dataset information

This dataset contains a collection of 5671 textual requests for pizza from the Reddit community "Random Acts of Pizza" together with their outcome (successful/unsuccessful) and meta-data. All requests ask for the same thing: a free pizza. The outcome of each request --- whether its author received a pizza or not --- is known. Meta-data includes information such as: time of the request, activity of the requester, community-age of the requester, etc.

Dataset statistics	
Number of requests	5,671
Timespan	December 8, 2010 - September 29, 2013
Average success rate	24.6%

Source (citation)

- Tim Althoff, Cristian Danescu-Niculescu-Mizil, Dan Jurafsky. How to Ask for a Favor: A Case Study on the Success of Altruistic Requests. ICWSM, 2014.

```
@inproceedings{althoff2014howtoaskforafavor,
  title={{How to Ask for a Favor: A Case Study on the Success of Altruistic Requests}},
  author={Althoff, Tim and Danescu-Niculescu-Mizil, Cristian and Jurafsky, Dan},
  booktitle={Proceedings of the International Conference on Weblogs and Social Media (ICWSM)
  year={2014}
}
```

Files

The full dataset archive can be downloaded here.

File	Description
pizza_request_dataset.json	The dataset in json format.
read_dataset.py	A simple python script illustrating how to read in the dataset as well as the attributes available for each single request.
narratives (folder)	A folder containing the lexicons used to detect the five narratives described in our paper (desire, family, job, money, student).
altruistic_requests_icwsm.pdf	The article introducing this dataset.
README.txt	A README file.

Data format

Each entry in pizza_request_dataset.json corresponds to one request (the first and only request by the requester on RAOP). For each request, the following fields are included:

Field	Description
`giver_username_if_known`	Reddit username of giver if known, i.e. the person satisfying the request ("N/A" otherwise).
`in_test_set`	Boolean indicating whether this request was part of our test set.
`number_of_downvotes_of_request_at_retrieval`	Number of downvotes at the time the request was collected.
`number_of_upvotes_of_request_at_retrieval`	Number of upvotes at the time the request was collected.
`post_was_edited`	Boolean indicating whether this post was edited (from Reddit).
`request_id`	Identifier of the post on Reddit, e.g. "t3_w5491".
`request_number_of_comments_at_retrieval`	Number of comments for the request at time of retrieval.
`request_text`	Full text of the request.
`request_text_edit_aware`	Edit aware version of "request_text". We use a set of rules to strip edited comments

	indicating the success of the request such as "EDIT: Thanks /u/foo, the pizza was delicous".
`request_title`	Title of the request.
`requester_account_age_in_days_at_request`	Account age of requester in days at time of request.
`requester_account_age_in_days_at_retrieval`	Account age of requester in days at time of retrieval.
`requester_days_since_first_post_on_raop_at_request`	Number of days between requesters first post on RAOP and this request (zero if requester has never posted before on RAOP).
`requester_days_since_first_post_on_raop_at_retrieval`	Number of days between requesters first post on RAOP and time of retrieval.
`requester_number_of_comments_at_request`	Total number of comments on Reddit by requester at time of request.
`requester_number_of_comments_at_retrieval`	Total number of comments on Reddit by requester at time of retrieval.
`requester_number_of_comments_in_raop_at_request`	Total number of comments in RAOP by requester at time of request.
`requester_number_of_comments_in_raop_at_retrieval`	Total number of comments in RAOP by requester at time of retrieval.
`requester_number_of_posts_at_request`	Total number of posts on Reddit by requester at time of request.
`requester_number_of_posts_at_retrieval`	Total number of posts on Reddit by requester at time of retrieval.
`requester_number_of_posts_on_raop_at_request`	Total number of posts in RAOP by requester at time of request.
`requester_number_of_posts_on_raop_at_retrieval`	Total number of posts in RAOP by requester at time of retrieval.
`requester_number_of_subreddits_at_request`	The number of subreddits in which the author had already posted in at the time of request.
`requester_received_pizza`	Boolean indicating the success of the request, i.e., whether the requester received pizza.
`requester_subreddits_at_request`	The list of subreddits in which the author had already posted in at the time of request.
`requester_upvotes_minus_downvotes_at_request`	Difference of total upvotes and total downvotes of requester at time of request.
`requester_upvotes_minus_downvotes_at_retrieval`	Difference of total upvotes and total downvotes of requester at time of retrieval.
`requester_upvotes_plus_downvotes_at_request`	Sum of total upvotes and total downvotes of requester at time of request.
`requester_upvotes_plus_downvotes_at_retrieval`	Sum of total upvotes and total downvotes of requester at time of retrieval.
`requester_user_flair`	Users on RAOP receive badges (Reddit calls them flairs) which is a small picture next to their username. In our data set the user flair is either None (neither given nor received pizza, N=4282), "shroom" (received pizza, but not given, N=1306), or "PIF" (given after received, N=83).
`requester_username`	Reddit username of requester.
`unix_timestamp_of_request`	Unix timestamp of request (supposedly in timezone of user but in most cases equal to the UTC timestamp which is incorrect since most RAOP users are from the USA).
`unix_timestamp_of_request_utc`	Unit timestamp of request in UTC.