DATA MINING

PROJECT REPORT[1]

Recommendation System for Reddit

Donald Chesworth, Yuning Ling, Zhuoyang Zhou, Sharath Pingula

Introduction

Reddit is a social media site where registered users can post a link, a comment or pictures and other members can down-vote or up-vote to determine the rank of the post. The content is organized according to *subreddits*, which are categories or areas of interest.

Problem Description

Reddit has more than 35 million registered users regularly engaged with 700,000+ subreddits. A registered user chooses/creates subreddits according to their areas of interest. Given the number of subreddits to choose from it is difficult for a new user to find a subreddit that is appropriate to their taste. The names of subreddits are typically a acronyms such as AMAA (Ask Me Absolutely Anything) or TIL (Today I Learned), most of the time not intuitive to most users. Even existing users may be interested in certain subreddits, but they were not aware that they existed.

For any social media site, a major objective is to continually engage their users. Regular user participation drives Reddit's generation of data, advertising demand, and their overall business model. Site accessibility and the availability of relevant comments is essential to make users read, post, reply, or comment.

We believe that a recommendation system based on the nature of discussions would help Reddit guide their users towards the subreddits of their interest.

[1] Update to the original proposal, *Data Mining Project Proposal: Recommendation System for Reddit*, 23 Nov. 2015

Over the past few years, the number of subreddits has grown exponentially, reaching about 670,000 as of July 8, 2015 (as shown in the figure from Statista below).[2] Since its inception, Reddit has never used a recommendation engine to make targeted suggestions, but only provided universal suggestions such as new, popular, or categorical subreddits. As it is essential for Reddit to maintain or extend the average user engagement to thrive in the exceedingly competitive social media industry, it may be important for reddit to adopt a recommendation engine to make users aware of appropriate subreddits to keep them constantly engaged. Although in the past users have had complete choice over subreddits selection, due to this phenomenal growth in subreddits it may become too cumbersome for users to discover subreddits of interest.

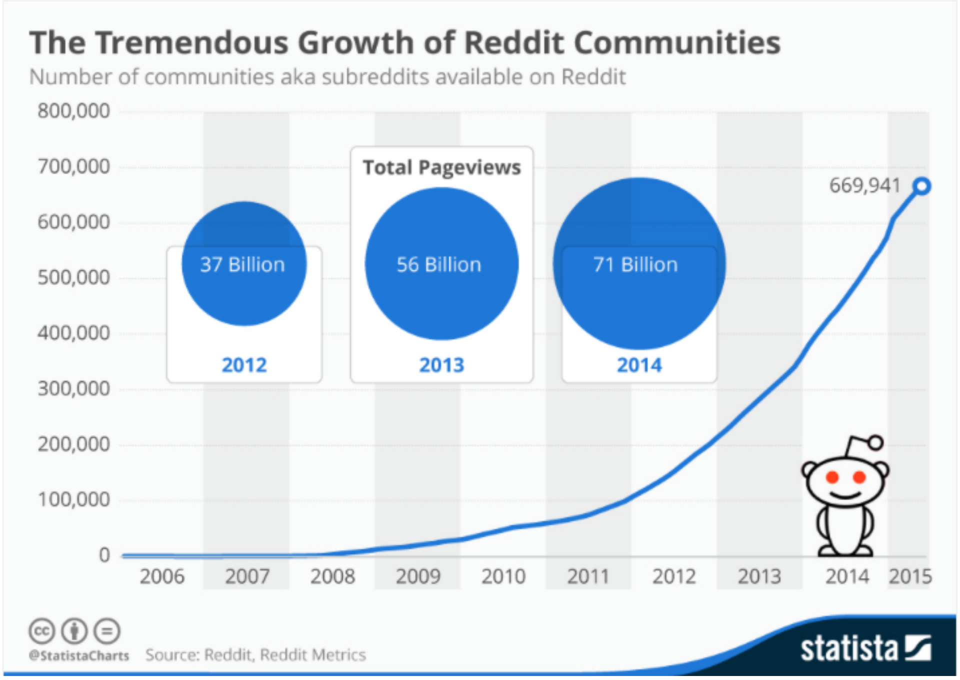


Figure 1: Statista Graph of Subreddit Growth

[2] Richter, Felix. *The Tremendous Growth of Reddit Communities.* Statista, 8 July 2015. Web. http://www.statista.com/chart/3623/reddit-growth/ 22 Nov. 2015.

Objectives

We are proposing to build a recommender system that can suggest specific subreddits to specific Reddit users. For a given user based on their subreddits and the nature of discussions they are engaged in, we would like to identify algorithmically additional subreddits the user would most likely want to be a part of. Our hypothesis is that targeted suggestions would improve the level of engagement of users on Reddit.

The idea behind a recommendation system is to link user space and subreddit space using topics and recommend a list of top subreddits for a user based on his/her top topics or his/her neighbor's topics. Neighbors will be defined as users who have similar areas of interest and are part of discussions of similar nature.

If neighborhood space is used for recommendation we propose to create it by building topic models on each user and clustering users who are closer with respect to their topic vectors. Once a user neighborhood is created, the list of topics of the nearest neighbors will be used to get the top topics. Then, the closest subreddits to those top topics will be discovered using a cosine similarity index. The subreddit list will be returned as recommended subreddits for the user.

Evaluation

To evaluate our system we originally proposed to use following metrics that are widely used in evaluation of recommendation systems, similar to those used by Sawar, Karypis, Konstan, and Riedal.[3]

[3] Sarwar, Badrul, et al. *Application of dimensionality reduction in recommender system-a case study*. No. TR-00-043. Minnesota Univ Minneapolis Dept of Computer Science, 2000.

$$Recall = \frac{|\,Common\ subreddits\ in\ both\ the\ Test\ Set\ and\ the\ Top\ 10\,|}{|\,Test\ Set\,|}$$

$$Precision = \frac{|\,Common\ subreddits\ in\ both\ the\ Test\ Set\ and\ the\ Top\ 10\,|}{10}$$

$$F1\ Measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Figure 2: Common Evaluation Metrics

Although we originally proposed these metrics, we have determined that the above metrics will not work in our case since we are recommending based on nature of the posts posted by users. If a user has posted about Australia in 'world politics' subreddit and our system recommends subreddit 'australia' to him, it should be considered as a valid recommendation but this particular scenario will not considered valid when above metrics employed.

Instead, we evaluated manually using subreddit heuristics where a valid recommendation is determined based on existing subreddits of the user.

Related Work

There has been a lot of work done on recommender systems, especially for products and services on e-commerce platforms. These are becoming necessary tools for helping users find relevant products and services on web. Our idea is inspired by the famous Netflix Prize.[4] We would like to apply the Singular Value Decomposition (SVD) algorithm in creating user neighborhoods using topics as our explanatory variables. Our approach of creating neighborhoods is developed based on the work of Sarwar et. al.[5]

[4] Netflix. *The Netflix Prize.* Netflix. 2 Oct. 2006, 22 Nov. 2015.

[5] Sarwar, Badrul, et al.

In our research, we came across similar studies by Aakash Japi[6] and Manuel Garrido.[7] Japi's approach was to find a user's nearest neighbors (70 users) based on like subreddits, then suggest the most common subreddit that the user had not yet subscribed to. Garrido's approach was to determine a universal list of "similarity" between subreddits based on how many users wrote comments in like subreddits. Then, Garrido suggested subreddits to users based on those similar to the user's subscribed-to subreddits.

We have not come across any work on recommender systems that could suggest topics or discussion groups that match with areas of interest of an user (although a commenter on Japi's blog suggested our topic modeling approach!). Our approach instead builds upon the engines developed for recommending products and services to e-commerce customers.

We consider our study to be novel with respect to finding the most likely discussion groups for users using topic modeling and other text analytics algorithms. if proved effective our solution can be applied in variety of applications such as Quora, Linkedin, Yelp or any other text based social media site that would want to increase its user engagement.

Approach

Data

Kaggle has produced an excerpt of all Reddit comments from May 2015 in the form a SQLite database.[8] The fields that contain the most relevant data are **`author`**, **`subreddit_ID`**, **`subreddit`** and **`body`**. The original size of the excerpted reddit database is 33 GB (54.5 million posts). In order to use one million posts, and to have enough posts per user, we determined that using all authors with 500-575 posts would provide 1,866 authors

[6] Japi, Aakash. "Re: Best Strategy: Fenced Pastures vs. Max Number of Rooms?" *The Sopranos, Silicon Valley, and Summer Afternoons*. 17 July 2015. Web. 22 Nov. 2015.

[7] Garrido, Manuel. "Building a Recommendation Engine for Reddit" *Manuel Garrido's Blog.* 12 Nov. 2014. Web. 22 Nov. 2015.

[8] Kaggle. "Reddit Comments: Get personal with a dataset of comments from May 2015" *Comments" Kaggle.* 4 Aug 2015. Web, kaggle.com/c/reddit-comments-may-2015/data. 22 Nov. 2015.

and 999,653 records. Although this method proved it would be successful, the first topic modeling run took 14.3 hours. Because of this, we decreased the number of records by a factor of ten so that we could run our simulation multiple times in order to create a proof of concept.

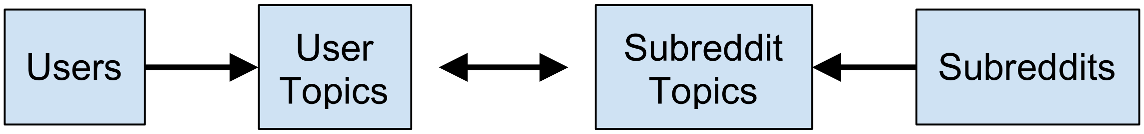
Assumptions

As on typical bulletin board sites, on Reddit we find more consumers of content than contributors. Since our system is based on analyzing the posts generated by users, this approach would not be effective for users who are only content consumers. However, this same approach could be leveraged if Reddit provided data on the comments that were *read* or *consumed* by users. Given the availability of data, we have limited our scope to generating recommendations for users who are contributors.

Methodology

The recommendation of subreddits was performed based on the overall themes of posts a user has posted. A user's theme is understood through generating topics using a Latent Dirichlet allocation (LDA) algorithm. We then used these topics to identify subreddits where similar topics were being discussed. Our hypothesis is that matching user topics to subreddit topics will provide a more accurate representation of a user's interests than would collaborative filtering.

In order to associate a user's topic to subreddits where similar topics are being discussed, we built topics on subreddits and matched the user topics and subreddit topics based on their similarity. Once matches were established between users and subreddits, we proposed two recommendation types for a given user:



- their two best matched topics
- the best matched topics of their neighbors

We found that recommending the user’s own best matching topics yielded better results in comparison to recommending neighbors’ topics.

In summary, topics are the bridge between the user space and the subreddit space.

Step-by-step Execution

The following actions were performed in order to recommend subreddits to users:

1. Established a database connection with a 54.5 million post SQLite database for querying and exporting data.
2. Extracted one million records using a multi-step query that extracted all posts by authors who have posted between 500 and 575 posts (high frequency authors will likely produce better results).
3. Since step 5 below (topic modeling) took more than 14 hours for one single run on one million records, we limited our posts to a random sample of 100,000 high-frequency author posts, and removed subreddits with less than 100 posts.
4. Aggregated posts for each user (using the **`author`** field) such that each user document contained all posts written by one user, and created in a document-term matrix (DTM) with term frequency (TF) rating.
5. Built topics on user DTMs using a Latent Dirichlet Allocation (LDA) algorithm, thereby representing each user as a vector of topics and each topic as a vector of words. Specifically:
   a. Our sample data contained 1,791 users with an average of 50 posts each
   b. Our LDA algorithm created 50 topics
6. Aggregated posts within each subreddit user (using the **`subreddit`** field) and built topics on subreddits similar to steps 4 and 5. In this case, each subreddit was represented as a vector of topics. Specifically:

    a. Our sample data contained 200 subreddits
    b. Our LDA algorithm created 50 topics
7. Binarized the user topic vectors and subreddit topic vectors by setting approximately the top 300 scoring topics to 1 and the rest to 0. Then, topics were matched based on the dot product between the vectors topic word vectors of corresponding topics. Similar topics have the same words with high probabilities in their topic word vector.
8. Calculated Recommendation Method 1: For a given user, we recommended all the subreddits that matched with his/her topics excluding the ones he/she has already registered with.
9. Calculated Recommendation Method 2: For a given user, we recommended all the subreddits that matched with his/her neighbor’s topics excluding the ones he/she was already registered with. Specifically, we:
    a. Applied Singular Value Decomposition (SVD) on the User-Topics matrix, creating a reduction in the matrix to get a user space and a topic space.
    b. Derived the top topics of K (K=3) closest neighbors in the user space for each user. The neighborhood vicinity factor (K=3) was selected by trying values from 1 to 10.

Results

We have created a list of matching topics from the user space for all the subreddits as described above. As shown in the following table, the subreddit “2007scape” can be best represented by Topic 21 in the subreddit space, and Topic 32 and Topic 20 in the user space. It can be intuitively understood that when a user’s posts contain Topic 32, then their posts are similar to the content discussed in subreddit “2007scape.” Similarity between the topics was calculated using an adjusted cosine similarity metric on topic word vectors of corresponding topics (basically the dot product of the two vectors).

	Subreddit Name	Best Matched Subreddit space Topic	First Best Matched User space Topic	Second Best Matched User space Topic
1	2007scape	21	32	20
2	4chan	38	30	5
3	ACTrade	5	31	41
4	Advice	34	2	45
5	AdviceAnimals	50	18	23
6	AFL	7	6	40
7	AgainstGamerGate	13	46	21
8	amiibo	43	7	17
9	Anarcho_Capitalism	45	33	48
10	Android	11	40	16

Figure 3: Subreddit Matching

The following tables show a four subreddits, along with the Top 10 Words from their author topic and their subreddit topic.

2007scape	
SR Topic 21	AU Topic 32
damag	damag
build	hit
weapon	met
pvp	run
star	pvp
cig	build
ship	star
end	hard
player	player
hard	might

Fitness	
SR Topic 34	AU Topic 2
fat	fat
weight	eat
god	weight
eat	bodi
bodi	week
believ	women
muscl	food
calori	live
problem	kid
wrong	long

4chan	
SR Topic 38	AU Topic 30
charact	win
kill	charact
smash	deck
player	player
shield	card
geralt	littl
hit	minion
attack	shield
run	power
level	kill

Buildapc	
SR Topic 20	AU Topic 25
card	card
price	price
case	case
power	power
cooler	cooler
cpu	motherboard
motherboard	atx
atx	cpu
rebat	rebat
drive	suppli

Figure 4: Four Subreddits with their Subreddit and Author Topics

Each subreddit topic can have one to many relationships with subreddits. For example, topic 45 in subreddit space is representing 4 different subreddits: “Anarcho_Capitalism”, “Bitcoin”, “GlobalOffensiveTrade” and “Games”.

Subreddit Topic 45
bitcoin
system
bit
buy
problem
uchangetip
exist
valu
money
sell

Figure 5: Top 10 Topics for Topic 45

Recommendation Approach 1

In our first approach of recommendation, we have proposed all the matching subreddits of a user’s top topics. The following table shows the recommendations generated by our algorithm using the user topics, which are appropriately matching with user interests. We have determined the match heuristically from the registered subreddits of users. The subreddits that are relevant to each other in both registered and recommended are in bold face. The relevancy is determined manually based on subreddit description.

Recommendations using Method 1 - User Topics		
User Name	**Registered Subreddits**	**Recommendations by the system**
knlmustard	**soccer | nba | Gunners** |	**nba | baseball | clevelandcavs | NewYorkMets | Mariners | Dodgers | CHICubs | KCRoyals | TexasRangers | warriors |**
Meghdoot	**india** | **worldnews** | asoiaf | **Cricket** | gameofthrones |	**news** | **worldnews** | politics | PoliticalDiscussion | **Cricket** | **worldpowers** | **syriancivilwar** | Libertarian | SandersForPresident | ProtectAndServe |

Recommendations using Method 1 - User Topics		
User Name	**Registered Subreddits**	**Recommendations by the system**
Tuberomix	**Android** | IAmA | todayilearned | **gaming | thebutton** | worldnews | **Games** | atheism | **pcmasterrace** | pics | science | AskReddit |	**pcmasterrace** | gameofthrones | Smite | DebateReligion | **TwoXChromosomes | PS4 | pcgaming | AgainstGamerGate** |
rkwittem	**nfl | hockey | baseball | AskMen | AskReddit | CFB | StarWars** |	**nfl | hockey | guns** | xboxone | IAmA | hawks | Warframe | **motorcycles | Patriots** |
Oomeegoolies	**leagueoflegends | witcher | soccer** | pics | Cricket | AdviceAnimals | **gaming** | worldnews | **pcmasterrace** |	**leagueoflegends | DotA2 | survivor | FIFA | apple |**

Recommendation Approach 2

In our second approach, we have generated recommendations using the topics of neighbors instead of using the user's own topics. We have observed that recommendations in this case are not as effective as first approach.

Recommendations using Method 2 - Neighbors' Topics		
User Name	**Registered Subreddits**	**Recommendations by the system**
knlmustard	**soccer** | **nba** | **Gunners** |	**nba** | SquaredCircle | movies | **baseball** | OkCupid | **clevelandcavs** | television | **NewYorkMets** | **Mariners | Dodgers | CHICubs | KCRoyals | TexasRangers | warriors** |
Meghdoot	india | worldnews | asoiaf | Cricket | gameofthrones |	|
Tuberomix	**Android** | IAmA | todayilearned | **gaming** | thebutton | worldnews | Games | atheism | **pcmasterrace** | pics | science | AskReddit |	**pcmasterrace** | gameofthrones | Smite | DebateReligion | TwoXChromosomes **| PS4** | pcgaming | AgainstGamerGate |
rkwittem	**nfl | hockey | baseball |** AskMen | **AskReddit** | CFB | StarWars |	**AskReddit** | **nfl | hockey** | hawks | **Patriots** |
Oomeegoolies	leagueoflegends | witcher | soccer | pics | Cricket | AdviceAnimals | gaming | worldnews | pcmasterrace |	DestinyTheGame | CasualConversation | asoiaf | YamakuHighSchool | marvelstudios | Wishlist | OnePiece | Marvel | StarWars | shield |

Evaluation

Subreddit recommendations proposed by our system for a target user will be evaluated against relevance of recommended subreddits with respect to actual subreddits the user has already registered with.

In the following table, all the recommended subreddits could be considered relevant since all are related to specific sports and relevant to the registered subreddits of the user.

4	Registered Subreddits	Recommendations by the system
knlmustard	**soccer | nba | Gunners** |	**nba | baseball | clevelandcavs | NewYorkMets | Mariners | Dodgers | CHICubs | KCRoyals | TexasRangers | warriors |**

This scenario cannot be captured by the regular evaluation metrics used in recommendation systems such as recall, precision and F1-measure.

We have manually given a relevance score of 1 if the recommended subreddits are relevant to the user's registered subreddits. Otherwise, we gave a relevance score of 0. We have generated recommendations for 100 random users using our two recommendation methods. We have generated relevant recommendations in 75 users using method 1 and 24 users in method 2. Therefore, approach 1 was relevant 75% of the time, and approach 2 was relevant 24% of the time.

Challenges

Our main challenge was the execution time to build topics on posts. We attempted topic modeling with one million posts and 300 topics, but terminated the process after 24 hours, while one million posts and 50 topics took 14.3 hours. Since we have to build two topic models for each execution (one on the user space and the other on the subreddit space), we limited our dataset to 100,000 posts and 50 topics. This is not only a challenge for our project, but will also be a challenge for implementing this project within Reddit.

We tried a novel approach of finding similar topics on two different runs of topic modeling algorithm on the same corpus. We achieved it finally through binarizing the topic word vectors and finding similarity between topics using a modified cosine similarity metric (essentially the dot product of the matrices). It took an exorbitant amount of time to determine a solution for determining similarity between two topics generated in two different runs of the algorithm.

Due to recommendations via topic modeling being a new approach, there were no standard evaluation metrics available. We have manually evaluated the recommendations using subreddit relevance as a metric. It would be a future scope to develop a standard metric for studies like these. In addition, manually evaluating matches is not a maintainable approach if our project were applied at Reddit.

Finally, our algorithm is based on topics generated on posts contributed by users, so it cannot directly be applicable for users who are content consumers. This can be overcome by considering the content consumed by those users as their posts.

Applications

Our algorithm can be extensively used in all bulletin-board-type sites such as Reddit, Voat, Quora, StumbleUpon, NewsVine or 4Chan. It is extremely helpful to direct users towards pages that match with their interests, thereby increasing their activity and engagement. Our proposed method can also be used in clustering subreddits/categories for better targeting of advertisements.

In addition, our algorithm could be modified for applicability to instant recommendations. Topic modeling with a matching list of topics between the user space and the subreddit space can be run as a batch process. The final matching list with subreddits and corresponding best matched user topics can be used for generating recommendations for any target user based on their activity, which could be either contributing content (posting) or consuming content (reading).

Methodology for Instant Recommendations

In order to create instant recommendations, we would:

- Create a user document based on activity (could be user posts or read content)
- Convert the document into a vector of words using term frequency as weighting
- Match this term word vector with topic word vectors using the adjusted cosine similarity proposed in our algorithm
- Generate recommendations instantaneously based on the subreddits associated with the best matching topics.

Future Scope

1. Extrapolation: Our recommendation system was designed using high frequency users, yet it is likely that new users would benefit most from recommendations, yet (1) our model is time-intensive, and (2) running topic modeling for each user is not possible when the user has little to no posts. Our model could be used to provide the topic to subreddit connection, and instead of topic modeling for the new user, we could use the TF rating of a user's posts, or the TF rating of the posts a new user reads and relate them to topics.
2. Time efficiency: The execution time for topic modeling for about 1,800 users with 500-575 posts each (one million posts total) was 14.3 hours. In the future, we would like to run topic modeling in parallel, using an approach known as parallel latent dirichlet allocation (PLDA).
3. Lack of Metric: Unlike predictions involving numeric values and other regular recommendation systems, we didn't find any existing metric which is appropriate in our case. For this project, as mentioned earlier, we manually gave relevance score

for the predictions. However, for future work, it will be more efficient if we can formulate a better evaluation metric for the text-based recommendation system.

References

Garrido, Manuel. “Building a Recommendation Engine for Reddit” Manuel Garrido’s Blog. 12 Nov. 2014. Web. 22 Nov. 2015.

Japi, Aakash. “Re: Best Strategy: Fenced Pastures vs. Max Number of Rooms?” The Sopranos, Silicon Valley, and Summer Afternoons. 17 July 2015. Web. 22 Nov. 2015.

Kaggle. “Reddit Comments: Get personal with a dataset of comments from May 2015” Comments” Kaggle. 4 Aug 2015. Web, kaggle.com/c/reddit-comments-may-2015/data. 22 Nov. 2015.

Netflix. The Netflix Prize. Netflix. 2 Oct. 2006, 22 Nov. 2015.

Richter, Felix. The Tremendous Growth of Reddit Communities. Statista, 8 July 2015. Web. http://www.statista.com/chart/3623/reddit-growth/ 22 Nov. 2015.

Sarwar, Badrul, et al. Application of dimensionality reduction in recommender system-a case study. No. TR-00-043. Minnesota Univ Minneapolis Dept of Computer Science, 2000.