Determining Community Interest in Text-Only Posts on Reddit

Project Proposal

Charles Lewis	Nathaniel Price	David Purser
University of Michigan	University of Michigan	University of Michigan
Ann Arbor, MI	Ann Arbor, MI	Ann Arbor, MI
noodle@umich.edu	nrprice@umich.edu	dpurser@umich.edu

ABSTRACT

This paper proposes a term project for the EECS 498 Information Retrieval course at the University of Michigan which seeks to analyze content on Reddit[1] to determine community interest in certain topics, keywords, and questions.

The project involves collecting a set of data from a number of subreddits (individual forums on Reddit) including textual content of each post, number of votes that the post received, number of comments on the post, and other information. This information will be used to estimate the community's interest in each post. These estimates will be compiled and indexed by keyword, allowing a user to query the system to determine the expected interest in a new post and whether a new post is very similar to previous posts.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Data analysis, Reddit, community interest

1. PROBLEM

We sought to explore whether it was possible to predict whether a post would be popular (community interest) or not on `www.reddit.com`, specifically text-only posts on subreddits sucht as `/r/AskReddit` or `/r/ELI5`. `Reddit` is a platform where users upvote and downvote posts based on various criteria. The users also have the option of posting a comment on a particular post. The overall motivation was to research different machine learning and information retrieval process in order to figure out which would prove to be helpful in determining community interest on `Reddit`.

---

[1]`http://www.reddit.com/`

There are various applicaitons in which this work would benefit. This could help indicate when it might be a useful time to repost something. The community of Reddit has a strong aversion to things that are posted often. These are often refered to "reposts". However, there is some benefit to reposts. Users that are recently new will find the content new and relevant to them. In addition, there is benefits to bringing up a topic again as it will generate close to the same community interest as it did when it was initially posted. Our work could provide a way of detecting when might be the best possible time to repost something for it to be meaningful and relevant to the community.

An alternative application could be for buisness marketting. Knowing how a particular community will respond to your post can help buisness tailor their marketing strategies to a particular subreddit. There is also the possibility that future work on some of the proposed solutions/methods that we researched could help indicate things that a marketer could change in order to make their content generate more interest.

2. PREVIOUS WORK

<NEED TO ADD TO THIS>

There are a number of websites dedicated to detecting "reposts" on Reddit, which are exact duplicates of previously posted content. One such website is KarmaDecay[2]. However, these websites only detect exact duplicates of hyperlinks and images, which make up the majority of posts in many subreddits. The textual content of posts in text-only subreddits such as `/r/AskReddit` or `/r/ELI5` is not analyzed. This gives these websites limited usefulness in text-only subreddits, where no images or external links are generally allowed.

Furthermore, existing systems are designed mostly to help users avoid reposting exact duplicates of existing material, and do not match based on a similarity measure. This means that related or very similar posts will not be matched, only exactly identical ones. By using a text-matching system based on similarity scores, our project will allow text-only topics to be analyzed not to detect exact duplicates (as these are rare in text posts due to the diversity inherently present in language), but to find similar posts from the past based on matching keywords, topics, phrases, and other content. These similar posts can then be used to judge or estimate

---

[2]`http://karmadecay.com/`

potential interest in the new post and to predict useful information such as whether a post will be popular or successful or which subreddits it may be most successful in by analyzing the interest in the previous posts. In addition, the uniqness of the post will also contribute to determing how popular a post might be.

Therre has been work done analyzing news content, not based on the article but upon the responses of the general public [?]. This paper describes a way of what they describe as "comment centric tagging" as a means to identify which articles were highly interesting and important. They use these comments within a post to alter and adapt the interest weighting on an article.

While this work is not the main focus of the paper, it does highlight how important the comments of a post can be. The study they performed showed that the comments of a post greatly impacted the community interest within that specific topic. This will help us direct or findings towards the comments found on Reddit, weighing them higher, than the actual post itself.

Some work has also been done by researchers to determine characteristics of articles that have been ranked by crowdsourced feedback from users [?]. The results of this study point out that while crowdsourced feedback mechanisms "can be relatively effective for ... promoting content of high quality, they do not perform well for ordinal objectives such as finding the best articles." In other words, while interesting topics and general themes are often highlighted as outstanding by user feedback, the best specific articles are not always at the very top of the list. Rankings which are based on user feedback will likely tend to positively identify good topics, but perhaps not the absolute best topics. This means that we should not use a post's score as an absolute ranking; that is, we should not conclude that one post is definitively better than another simply because it has a higher score. For our purposes, this means that all posts with a reasonably high score could be determined to be interesting to the community, but that comparing levels of interest (or ranking the level of interest) between two interesting topics will likely be very difficult.

The article also has some interesting discussion about crowdsourced scores over time, indicating that posts tend to take a while to gain momentum, but then explode rapidly if they are determined to be interesting. This means that some posts which are not "discovered" by users (they may be posted at the wrong time of day or when few users are online, so they are not identified by interested users) may fail to gain traction and perform as they should. This may mean that the same post, created at different times, may be extremely popular one time and not at all another time. In fact, the post may go unnoticed several times and only be discovered once. Our scoring algorithms will have to take this into account, likely by giving the positive feedback of posts which are determined to be interesting much more weight than the negative feedback of posts which are determined to be uninteresting (because a post that scores highly is almost certainly an indicator of community interest, while a post that scores poorly may have many different reasons for receiving that score).

Work has been done in this area before by another Michigan student, targeting Digg instead of Reddit. His website[3] analyzed all of the most recently-posted links on Digg, and used various heuristics to predict which articles would make it to the front page. Various differences between the algorithms used by the two sites make it impossible to use the exact same approach here–the code relied on being able to see which users voted on which articles, while on Reddit this information is made private by default. In addition, Digg weighted the votes of some users above others instead of treating everyone equally. The fundamental concept of extrapolating early votes to predict eventual popularity will still be an important concept for us to investigate.

In addition to using upvotes and downvotes to rank content, reddit keeps track of the total score of all of a user's comments. Power users of reddit have a strong incentive to identify posts that will eventually reach the front page, since commenting early on these threads provides the maximum return of points for the time spent. To this end, the subreddit /r/risingthreads tries to identify these threads within minutes of when they are posted, but how it does this is not known. The submissions to this subreddit are submitted by a bot, that presumably flags posts with lots of early upvotes. In the sidebar of the subreddit, /r/askreddit is specifically named as a subreddit for which the bot's algorithm performs poorly. Our project, by analyzing the text content of the post and its comments, may be able to improve upon this performance. This subreddit also shows a way to get user feedback on our project's effectiveness. If our project works well enough, we could set up our own subreddit to share and test its results.

3. APPROACH

Our approach for this project involves three major areas: data collection, processing, and retrieval.

First, data must be collected from Reddit. This will involve collecting and indexing a number of past posts in text-only subreddits. The posts and their comments will be analyzed, looking for common keywords, themes, or topics which seem relevant to the post in order to match the post with other submissions that have similar topics.

We focused on the "score" of a redit post (the number of upvotes minus the number of downvotes). When implementing our various different systems, we focused on the score as our measure of community interest. We thought of using the comments instead of the score, or inconjunction with the score, but we did not experiment and analyze that within our research and implementations.

There were three different implementations that we approached: cosine similarity between topics, naive bayseian on the words found within the posts, and a boolean classifier machine learning approach. We believed that each method had benefits and disadvantages and wanted to experiment with the various different methods to see which method, if any, peformed better than the baseline.

---

[3]http://gigaom.com/2010/08/24/hey-digg-this-17-year-old-knows-what-you-are-thinking/

<cosine approach>

<keyword approach>

<machine learning approach> Next, the popularity or interest of the post will then be measured through various statistics including number of comments, number of upvotes/downvotes received (through Reddit's voting system), number of distinct users who commented on the post, average length of the comments, average depth of comment trees[4], number of upvotes/downvotes on comments and their distribution among the comments, time of day that the post was made, and how long the post has been on Reddit.[5]

Finally, the keywords, and posts matching those keywords, will then be used to create a database which can be used to judge popularity of a post based on which keywords appear in the post title or text body. This database can then be queried to determine expected popularity of a new post given its text. This database will also be designed in such a way that new posts on Reddit will be automatically incoporated into the corpus. The query system may also be able to suggest similar keywords (words which are frequently used with the ones in the given text), topics, or information to include in the post which may increase the post's popularity score (and thus hopefully lead to a more interesting post on Reddit). [6]

4. EVALUATION
<Nathan will fill this in>

5. RESULTS
<keyword approach>

<cosine approach>

<machine learning approach>

6. CONCLUSION
<Nathan will fill this in>

---

[4]On Reddit, users can comment on other comments, forming a tree-like structure of comments.

[5]Reddit's algorithm works in such a way that posts that have been on Reddit a long period of time have a lower "score". We may have to inflate the score given to older posts as their ranking has decayed over time.

[6]This is a possible extension of the project and will need to be explored further.