Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow

Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, Jure Leskovec
Intro + Motivation

Q&A sites have evolved:
from places to get one-off answers to questions

to large repositories of long-lasting, valuable knowledge
Intro + Motivation

In this work, we promote a systemic view of Q&A sites. Rather than focus on question-answer pairs, we view a question together with its full set of answers. We show that this new approach can help solve important problems in modern Q&A sites:

- Early identification of pages with long-lasting value
- Finding questions with insufficient answers
Outline

1. Data
2. Introduce tasks
3. Empirical findings
4. Task performance
Outline

1. Data
2. Introduce tasks
3. Empirical findings
4. Task performance
Large, focused programming-related Q&A site

Very well curated by the community

<table>
<thead>
<tr>
<th>Users</th>
<th>440K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questions</td>
<td>1M</td>
</tr>
<tr>
<td>Answers</td>
<td>2.8M (26% marked as accepted)</td>
</tr>
<tr>
<td>Votes</td>
<td>7.6M (93% positive)</td>
</tr>
<tr>
<td>Favorites</td>
<td>775K (on 318K questions)</td>
</tr>
</tbody>
</table>

Complete dataset
How to format a JSON date?

I'm taking my first crack at Ajax with jQuery. I'm getting my data onto my page, but I'm having some trouble with the JSON data that is returned for Date data types. Basically, I'm getting a string back that looks like this:

```
/Date(1224043200000)/
```

From someone totally new to JSON - How do I format this to a short date format? Should this be handled somewhere in the jQuery code? I've tried the `jQuery.UI.datepicker` plugin using `$ .datepicker.formatDate()` without any success.

FYI: Here's the solution I came up with using a combination of the answers here:

This solution got my object from the callback method and displayed the dates on the page properly using the date format library.
Eval is not necessary. This will work fine:

```javascript
var date = new Date(parseInt(jsonDate.substr(6)));
```

The `substr` function takes out the "VDATe(" part, and the `parseInt` function gets the integer and ignores the ")V" at the end. The resulting number is passed into the `Date` constructor.

This worked for me, thanks! – Jhonny D. Cano -Leftware- Mar 16 '10 at 21:20

+1 for not using eval, and for also working if a timezone offset is included in the string (which was the case for me). – Remi Despres-Smyth Apr 13 '10 at 17:34

+1 This answer should be marked the answer. It is the most universal. – Evildonald Aug 23 '10 at 19:20

Here's a good explanation of why Microsoft chose this format to represent JSON dates in ASP.NET:

http://weblogs.asp.net/bleroy/archive/2008/01/18/dates-and-json.aspx

Good link thx Chris – aromawebdesign.com Mar 6 '11 at 22:00

+1 This link actually contained just the info I needed. Thanks! – Gundre Aug 8 '11 at 8:52

exploiting loopholes in json is evil. I don't like it. – neoneye Nov 10 '11 at 10:58

You can use this to get a date from json:

```javascript
var date = eval(jsonDate.replace(/\Date\[(\d+)\]/gi, "new Date($1)");
```

and then you can use JavaScript Date Format script (1.2 KB when minified and gzipped) to display it as you want.
Reputation

Stack Overflow is endowed with a highly respected reputation system

<table>
<thead>
<tr>
<th>Action</th>
<th>Reputation Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q/A is upvoted</td>
<td>+5/+10</td>
</tr>
<tr>
<td>Q/A is downvoted</td>
<td>-2 (-1 to voter)</td>
</tr>
<tr>
<td>Answer is accepted</td>
<td>+15 (+2 to acceptor)</td>
</tr>
<tr>
<td>Answer wins bounty</td>
<td>+ bounty amount</td>
</tr>
<tr>
<td>Offer bounty</td>
<td>- bounty amount</td>
</tr>
<tr>
<td>Answer marked as spam</td>
<td>-100</td>
</tr>
</tbody>
</table>
Outline

1. Data

2. Introduce tasks

3. Empirical findings

4. Task performance
Tasks

Two questions from the Q&A site owner’s perspective:

1. Predict long-term value of a question page
   help guide consumers of information to high-quality content

2. Predict whether a question has been sufficiently answered
   help direct producers of information to questions in need of expert attention

What features should we use to predict this?
Outline

1. Data
2. Introduce tasks
3. Empirical findings
4. Task performance
Is there a relationship between the site-level reputation system and question-level dynamics?

Higher-rep users arrive earlier
First principle: Reputation Pyramid

Mental model, not an explicit structure
The longer it takes for the first answer to arrive, the less likely that any answer will be accepted

Consistent with reputation pyramid picture!
Two competing notions of answer quality:

Earlier More rep points
Later Better vote score

Resolving these 2 notions is an open problem
Second Principle: “rising tide lifts all boats”

Is there competition between answers?

More activity → more votes for everybody

Supports our systemic view of Q pages
Outline

1. Data
2. Introduce tasks
3. Empirical findings
4. Task performance
Task 1: predict long-term value of a question page given how it looks a short time after it is created.

Long-term value = Number of page-views one year after creation (in our data).

See one hour of data, predict views one year later.

Set up as binary classification task: high/low page-views.

We optimize for simplicity and interpretability — use logistic regression.
<table>
<thead>
<tr>
<th>Set</th>
<th>Description (# feats)</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Questioner features (4)</td>
<td>reputation, number of previous Qs, ...</td>
</tr>
<tr>
<td>B</td>
<td>Activity &amp; Q/A quality (8)</td>
<td>highest answer score, highest answerer rep, ...</td>
</tr>
<tr>
<td>C</td>
<td>Community processes (8)</td>
<td>average answerer reputation, # comments on answer by highest reputation answerer, ...</td>
</tr>
<tr>
<td>D</td>
<td>Temporal processes (7)</td>
<td>average time between answers, time for highest-scoring answer to arrive, ...</td>
</tr>
</tbody>
</table>
We perform feature selection and end up using 8 important features ($S_8$):

<table>
<thead>
<tr>
<th>Feature</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of answers</td>
<td>+0.61</td>
</tr>
<tr>
<td>Sum of answer scores</td>
<td>+0.47</td>
</tr>
<tr>
<td># of questioner’s questions (log scale)</td>
<td>-0.46</td>
</tr>
<tr>
<td>Length of highest-scoring answer</td>
<td>+0.38</td>
</tr>
<tr>
<td>Questioner’s reputation (log scale)</td>
<td>+0.31</td>
</tr>
<tr>
<td>Time for highest-scoring answer to arrive</td>
<td>+0.22</td>
</tr>
<tr>
<td># comments on highest-scoring answer</td>
<td>+0.19</td>
</tr>
<tr>
<td># comments on highest-reputation answerer’s answer</td>
<td>+0.17</td>
</tr>
</tbody>
</table>

Compare against “crowd-sourced” baseline: # favorites on question and question score (upvotes-downvotes) – 2 explicit mechanisms that measure value
Results

Features of the community processes that underlie the creation of the entire question page are useful for discovering long-term value at a very early stage
**Task 2**: Predict whether a question has been sufficiently answered

Setup: Given features of a question page, determine whether the question is about to accept one of the existing answers or offer a bounty

- Same logistic regression framework (with a balanced dataset)
- No natural baseline, so we compare our 4 classes of features
- Again perform feature selection, narrow down to set of 18 features
Results – Task 2

– Questioner features are powerful

– But adding features of community + temporal processes significantly boost performance

Features of the community processes underlying Q&A activity can provide important early indications
Conclusion

Q&A sites have evolved into focused communities

We suggest a shift in perspective from question-answer pairs to viewing questions together with their complete set of answers as one unit

There is useful information in the community and temporal processes for tasks like predicting long-term value and deciding if a question needs help
Thanks!