A Question-Answering Dataset Extracted From Stack Overflow

Zainab Zolaktaf

May 31, 2017

1 Brief Summary

This document describes the procedure followed for extracting a Question-Answering dataset from Stack Overflow. The dataset is the result of research conducted in the MALNIS group at Dalhousie University. We used it to evaluate our Question Answering Topic Model (QATM) for both Question Answering task and Automatic Tagging tasks in [1]. Please cite the corresponding paper if you make use of this dataset:

```
Bibtex:
@inproceedings{zolaktaf2011modeling, title={Modeling community question-answering archives},
author={Zolaktaf, Zainab and Riahi, Fatemeh and Shafiei, Mahdi and Milios, Evangelos},
booktitle={Proceedings of the Workshop on Computational Social Science and the Wisdom of Crowds at
NIPS}, year={2011}}
```

File Name	Description
createXMLdata.pl	Extracts the dataset from posts.xml in the data dump (usage: ./createXMLdata.pl TrainQuestionNames.txt TrainAnswerNames.txt TestQuestionNames.txt posts.xml dataset.xml > out.txt)
out.txt	Disregard this, it is the output generated from createXMLdata.pl
dataset.xml	The dataset
TestQuestionNames.txt	Contains 822 questionIDs for testing
TrainAnswerNames.txt	Contains 15822 answerIDs for training
TrainQuestionNames.txt	Contains 4184 questionIDs for training
TrainQuestionAnswerNAmes.txt	Contains 15822 pairs of TrainQuestionID-TrainAnswerID for question answer pair retrieval
testdups852.txt	Contains 852 pairs of TestQuestionID TrainQuestionID, provides the duplicates

Table 1: Files included inside package.

2 Stack Overflow

Stack Overflow is a programming Question and Answering (Q & A) website, where developers can share technical information amongst themselves. To maintain an archive of high quality questions and answers, Stack Overflow employs popularity voting and allows users to vote upon and edit questions and answers. The users' contribution to the website is represented by reputation points and badges, based upon which they are granted more moderation capabilities and permissions. An archive of the content of this website is released every two months. For our experiments, we used the January 2011 Stack Overflow data dump. Some statistics of this data are given in Table 2.

#Questions	1,188,585
#Answers	2,939,767
#Tags	27,659
#Users	440,672
Avg #answer per question	2.4818
Avg #tags per question	2.9254
Avg score of questions	1.4967
Avg score of answers	1.8478

Table 2: Stack Overflow statistics, January 2011

2.1 Main Entities on Stack Overflow

1. Questions: are posted by users. Their properties include: (a) Tags (b) Author (c) Total points (d) Votes (e) Favorite (f) Can be commented on, edited, revised

2. Answers: Answers posted to questions by users. Properties include (a) Author (b) Point (c) Votes (d) Can be commented on, edited, revised

3. Users Members of the community who post or answer questions. Properties include (a) Profile (b) Reputation, profile views, badges, etc (c) Questions asked (Question ID, how many times the question has been favorite, number of answers, number of views, tags, person who answered and his reputation) (d) Questions answered(Question ID, total number of votes for the answers) (e) Total number of votes (positive and negative) (f) Total number of tags (g) Total number of badges (h) Detailed overview of activity(badges awarded, comments, answers, revisions, etc) (i) Reputation (j) Favorites (k) Accounts

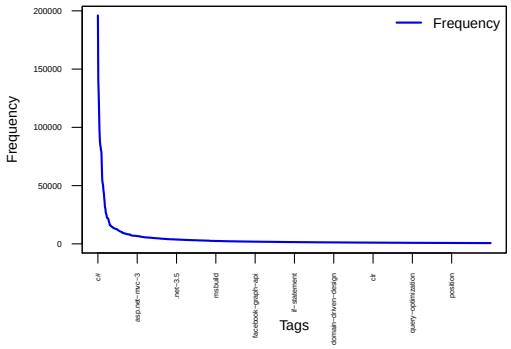
3 Dataset Extraction

In the following, we provide a brief summary of main intuitions followed for dataset creation. Appendix A provides more detailed description.

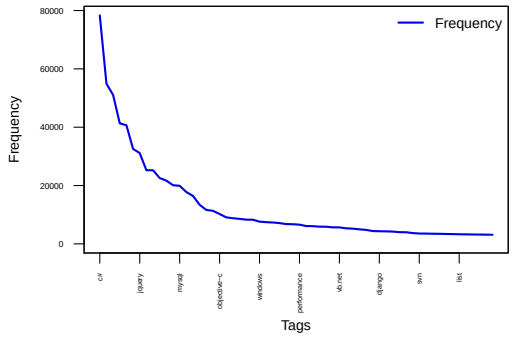
3.1 Train Data

When a question is posted on Stack Overflow, it is tagged with labels or tags. To extract a representative subset of questions and answers from the large archive available on this website, we initially extracted a smaller subset of tags. For this, we examined tag frequency and tag co-occurrence statistics, as shown in Figure 1, and identified three criteria for tag selection:

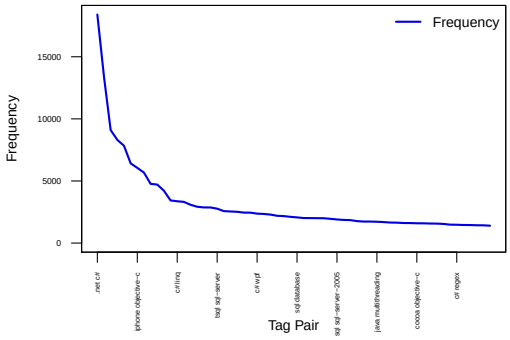
1. To ensure that there would be sufficient data available for training the model, popular tags were required.



(a) Tag frequency of top 1000 tags



(b) Frequency of top 60 tags, from posts with more than two answers



(c) Pairwise frequency of top 60 tags, from posts with more than two answers

Figure 1: Tag frequencies.

2. To maintain a similar tag distribution as the original data collection, and construct a realistic dataset, tag overlap (or tag co-occurrence) had to be preserved.

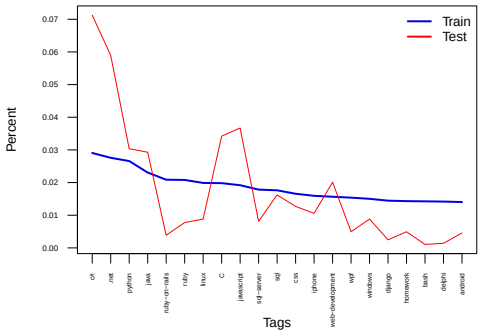
3. However, to ensure the model could uncover challenging patterns in the data, we required a set of conceptually overlapping tags and a set of distinctly different tags with distinguishable keywords.

With regard to the three criteria, we manually selected a total of 21 tags. We chose 7 tags that were popular and that co-occurred frequently with other tags, 7 tags that were popular but co-occurred less frequently, and 7 tags that were popular and rarely co-occurred with other tags. The selected tags are shown in Table 2a. Subsequently, for each tag we randomly collected 200 questions (4200 questions in total).

The content on Q&A websites is user-generated and therefore, numerous questions and answers with varying quality levels have been created. However, on Stack Overflow, users score posts based on their relevance and correctness. As a result, correctly posed questions or more relevant answers usually have higher scores. To allow the model to correctly learn the topic dependencies of a question and its answers, we extracted the 4 most relevant answers for each question using the scores given to answers by users based on their perceived relevance and correctness. At the end of this step we had extracted 15822 question-answer pairs for the train dataset. Appendix A.1 describes a summary of the train data creation steps.

C#	.net	Sql
Sql-server	Java-script	css
Java	Ruby	Ruby-on-rails
Wpf	iphone	Web-development
Android	Windows	Delphi
Django	Python	C
Bash	Linux	homework

(a) Subset of 21 selected tags



(b) Tag distribution in train set and test set.

3.2 Test Set

3.2.1 Question Answering

To evaluate the answer retrieval performance of our model, we extracted a set of questions from Stack Overflow referred to as *duplicates*. These are questions that are similar to one or more existing questions in the archive but use different words and structure. Because they increase the archive's redundancy, duplicates are considered as a negative feature of CQA websites. Therefore, Stack Overflow users identify, vote upon and close such questions.

However, there are also benefits to having alternates for the same question. Users often tend to search the archive using different terminology; duplicates increase visibility and make it more likely for users to find relevant information. From the viewpoint of our research however, duplicates are extremely important because they provide multiple variants for a question. We create a gold standard set from these duplicates and incorporate them into our evaluation paradigm. Appendix A.2 gives a detailed description of the duplicate extraction process.

Without duplicates, a possible methodology for evaluating the models performance, would have been to conduct a user study. However, there are some disadvantages with user studies. In particular, they are expensive to conduct, both in terms of user's time and compensations. A significant amount of time and effort is required to carefully consider various components of the study and conduct successful user studies. Experiments have to be designed so that they provide robust answers to the questions of interest and lead to statistically significant results, the participants have to be chosen so that they represent the true population of users, future information needs have to be predicted, and important decisions must be made based on vague expectations. In addition analyzing and interpreting experimental results can be hard. Typically, the outcome of studies can only answer small questions, and larger conclusions cannot be drawn.

Duplicates enabled us to incorporate perceptions of the end-users of the technology into the evaluation without the need to conduct a user study. They also represent natural information seeking behavior which is important because the results are real world and can lead to valid generalizations.

On Stack Overflow three classes of duplicate questions have been identified and different handling mechanisms have been defined [1]:

[1] Refer to:

1. `http://blog.stackoverflow.com/2009/04/handling-duplicate-questions/`
2. `http://meta.osqa.net/questions/1086/`
3. `http://blog.stackoverflow.com/2009/05/linking-duplicate-questions/`

1. Cut-and-paste duplicates: These questions are the exact copy of a previous question. They are the very definition of exact duplicates. This group of duplicate questions are voted down by users and flagged for moderator attention. They are then deleted from the system by the moderators.

2. Accidental duplicates: These questions are semantically similar but lexically different from a previously asked question in the archive. They have the same answer. Stack Overflow users identify, vote upon and close them as "exact duplicates" of another question. They link these duplicates to the original question by posting the URL of the original question as a comment or edit in the duplicate.

3. Borderline duplicates: The questions in this set cover the same grounds as a previous question in the archive, however their overlap with those questions is ambiguous. This means that their uniqueness and commonality is subject to interpretation. These questions are tagged by users of Stack Overflow so that they naturally group with other relevant questions.

We extracted a set of accidental duplicates for questions in our train data and created a gold standard set. Examples of accidental and borderline duplicates are given in Appendix B.

3.2.2 Automatic Tagging

Note, our dataset can also be used for automatic tagging. In particular, we can evaluate the clustering quality of our model against a *gold standard* available in the data on the website. This gold standard consists of the tags assigned to questions in our training dataset by users of Stack Overflow (refer to "dataset.xml" to find tag information).

3.3 Dataset Tag Statistics

Figure 2b compares the distribution of tags in the training and the test set. The figure shows that the tag selection procedure has resulted in balanced classes of tags in the training data. In addition, even though there are some discrepancies between the test and train set tags, they generally follow the same probability distribution (the data samples come from the same statistical populations).

A Dataset Extraction Details

A.1 Train Data

Train Data Preparation Steps:

1. We collected statistics about tag frequencies and also tag co-occurrence frequencies. To be specific, the tag frequency of questions with more than two answers was calculated. In addition, the pairwise tag frequency of questions with more than two answers was calculated.

2. These statistics were examined, and with regard to our objectives we manually selected a total of 21 tags. We chose 7 tags that were popular and that co-occurred frequently with other tags, around 7 tags that were popular but co-occurred less frequently, around 7 tags that were popular and rarely co-occurred with other tags. The tags in Table 2a were chosen.

3. For each of the tags selected in the previous step, we randomly collected 200 questions. 4200 questions were extracted.

4. For each of the questions selected in the previous step, we extracted at most four of its most relevant answers. At the end of this step, we had extracted 15,822 answers.

Note, some of the questions had less than 4 answers. The 15822 question-answer pairs correspond to 4184 questions. There are two reasons for the decrease in the number of QAs from 4200 to 4184: 1) Around 12 of the Questions were invalid, meaning that they were identified as duplicates of other questions that existed in our train data, so these duplicates were removed (4200 -12 = 4188) 2) 4 of the questions selected in the second step did not have answers and so were dropped, that left us with 4184 valid QAs.

A.2 Test Data (Duplicate Questions)

To compare the answer retrieval performance of our model with existing methods, we extracted a ground truth set from the accidental duplicates. To extract the duplicates the following steps were taken:

1. Duplicate questions that had been voted on and closed as exact duplicates were identified and extracted. 5783 questions were marked as duplicates in the January data dump.

2. This group was refined to extract questions that were duplicates of the questions in our train dataset. 822 duplicate questions were found. Since each duplicate question may be a duplicate of multiple questions and a question may have multiple duplicates (cardinality of the relation is many to many) these 822 duplicate questions correspond to 852 relations.

3. The 822 duplicates were split into two subsets consisting of 700 duplicate questions for the train set and 122 duplicate questions for the validation set. The 700 duplicate questions correspond to 722 duplication relations and the 122 duplicates in the validation set correspond to 130 duplication relations.

B Examples of Duplicate Questions on Stack Overflow

1. Original Question (Train 113547)

   URL `http://stackoverflow.com/questions/113547/`

   Title iPhone development on Windows

   Content Is there a way to develop iPhone (iOS) applications on Windows? I really don't want to get yet another machine. There is a project on http://code.google.com/p/winchain/wiki/HowToUse that seemed to work with iPhone 1.0, but had limited success with iPhone 2.0, plus it requires all the Cygwin insanity.Is there anything else, or do I have to buy a Mac?

2. Accidental Duplicate (Test 68196), lexically similar to original question

   URL `http://stackoverflow.com/questions/68196/`

   Title Develop iPhone applications using Microsoft Windows [closed]

   Content I don't have a mac, but I wish to develop iPhone applications on Windows platform. Is this possible?

3. Borderline Duplicate, semantically related to the original question

   URL `http://stackoverflow.com/questions/22358/`

   Title How can I develop for iPhone using a Windows development machine?

Content Is there any way to tinker with the iPhone SDK on a Windows machine? Are there plans for an iPhone SDK version for Windows? The only other way I can think of doing this is to run a Mac VM image on a VMWare server running on Windows, although I'm not too sure how legal this is.

4. Borderline Duplicate, semantically related to the original question

   URL `http://stackoverflow.com/questions/2642877/`

   Title Best windows iphone app development alternative

   Content What do you think is the best way to develop iphone apps on windows? What are the pros / cons of your method, and why do you use it over other options? How complex is your method in relation to other options? I am more interested in standalone and web apps but fell free to discuss gaming graphics. Yes I know you need to build on a mac to be able to put it on the app store, so no "use a mac" answers please.

5. Borderline Duplicate, semantically related to the original question

   URL `http://stackoverflow.com/questions/2261267/`

   Title iPhone development on PC

   Content Can anybody shortly describe solutions to start develop for iPhone on PC?

6. Borderline Duplicate, semantically related to the original question

   URL `http://stackoverflow.com/questions/2438718/`

   Title developing iphone apps on windows is it worth the hassel

   Content I'm only after a simple solution and won't be developing anything particularly complex. But I'm wondering whether the hassals of developing an iPhone app NOT on MacOS are really that significant to avoid giving it a shot. Bearing in mind that I do have access to a mac every now and again. So I would be able to compile it using the official Apple supported SDK, but I just want to be able to develop it in my own environment (windows laptop). I heard someone mention a while ago that there are various objective C compilers that allow writing code in various other web technologies as well. Are these really valid. And am I alone in thinking Apple's whole attitude towards this is totally imoral. Charging 200 for the privelege of having your app unequivocally rejected etc etc and then not being allowed to look directly at Steve Jobs or his golden retrievers.

References

[1] Zainab Zolaktaf, Fatemeh Riahi, Mahdi Shafiei, and Evangelos Milios. Modeling community question-answering archives. In *Proceedings of the Workshop on Computational Social Science and the Wisdom of Crowds at NIPS*, 2011.