User Tendency to Write Closed Questions on StackOverflow
CSE 190 Assignment 2

Atl Arredondo
UC San Diego
San Diego, CA
ararredo@ucsd.edu

Israel Cruz
UC San Diego
San Diego, CA
i3cruz@ucsd.edu

Abstract

In this projects, we looked at a dataset of around 4 million Stack Overflow questions with the goal of classifying whether a was closed. Questions on Stack Overflow are closed when they do not follow the guidelines set by StackOverflow. The guidelines involve that a questions has to be an actual questions, not a duplicate or out of topic. By closing questions Stack Overflow tries to obtain the best content on their website. Our objective for this task was to use the tools we learned from CSE 190 Data Mining and Analytics by Professor Julian McAuley to predict the current status of questions posted on the website StackOverflow. This was a classification task classify a questions as open or closed.

1. Introduction

For this assignment, our group consisting of two members, Atl Arredondo and Israel Cruz searched through various data sets in order to find a set with data we could manipulate to find interesting results. Our search ranged from taxi and uber data to twitter hashtags.

Our initial search started when we came up with our first idea for the project, which was the ambitious task of predicting the amount of attention a terrorist attack or disasters received from social media. We planned to look at which countries tweeted about which types of disasters, however we quickly found that gathering data for this task was too difficult.

Another problem we ran into when choosing our task was insufficient data entries. We ran into this issue when exploring a dataset that was used to predict personality traits, which only contained a little over 1000 data entries and another dataset that predicted a user's influence on a social network which only contained around 5000 data entries. Although these data sets would have been interesting to explore we could not use them due to lack of data.

Ultimately we decided on using a dataset we found on Kaggle as it was big and diverse enough for our project's needs.

The Kaggle competition consisted on predicting if a questions was going to be closed or open. For this task a dataset of 3.8Gb on a csv file format was provided. Around 4 million questions exist on this dataset, however we were unable to explore all this data due to a lack of computational power. As a consequence we had to restrict our scope to only the first 1.5 million questions from the dataset.

2. Exploratory Analysis

For the project we decided to use StackOverflow data we found on Kaggle due to its vast size. The data is structured by the following 15 fields:
Input, PostCreationDate, OwnerUserId, OwnerCreationDate, ReputationAtPostCreation, OwnerUndeletedAnswerCountAtPostTime, Title, BodyMarkdown, Tag1, Tag2, Tag3, Tag4, Tag5, PostId, PostClosedDate, and an OpenStatus,field, which was the field used for the Kaggle competition.

The "PostCreationDate" field contains the date and the time the post was created. The "OwnerUserId" field contains the user's identification number. The "OwnerCreationDate" field contains the date and time the user created their StackOverflow account. The "title" field contains the title of the post for the question, which is usually the question or what the topic about the question asked. The "BodyMarkdown" field contains the body of the post. The "Tag" fields contain tags relevant to the question asked. The "PostId" field contains a number with the post identification number. The "PostClosedDate" field contains either the date and time the post

was closed or is left blank. Finally the "OpenStatus" field is either marked as "open" if the question has not been closed or if the question has been closed it contains the reason why the post was closed, which is "not a real question", "too localized" or "off topic".

These features did not give us direction at the beginning, but after looking at the distribution of the ReputationAtPostDelete and the Tag frequency over the data, we guess that these might be valuable features that could have an impact on the data. Around 90% of the users had a ReputationAtPostDelete below 200 and on the closed questions this number was even lower. Also, most of the questions used the top 10% tags.

When we looked at the question body the number of words varied a lot between questions. Also, some bodies contained code and more details about the questions. Normally the questions that had code were most of the time open questions, but we did not have the time and tools to identify each of the questions that contained code on them.

Finally, when we started with out our third model we observed that from the sample data used to train the model most of the tags that were correlated with closed questions were not that frequent on the dataset.

From the 5066 tags correlated with closed questions only the first 20% stand out to be on average frequent on the dataset. This lead us to restrict the feature representation of the model by using less tags to train it.

3. Predictive Task

Initially our predictive task was to remain the same as the one featured in the competition, which was to build a classifier that predicts whether or not a question will be closed given the question as submitted. The competition entailed categorizing the closed questions by 4 reasons. We modified the task to only predict if a questions was going to be closed or not. We look at the Classification Accuracy and the ratio between True Positive and False Negatives to determine the performance on our models.

We were able to retrieve the baseline model used to during the competition for use in our project. This model worked by randomizing the classification using a Random Forest Classifier which lead to a 0.46 Classification Accuracy. Our goal was to beat this benchmark by looking mostly on the closed questions answered right.

We ran into trouble when we tried to run all of the data on our machines at home. To solve this problem we split the data and only used the first 1.5 questions to obtain a sample dataset of 200,000. Since the questions were ordered in chronological order our sample dataset only looked at the years from 2008 to 2010.

We decided to use Logistic Regression and Support Vector Machines because our model had a categorical prediction task. Both of these were also regularized using Stochastic Gradient Descent. The hardest part of the task was to obtain the features that will give us the best on performance and train complexity. Some of these features were so complex that Support Vector Machines were too slow and unable to give us a valuable model.

4. Model

First we looked at the first 1.5 Million questions and gather a train sample of the first 100,000 questions.

From this small sample we started by analyzing the features that might correspond to a closed questions. After looking at the tags and their frequency on the closed questions we decided that it was a good way to start. We created a model using logistic regression using all the existing tags found in the train sample questions, the ReputationAtPostCreation and OwnerUndeletedAnswerCountAtPostTime. This became a really complicated model to train since there were around 16,000 distinct features, which lead to around 45 minutes of training. The performance of the dataset was far from perfect. We tested the set using the next 50,000 questions on the dataset. Although we were able to achieve a 97% Classification Accuracy, only 3% of the questions were closed questions on the test dataset. This lead us to change the sample data to one that resembled the nature of the whole dataset. Also, after looking at the feature weight we were able to analyze the Tags with the highest weight and the ones with the least weight. This helped us find that there were specific tags that just lead to a closed question. The Tags are located under results on Model 1.

After, we had this experience on Model 1 we decided to change our train and test dataset to have 6% closed questions. For this we took the first 850,000 questions and we created a 200,000 question with 6% as closed questions. From this dataset 120,000 questions were assigned randomly to the train set and other 50,000 questions were assigned to the test set.

When we acquired the data we tried to do the previous approach using Logistic Regression. The results were not far from the previous model, however we started to notice that the ratio between the True Positives and the False Negatives started to decrease.

From this stage we tried to use Stochastic Gradient Descent on Logistic Regression and with a Support Vector Machine. Both of these approaches failed to give us better performance. We assume that the cause was because of the incredible complexity of our mode.

Later, we looked back at the Tags used as features and we did an analysis to observe which Tags had higher value than other. Since we wanted to predict the closed questions our focus was on Tags that were highly correlated to a closed question. From the 20,000 Tags only 5,066 Tags were used on closed questions. We used these tags and assigned each of them a ratio of frequencyClosedQuestions/ frequencyAllQuestions to describe its value over all the tags. From this ratio we were able to determine which tags had higher value than others, since the higher the ratio the more this tag was used on closed questions. But this was only one part of the story. We were not sure how frequently these tags were used on the whole train set. This lead us to order the Tags by their overall frequency on the questions and pick the top 2,000 as our new feature representation . Using this procedure we used Logistic Regression with the top 1,000 , 2,000 and all of the most frequently closed question Tags based on their weight.

With this feature representation we were able to highly increase the ratio of True Positive/ False Negative classification while maintaining a Classification Accuracy of 94%.

At the end our model looked as following:

f(QuestionTags, ReputationAtPostCreation, OwnerUndeletedAnswerCountAtPostTime) =
{ 1 if Question is closed,
0 if questions is Open.}

QuestionTags was obtained by analysing the tags by its *weight* (freqClosedQuestions/ freqAllQuestions) on closed questions and only choosing the top 1,000 and 2,000 and all most frequent tags for three models. The 2,000 features representation of the tags ended up giving us the best performance overall.

To test the performance of our model we used the Classification Accuracy (TP+TN)/(TP +TN+FP+FN) and we focus on increasing the accuracy on the closed questions which as TP/ FN.

5. Literature

Our dataset came from Kaggle, where it was used in a prediction competition similar to Assignment 1. The competition asked users to manipulate the data in order to build a classifier that predicts whether or not a question will be closed given the question as it was submitted to their site along with some additional data about the user who created the question.

This competition began on August 21st and lasted until November 3rd and concluded with 3 winners and a grand prize of 20,00$.

A contestant who ranked 10th in the Kaggle competition posted his solution which used various techniques [2]. This contestant used 5 logistic regression models taking into account number of sentences, number of words and number of special tokens for features. Another detail he used was placing special emphasis on the first and last sentences of the post's body field. Other techniques used were things not covered in class. These included a one-against-all model and a logistic loss function model.

This winning solution shared some similarities to our model. We both used logistic regression to train the data.

6. Results

Baseline Model:

Our baseline model came directly from the competition.

Model 1:

The subset of samples contained a total of 16,005 tags in which only 8,999 had a weight within . 05 and -0.05 and similarly another 1,556 tags had a weight within .01 and -.01.

From the initial model we found that the tags with the highest weight towards a closed question were:
(3.9629141965120551, 'polls'),
(3.6011583362067965, 'books'),
(3.2733806274899968, 'legal'),
(2.9575760535200808, 'enterprise-development'),
(2.8501323043584139, 'podcast'),
(2.6786220574520621, 'outsourcing'),
(2.665884061600766, 'discussion'),
(2.569109605124678, 'teamwork'),
(2.5499409434584801, 'career-development'),
(2.4798313767856413, ‘ebook')

And the tags with the least amount of weight towards a closed question were:
[(-1.7541927498124841, 'validation'),
(-1.6133853094700588, 'tsql'),
(-1.5130293812435553, 'ms-access'),
(-1.4914331363527273, 'memory'),
(-1.4872792927655318, 'wcf'),
(-1.4527833333306297, 'msbuild'),
(-1.4287056471397914, 'session'),
(-1.3818008074795778, 'class'),
(-1.3784245751732331, 'com'),
(-1.3756143910474969, 'sorting'),
(-1.3674756125471301, 'events'),
(-1.3634261307550097, 'usercontrols'),
(-1.3410296501564201, 'actionscript-3')

Ultimately we found these results made sense. One reason StackOverflow marks a question as closed is if the question is off topic and not related to programming [1]. Tags with high weights such as “books”, “legal”, and “podcast” fall within this category. Another reason for closing a question is if the post is mainly opinion based. Since the poll tag indicated that the post was not a question thus it would be closed and tags. We can also observe that tags with weights indicating a question would not be closed were related to programing.

Model 1 Classification Error:

TP	TN	FP	FN	Accuracy
386	96,463	158	2,993	96.849 %

Model 2:

In the second model we realized that the data sample we took did not correspond to the features that the main data has. So all of the data has a 6% closed question rate in our case it was a lot less. In order to maintain the same rate of closed question we used a bigger sample while maintaining the number of closed questions at 6 % to match the entire dataset. We choose 100,000 samples for the train and test set.

Model 2 Classification Error:

CA = Classification Accuracy
CLA = Closed Question Accuracy

Set	TP	TN	FP	FN	CA	CLA
TA	848	93714	286	5152	94.56 %	14.14 %
TE	76	93925	75	9392	90.85 %	0.8%

Model 3:

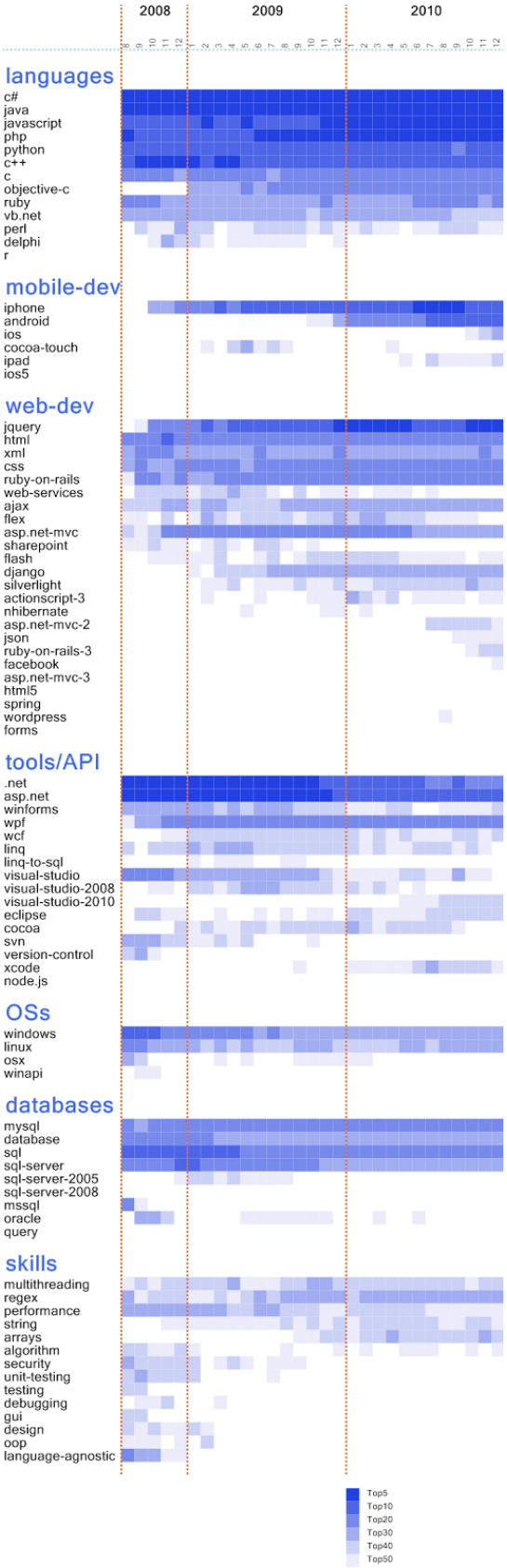
In this case we reduced the feature representation to the most relevant Tags in order to have a higher TP rate on the closed questions. Also we use a train dataset of 120,000 samples and a test set of 50,000 samples. This was in our opinion the dataset the performed the best on the unseen data.

The model that used 2,000 Tag features performed better that the model using all the 5,066 features. We think that this happened because as a model become more complex its performance agains unseen data gets compromised and as a consequence it overfit the data.

#Tags	Set	TP	TN	FP	FN	CA	CLA
1000	Train	422	112685	106	6787	94.26 %	6%
1000	Test	181	46923	47	2848	94.29 %	6.98%
2000	Train	864	112483	308	6345	94.46 %	11.99%
2000	Test	326	46827	143	2703	94.31 %	10.76%
5066	Train	886	112505	265	6344	94.49 %	12.25%
5066	Test	310	46914	126	2649	94.49 %	10.47%

Graphical Representation of the Data

The following chart represents the most popular tags over the data over the course of two years and 5 months. The popularity of tags is denoted by how darkly shaded a box is during that time period. The tags are separated into relevant categories.



Conclusion

At the end we managed to beat the baseline code given during the competition and came up with thought process that helped us analyze the Tags more efficiency to come up with a smaller feature representation of the model. However, there were many different techniques that we were unable to use to train our model. Our biggest setback was the machines we were using. Training new models took a long time since our computer's only had 8 gigabytes of RAM each. Atl even created a script that allowed us to use school computers, however we were disappointed to find that they were even less powerful than the machines we were using for the task we were doing, with only 5 gigabytes of RAM.

One of the followed procedures to get better results is to analyze the question titles since they tend to be linked to the Tags and as a consequence to the closed questions. In addition, user preference and user tendency to write closed questions may also be a strong feature on the dataset that was not explore on this project. If more computation power was available temporal dynamics in the data may also give a better model to predict closed questions since in this type of data time change the community behavior and technologies change the way users asked questions.

Ultimately we found this project to be very enlightening and a nice way to end the course as it allowed us to apply the techniques we learned.

References

[1] "What Does It Mean If a Question Is "closed" or "on Hold"? - Help Center."*What Does It Mean If a Question Is "closed" or "on Hold"?* N.p., n.d. Web. 02 Dec. 2015.

[2]"Predict Closed Questions on Stack Overflow." *Sharing My Solution (Ranked #10)* -. N.p., n.d. Web. 02 Dec. 2015.