Multi-label Classification and Prediction of Tags for Online Platform Questions

Yeeleng Scott Vang

Abstract

Past studies have shown personalized tutoring offers students the best means to master new concepts. The current open online course (MOOC) platforms offer various means for instructors to interact with students from vote-polling to Q&A-style format. In this project, a multi-label supervised classification framework is used to classify questions posted in a Q&A-style platform to predict the appropriate tags based on the text of the title and body alone.

1. INTRODUCTION

Past studies have shown personalized tutoring offers student the best means to master new concepts. As focus shift from the traditional brick-and-mortar classroom settings to online offerings, a natural question then is to ask what online platform can best mirror personalized tutoring. Popular current platform such as Piazza uses a Q&A-style format that allows student to post a question and instructors to answer. One of the main issue with these current platforms is that there are often many repeated questions ask which ultimately leaves the collection of questions indiscernible and unusable. While a small blame can be place on the offending students, the existing search mechanism lacks intelligence to effectively return relevant results thus leaving student frustrated with no other choice but to "repost" in hopes of receiving an immediate answer. Although Piazza allows for manual tagging of questions by the original poster, as with many things in life, if two students are asked to tag the same question, there is a high probability that both will classify it with very different tags. With the total number of questions aggregates during the entirety of the course, the number of search results returned increases substantially but bares little relevance to what the student is asking. In this paper, I will show a classification framework to provide intelligence to automate tagging of questions based solely on the text of the question title and body. In particular I address this problem of multi-label classification using K-many one-vs-all logistic regression and linear SVM and compare their performances.

2. DATA SET

Data collection used is "Facebook Recruiting III - Keyword Extraction"[1] dataset from Kaggle and this collection represent questions asked from the Stack Exchange website. The training and test files were provided in separate CSV files. The training file consist of four columns: Id, Title, Body, and Tags. The test file consists of the same first three columns as the training file minus the tag column. Due to the large size of the training set and computer memory limitation, only a subset of the training file (10000 data points) is used for this project. 7000 of these data points are used to train the model and the remaining 3000 points are tested against. Some statistics are provided in the following table.

	Training Set	Test Set
No. of Data Points	7000	3000
No. of unique Tags	4891	2937
No. of Occurence of Tags	20401	8769
Max no. of tags per question	5	5
Avg. no. of tags per question	2.914	2.923

Training Set		Test Set	
Tag	Occurrences	Tag	Occurrences
C#	538	C#	240
java	492	php	222
php	480	java	211
javascript	441	javascript	183
android	375	android	143
jquery	367	jquery	137
c++	229	c++	104
python	216	asp.net	100
iphone	212	.net	96
asp.net	209	html	92

Table 1: Data set statistics

3. PREPROCESSING

Feature extraction was performed on the dataset to make the data useful for the learning algorithm. This involves replacing all non-ascii characters with white space as they would be infrequent and useless as discriminative features. Each data point's Title and Body corpuses is combine into one larger corpus. This corpus is then represented using the bag-of-words representation by tokenizing, counting, and normalizing using Scikit-Learn[2] Python library vectorizer. Tags are parsed into tokens and maintained in a list.

4. CLASSIFICATION METHODOLOGY

Classification of this data set was performed using logistic regression and SVM with a linear kernel. Since multiple labels are being predicted, Scikit-Learn Python library OneVsRestClassifier is use to model K-many one-vs-all binary classifier for each unique tag. Logistic regression is chosen as a baseline, general classifier while SVM is chosen as it has been shown to be an effective text classifier in the literatures.

To verify the accuracy of the model, the Scikit-Learn Python library f1_score is used to calculate the f1-score of the multi-label classifier taking into account the average precision and recall value for each tag instances. The learning models are tested for training set size of 1000, 2000, 3000, 4000, 5000, 6000, and 7000 data points.

5. RESULT

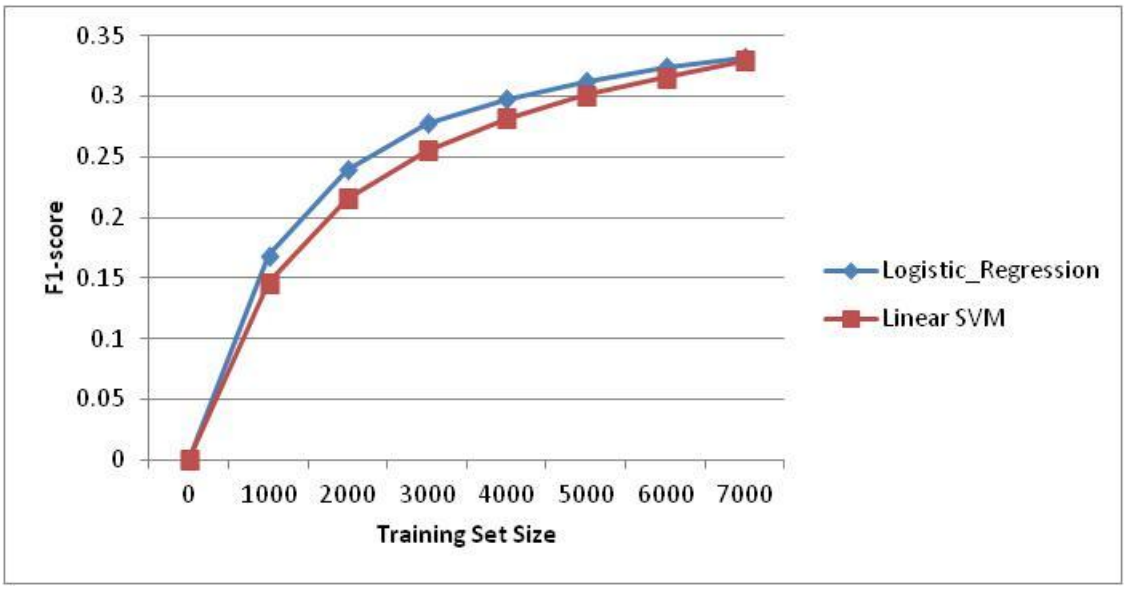

Figure 1: Classifier F1-Score

6. ANALYSIS

Logistic regression performs slightly better than the linear SVM. As the training set size increases however, it is observed that the linear SVM's performance approaches that of the logistic regression model.

The overall accuracy of both models improves as a function of training set size but remained low due to many factors. The framework of implementing K-many one-vs-all models is computationally expensive and intractable as the set of tags grow with the training set size. The large size of the original dataset means the small subset of data points used for training suffers from a high bias issue as many tags in the testing set was never seen in the training set.

7. FUTURE WORK

As the result shows, there is still large room for improving F1-score. Some of these approaches includes:

1) Instead of a batch framework, implement an online stochastic gradient descent classifier to incorporate more training data points. Although this would solve the issue of a large feature matrix, there would still be concern of computation time having to train that many more one-vs-all classifiers with the growing possible collection of tags.

2) Try running on a cloud cluster. Since computer memory is a limiting factor, potentially limitless computer memory on a cloud cluster may provide a quick, scalable solution using the current batch framework.

3) Instead of the simple 1-gram bag-of-word approach used here, use n-gram bag-of-word representation to accounts for syntactic meaning which may perform better as predictive features.

4) Approach feature selection from a statistical perspective using topic model. This may better discover correlation between text corpus and tag which would lead to superior prediction accuracy.

8. CONCLUSION

In this project, a dataset of questions and tags were used to train an one-vs-all classifier to predict the multiple tags of unlabeled questions. Logistic regression and linear SVM classifiers were developed, trained, and tested on this dataset. The performance of the logistic regression and linear SVM classifier were comparable although overall F1-score was still relatively low. This seems to be an issue of high bias where much more training point is needed to better predict the various tags that showed up in the test set and not the training set. With the enormous size of the original dataset, better feature selection and online learning framework may improve the quality and accuracy of the classifier.

REFERENCE

[1] Facebook Recruiting III - Keyword Extration, http://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction/data

[2] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.