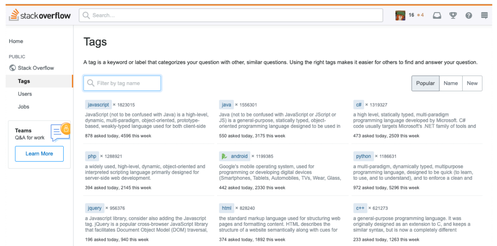
# Tag Prediction from Stack Overflow Questions

Jalal Buckley, Kevin Fuhs, Reid M. Whitaker
{jalalb91, kfuhs, reidw}@Stanford.edu

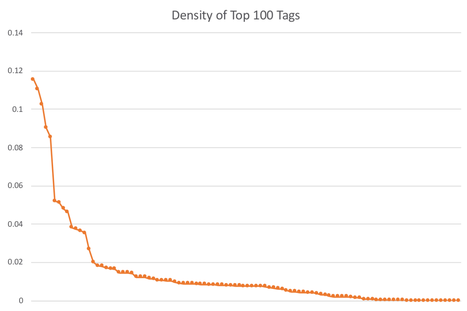
## Motivation

Keyword assignment is a central task in information retrieval and NLP. Given user-submitted questions from Stack Overflow, our project predicts the true tags which users choose for their question. Each question can contain from one to five tags, and there are many potential combinations of tags per question.




## Data

Our data consists of 7,000 questions, titles, and tags taken from Stack Overflow. Each question has between one and five tags; tags represent the themes and keywords that appear in a question. Aside from a minority of tags which occur frequently in the dataset, the majority of tags occur in a small fraction of the dataset.



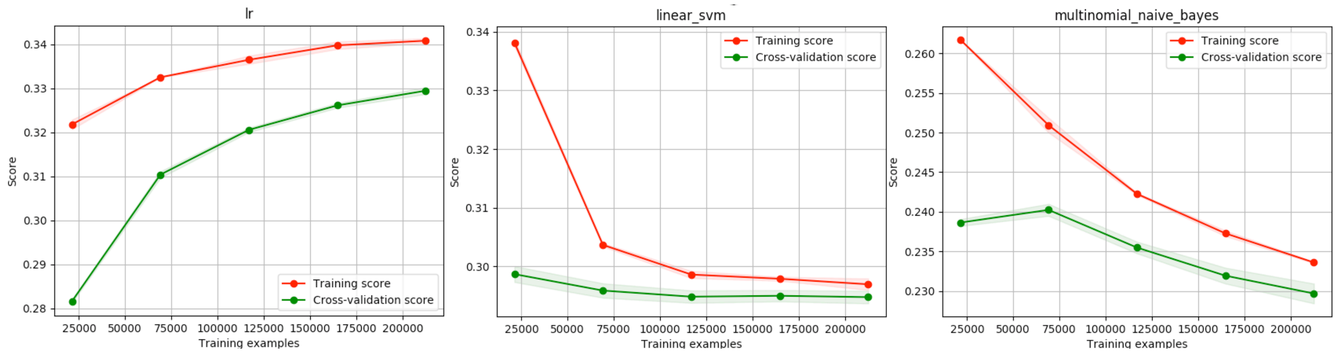

## Features

In order to create features from the question and title text, we first cleaned question text by removing all html code and markup tags, removing stop words, and replacing all words with their stems. We then created features from question and title text by using vectors in which a single column in a vector represents the presence of a word in the text vocabulary. We then selected the top 1000 title and question features using Univariate feature selection, and concatenated these into a final feature vector of 2000 features.

## Models

We experimented with Logistic Regression, Linear SVM, and Multinomial Bayes using the One vs Rest strategy, whereby a separate classifier is created for each tag. These models fit linear decision boundaries to the feature space, and assign probabilities for each tag. We also experimented with a simple Neural Network. Our Neural Network consisted of an input and hidden layer, both using the ReLU activation function, and an output layer using the sigmoid activation function.

| Tag | Associated Features |
|---|---|
| android | asynctask, spinner, apk, edittext, intent, webview, bitmap, adt |
| django | queryset, tastypie, celery, manytomanyfield, modelform |
| facebook | fql, fb, graph, wall, likes, friends, timeline, social |
| git | gitignore, github, commit, branch, commits, pull, egit, control |
| ruby | rails, nokogiri, ror, rspec, sinatra, activerecord, mongoid |




## Results

We trained on a dataset of 5,000 rows, and used a test set with 2,000 rows.

| | Training Set F1 Score (Micro Avg) | Test Set F1 Score (Macro Avg) |
|---|---|---|
| Logistic Regression | 0.632 | 0.623 |
| SVM | 0.566 | 0.563 |
| Multinomial Bayes | 0.567 | 0.562 |
| Neural network | 0.455 | 0.450 |

## Discussion

Our classifiers performed reasonably well; they succeeded in predicting at least one correct tag for many questions correctly, even if they did not predict every tag correctly.

- Because of the sparsity of the tags present in our dataset, we were curious as to whether tag prevalence in the dataset was strongly correlated with the performance of that tag's corresponding classifier.
- However, our analysis showed that a better criterion for success of a classifier is how unique its tag is, and if there are other words in the vocabulary which uniquely identify it.
- While tags which are easy to predict usually correspond to some specific keywords, tags which are difficult to predict usually cover a larger subject area, are more broad, or have many applications.

## Future

If we had more time, we would investigate neural network architectures which could make more use of the sequential nature of the data. We would also investigate using multitask learning in order to improve our neural network's performance.

## References

[1] Kaggle (2013). Facebook recruiting III - keyword extraction. https://www.kaggle.com/c/ facebook-recruiting-iii-keyword-extraction.
[2] Slobodan Beliga, Ana Me˘strovi´c, and Sanda Martin˘ci´c-Ip˘si´c. Toward selectivity based key- word extraction for croatian news. arXiv preprint arXiv:1407.4723, 2014.
[3] Eibe Frank, Gordon W Paynter, Ian H Wit- ten, Carl Gutwin, and Craig G Nevill-Manning. Domain-specific keyphrase extraction. volume 2, pages 668–673, 1999.