Predict Stack Overflow Tags

Denton Zhao

LinkedIn: linkedin.com/in/denton-zhao | Github: github.com/Denton044/StackExchange-Analysis

Introduction

- Stack Overflow is a knowledge sharing platform for users to ask and answer questions.

- Questions are organized by “tags” which are generated by the user

- Build a model that will predict the “tag” based on the underlying text of the post

- Compare and contrast different methods of document classification and vector representations of words.

SAMPLE QUESTION:

how can i make the headline sticky as the table headers? i am trying to make sticky table headers. here is what i have so far as an example of what i need.here is the code i found from someone else.here you can see a demo of what i want to achieve jsfiddle

TAGS:

Javascript, Jquery

stackoverflow

LinkedIn: linkedin.com/in/denton-zhao | Github: github.com/Denton044/StackExchange-Analysis

Process

Raw Data

- Tags
- Questions

Process and Clean

Process/Clean Text Data

- Strip HTML tags
- Strip code (only care about english)
- Stop words removed

Vectorize Text Data

TFIDF Vectorizer

- N-gram range (1,3)
- Min df 0.001

Predict Tags with Traditional NLP Algos

- MultiNomial NB
- Random Forest
- Logistic Regression
- Linear SVC

Word2Vec Skipgram

- Window Size 5
- Negative sampling
- Dimension size of 150

Predict Tags with Neural Net

- Convolution Neural Network

Overall Process

1. Process the Data
2. Vectorize the Data
3. Predict the Tags

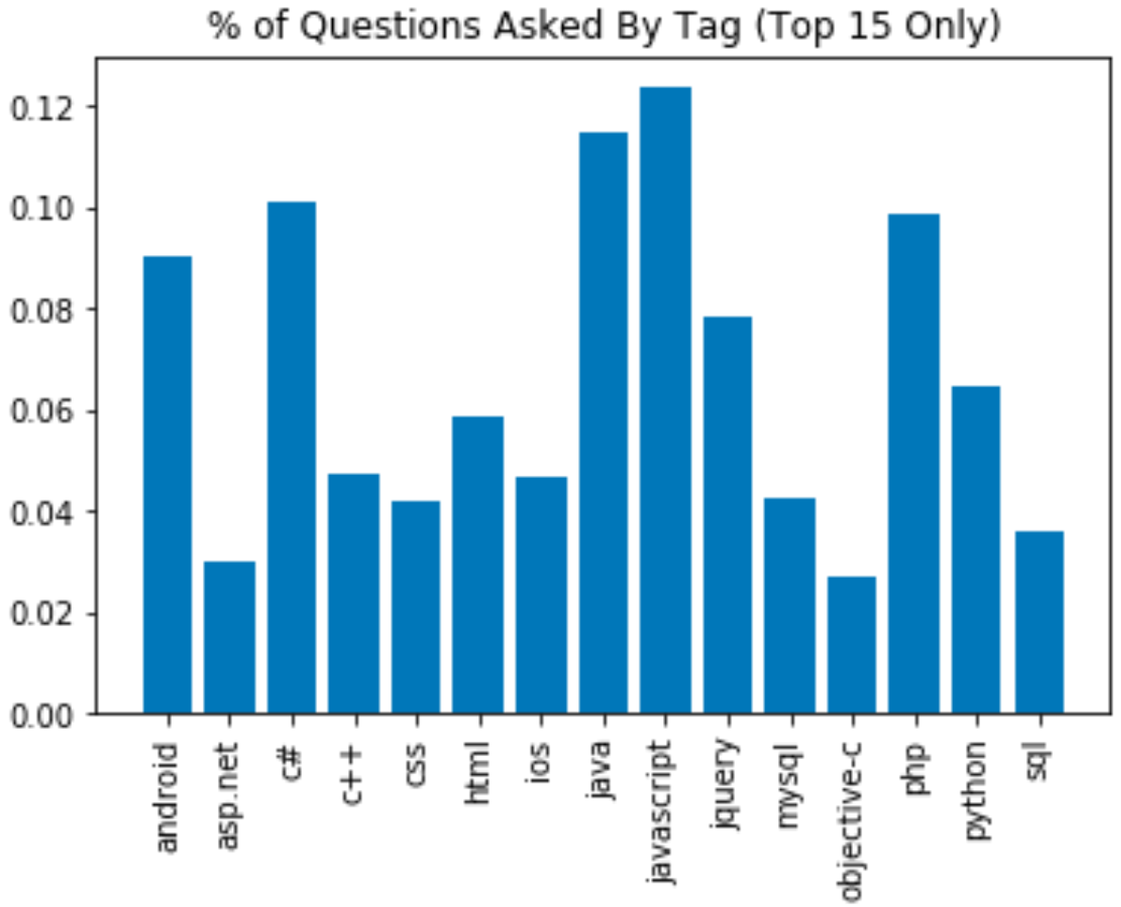
LinkedIn: linkedin.com/in/denton-zhao | Github: github.com/Denton044/StackExchange-Analysis

stackoverflow

Data Processing

Tag Handling:

- Originally our dataset contained over 36,000 unique tags
- Focus on the top 15, which came out to be ~ 1M posts
- One post can have multiple tags. (I.e. question about jquery can also be related to html)
- Multilabel classification: ~ 23% of the data had multiple tags



Text Handling:

- Title and body of the post was combined, HTML tags cleaned
- I removed the code segments from the posts, we will only be looking at the English language to attempt classify the data
- Stop words were removed from the corpus. Numbers, punctuation, white spaces as well

LinkedIn: linkedin.com/in/denton-zhao | Github: github.com/Denton044/StackExchange-Analysis

Embeddings: (TFIDF)

- TFIDF Vectorizer- tf * idf -> words that occur more frequently within a **specific** document are more important.
- From my testing implementation of trigrams worked the best
- For computational speed, used minimum df of 0.0001. (i.e. ignore terms that appear in only 0.01% of the data)
- Returns a sparse, high dimensions matrix.

TFIDF

For a term i in document j:

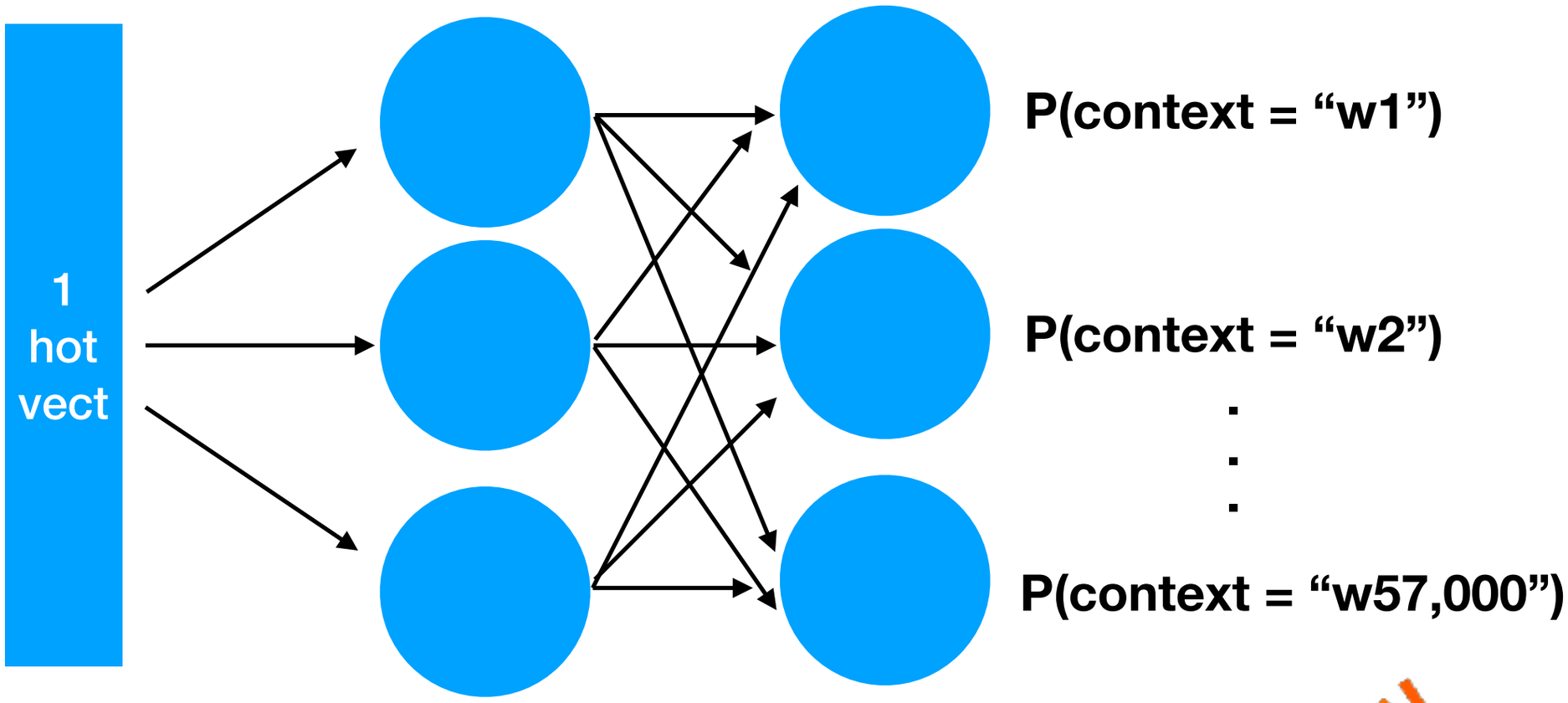
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

stackoverflow

Embeddings: (Word2Vec Skipgram)

- Word2Vec- skipgram- Given a single word (w*), can we predict the surrounding words (w1, w2…wn) within a sliding window?
- Choose to use 150 dimensional space, sliding window size of 5
- 57,000 vocabulary size



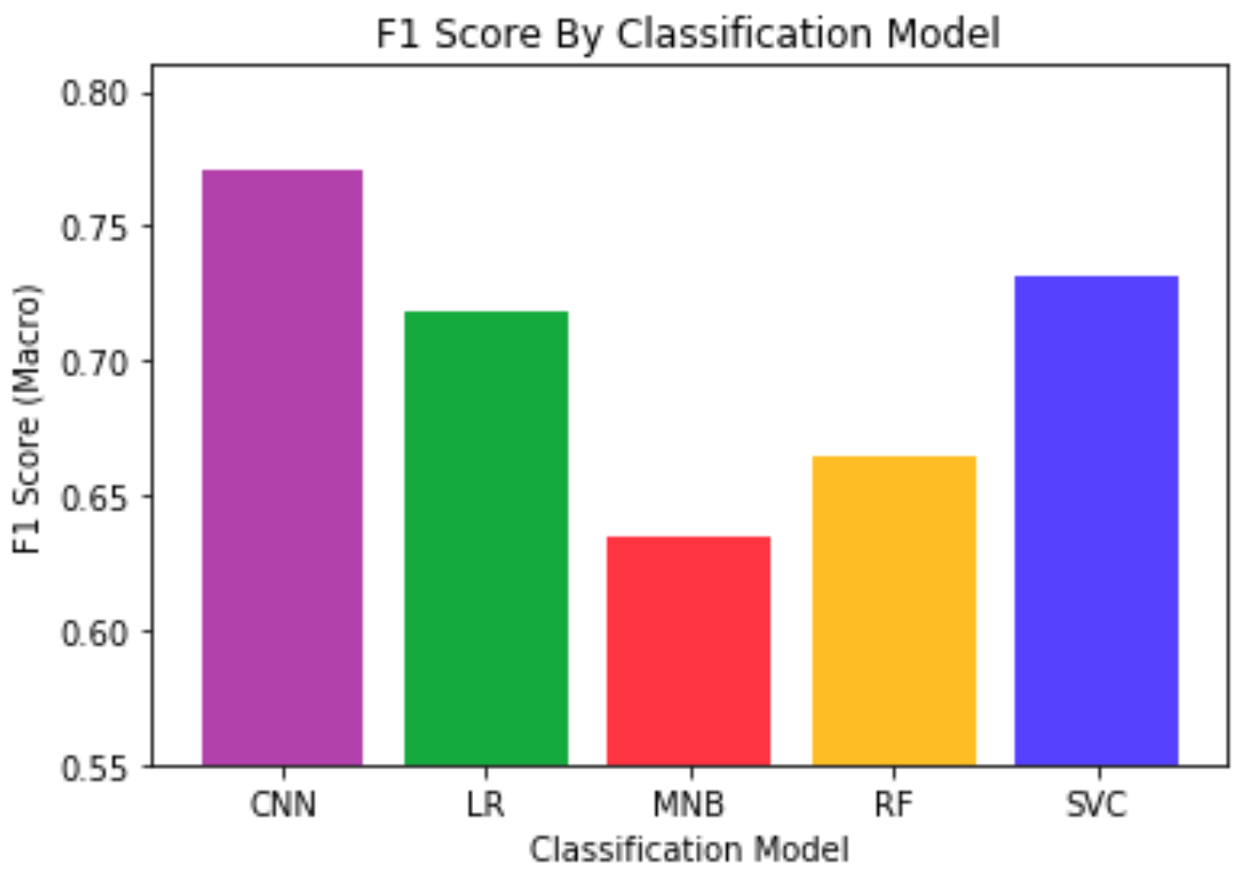
LinkedIn: linkedin.com/in/denton-zhao | Github: github.com/Denton044/StackExchange-Analysis

stackoverflow

Models

TFIDF Embeddings:

- SVC Classifier
- Multi-nomial Naive Bayes
- Random Forest
- Logistic Regression



- **Of traditional models, SVC classifier did the best. - 0.73 F1 score (Macro)**
- **My CNN model - improved the prediction to 0.77 F1 score (Macro)**

Word-2 -Vec Embedding:

- CNN with Word-2-Vec.
  - Take the first 100 words and build a model.
  - 2 layered 1 dimensional Convolution max pooling -> Flatten -> 1 layer ANN
  - (sigmoid activation with binary cross entropy loss)

Next Steps

- Different ways to clean and feature text data
    - Build model to also include <code> html tags
    - Consider stemming
    - Consider other word embeddings (GloVE)
- Change the models
    - Expand my classification a broader amount of tags (beyond top 15)
    - Consider other deep learning

LinkedIn: linkedin.com/in/denton-zhao | **Github:** github.com/Denton044/StackExchange-Analysis