



# STACK-EXCHANGE TAG PREDICTION

Bhavya Goyal  
2012CS10222

Nishant Kumar  
2012CS10239

# Problem Statement

2

- Given a question title and html markup of body, predict the appropriate tags/labels
- Kaggle Competition

read pixel value in bmp file



4



9

How can I read the color value of 24bit BMP images at all the pixel [h\*w] in C or C++ on Windows [better without any 3rd party library]. I got **Dev-C++**

A working code will be really appreciated as I've never worked on Image reading & have come to SO after Googling [if you can google better than me, plz provide a link].

c++

c

image

bmp

dev-c++

share edit flag

asked Feb 15 '12 at 15:22



Sourav

5,506



20



72



121

Sample question

# Motivation

3

- Easier Posting
- Better organization and search
- Tag synonyms
  - ▣ Eg: Java5 vs Java5.0 vs Java-5.0
- More tags for questions with less tags

# Dataset

4

- Kaggle Competition Dataset
- ~8 GB of data
- 6,034,196 (~6 million) Questions
- All 110 StackExchange sites
  - ▣ StackOverflow, MathOverflow, AskUbuntu etc
- Each question
  - ▣ Id, Title, Html markup of body, set of tags

# Dataset(cont.)

5

- ~42,000 unique tags
- Most frequent Tags :
  - ▣ C#, Java, php, javascript, android, jquery, C++, python, iPhone, asp.net, mySQL, html, .net, ios, Objective-C

# Technical Challenges

6

- Number of tags not constant

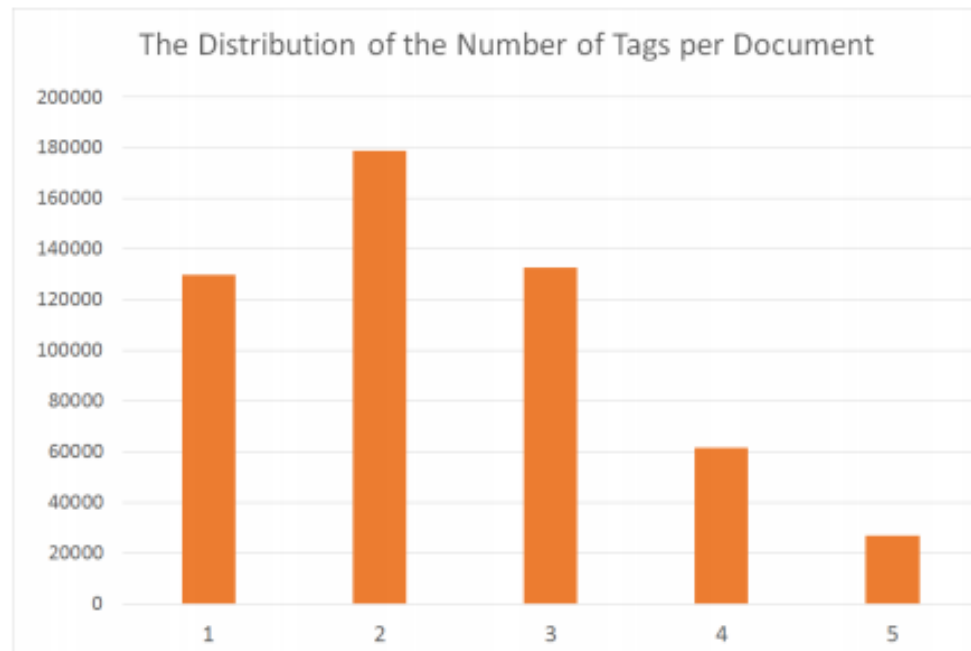


Figure 1: Distribution of the number of tags per document

# Technical Challenges(cont.)

7

## □ Tag Synonyms

- ▣ Similar objects can get tagged differently
- ▣ zombie and zombies – zombie process in Unix
- ▣ xmlparser , xmlparsing – parser of xml file
- ▣ xsltproc abbreviation of xsltprocessor

## □ Html Markups

- ▣ Not just plain-text classification
- ▣ Contains code-snippets, URLs etc.

# Baseline System

8

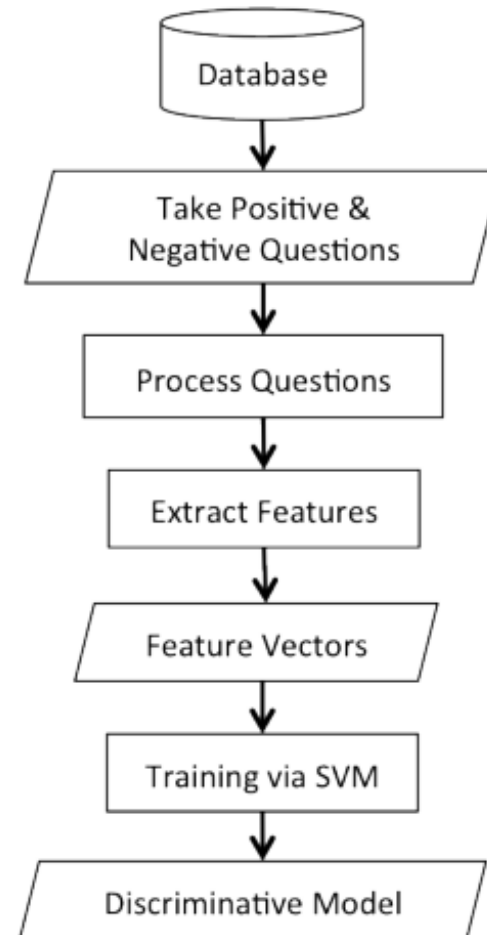
- Saha et al, A discriminative model approach for suggesting tags automatically for stackOverflow questions
- Tenth International Workshop on Mining Software Repositories, IEEE Press, 2013



# Baseline System(cont.)

9

- Train classifier for every tag (frequent)
- For new question, predict likelihood score per tag
- Take output as top k tags



# Baseline System(cont.)

10

- Code not available
  - ▣ Implemented ourselves
- Average accuracy
  - ▣ 68.47%
- Similar accuracy achieved in Part 1 of project (within reasonable fluctuations)

# Scope for improving Baseline

11

- Evaluation metric
  - ▣ Not a good criteria
  - ▣ Precision vs Recall
    - Better evaluation criteria – recall
- Synonyms tags not handled
- >73% questions have code snippets
  - ▣ Programming language detection
  - ▣ Inference based on code snippets
- Baseline – Recall@5 – ~52.4%

# Experimental Evaluation

12

- Meta – features
  - ▣ Number of code segments
  - ▣ Punctuations used in code snippet
  - ▣ Number of “a href” tags
  - ▣ Number of links occurring
- Different Classifiers – Logistic Regression, SGD Classifier, LinearSVC
- Increasing weight of title than body of question
- Recall -  $\sim 54.0\%$

# Experimental Evaluation (cont.)

13

- Doc2Vec
  - ▣ Gives continuous vector representations of documents
  - ▣ Trained on pre-processed question data
  
- Recall Drops by 1% (approx)
  - ▣ Possible reasons:
    - Overfitting due to some features
    - Doc2vec performs good on plain text, not code snippets
    - Tried with non-code text

# Experimental Evaluation (cont.)

14

- KMeans with Word2Vec
  - ▣ Used Google's Pre-trained Word2Vec model
  - ▣ Cluster word embeddings using kMeans
  - ▣ Add cluster ids to title and body
  
- ▣ Marginal improvement (less than 0.5%), not significant

# Experimental Evaluation (cont.)

15

- Tag synonymy
  - ▣ Scrapped data for synonymous tags from stackOverflow website
  - ▣ Replace all synonym tags with their master synonym
  - ▣ 3365 pairs of synonyms
  - ▣ Recall -  $\sim 55.5\%$

Master	←	Synonym
gulp-watch × 430		gulp.watch
fabric-twitter × 409		twitter-fabric × 125
ansible × 2668		ansible-playbook × 1167
swift2 × 6525		swift2.2 × 122
unity3d × 16678		unity
bluetooth-lowenergy × 2725		ble × 360
conditional-operator × 531		inline-if × 13
vs-team-services × 1888		visual-studio-online

# Experimental Evaluation (cont.)

16

## □ Term Affinity

- ▣ Measure of co-occurrence

- ▣  $Aff(tag, t) = \frac{n_{t,tag}}{n_{tag}}$

- ▣  $TagTerm_{se}(tag) = 1 - \prod_{t \in se} (1 - Aff(tag, t))$

- ▣ Recall -  $\sim 56.1\%$



# Summary of techniques

17

Technique	Recall	Accuracy
Baseline	~52.4%	68%
Meta-features/Classifiers/ Title Weight	~54.0%	
Doc2Vec	Drops by 1%	
KMeans with Word2Vec	~54.5%	
Tag synonymy	~55.5%	
Term affinity	~56.1%	75%

# Further Work

18

- Exploit co-occurrence of tags like, java and android, microsoft-sdk and C#, flask and python
- Get tag synonyms by stemming etc.
- Handle other than frequent tags

# References

19

- A. Goldbloom, “Kaggle,” <http://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction>, 2013, [Online; accessed 13-November-2013].
- K. Saha, R. K. Saha, and K. A. Schneider. A discriminative model approach for suggesting tags automatically for stack overflow questions. In Proceedings of the Tenth International Workshop on Mining Software Repositories, pages 7376. IEEE Press, 2013
- Xin Xia, David Lo, Xinyu Wang, Bo Zhou, Tag Recommendation in Software Information Sites , Proceedings of the 10th Working Conference on Mining Software Repositories, IEEE Press, 2013
- Stanford Project Document : <http://cs229.stanford.edu/proj2013/SchusterZhuCheng-PredictingTagsforStackOverflowQuestions.pdf>
- Clayton Stanley and Michael D Byrne. 2013. Predicting tags for stackoverflow posts. In Proceedings of ICCM 2013

A word cloud featuring the phrase "Thank You" in numerous languages. The words are arranged in a horizontal, cloud-like shape. The largest words are "THANK" and "YOU". Other prominent words include "GRACIAS", "ARIGATO", "SHUKURIA", "JUSPAXAR", "DANKSCHEEN", "BIYAN", "SHUKRIA", "TINGKI", "YAQHANYELAY", "TASHAKKUR ATU", "SUKSAMA", "EKGHMET", "GRAZIE", "MEHRBANI", "PALDIES", "BOLZIN", and "MERCII".

20

## Any Questions ?