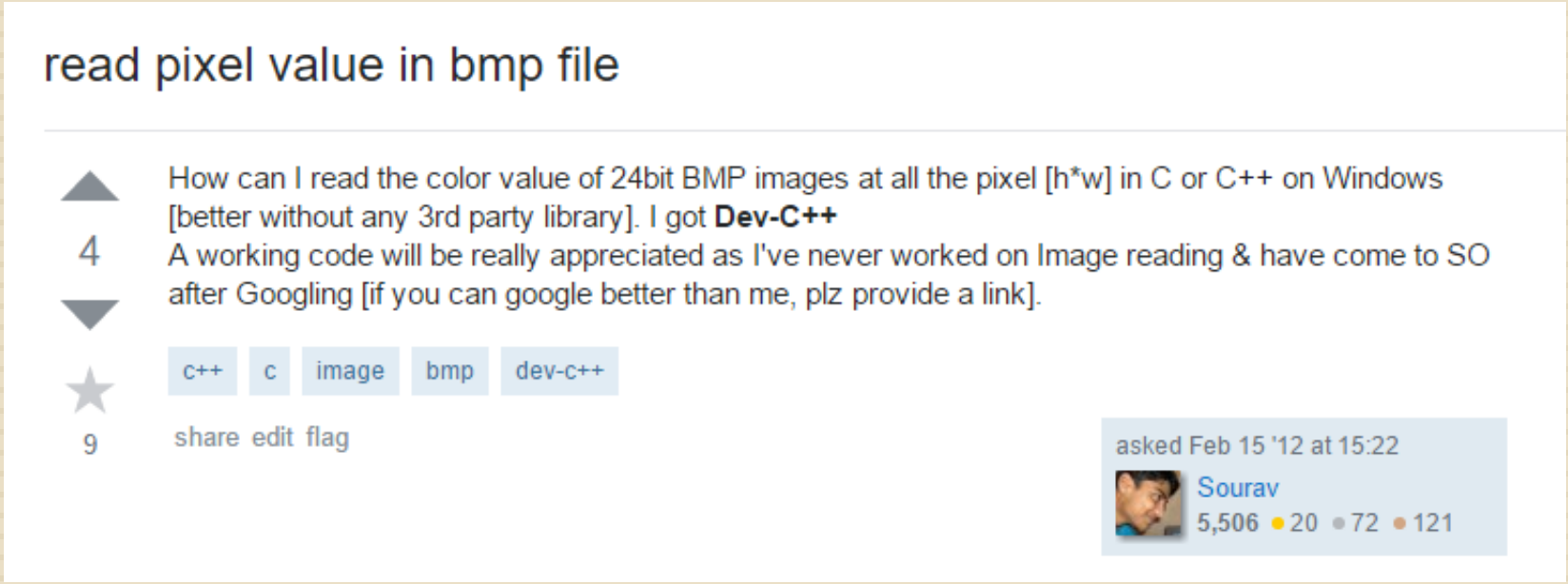
STACK-EXCHANGE TAG PREDICTION

Bhavya Goyal
2012CS10222

Nishant Kumar
2012CS10239

Problem Statement

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



Sample question

Motivation

- 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

- 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.)

- ~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

- Number of tags not constant

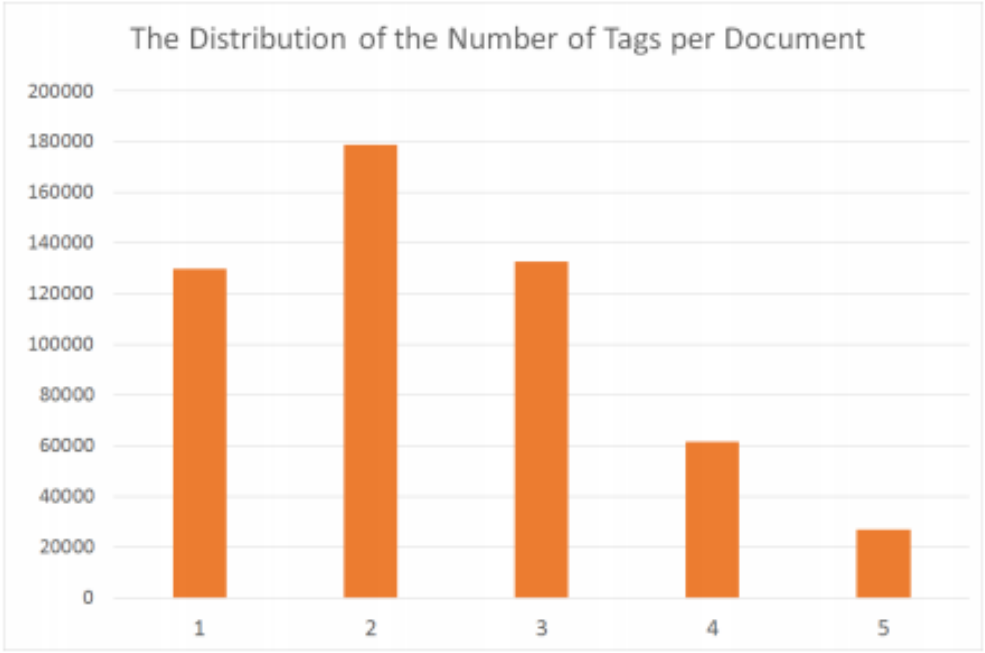


Figure 1: Distribution of the number of tags per document

Source: http://cs229.stanford.edu/proj2013/SchusterZhuCheng-PredictingTagsforStackOverflowQuestions.pdf

Technical Challenges(cont.)

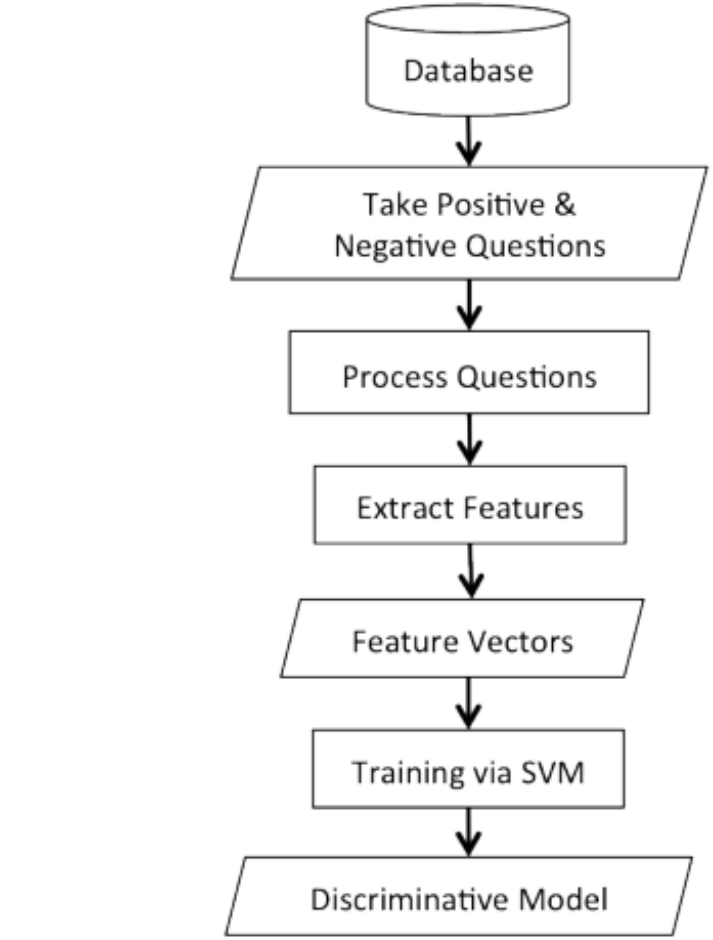
- Tag Synonyms
  - Similar objects can get tagged differently
  - zombie and zombies – zombie process in Unix
  - xmlparser , xmlparsing – parser of xml file
  - xsltproc abbreviation of xsltprocessor
- Html Markups
  - Not just plain-text classification
  - Contains code-snippets, URLs etc.

Baseline System

- 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.)

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



Source : [1]

Baseline System(cont.)

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

Scope for improving Baseline

- 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

- 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 - ~54.0%

Experimental Evaluation (cont.)

- 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.)

- 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.)

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

Experimental Evaluation (cont.)

- 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 - ~56.1%

Summary of techniques

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

- 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

- A. Goldbloom, “Kaggle,” http://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction, 2013, [Online; accessed13-November-2013].
- K. Saha, R. K. Saha, and K. A. Schneider. A discriminative model approach for suggesting tags automatically for stack overow 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

Any Questions ?