IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Tagging Questions Automatically Through Multilabel Classification

CH. Pujitha, G. Chandra Sai, G. Durga Sravani, K. Ajay Kumar Reddy

Computer Science Engineering,
NRI Institute of Technology, Vijayawada, India.

Abstract : Automatic Question Tagging using Multi-Label Classification in Community Question Answering Sites helpful to tag the questions automatically to give correct and accurate results to the users who ask the question and get answer from other users in CQA and also helps to enhance the user experience. CQA sites generally use tags to specify questions for easy navigation to users. But the task of annotating tags for the questions is not entirely left on users. Sometimes the user can not specify correct and accurate tags to some questions. But to get best answers the tags must be informative. So, we considered the problem of annotating tags to questions as a multi label classification task. Multi label classification is implemented to formulate question tagging because it creates single instance to multiple labels.

Index Terms **– Tagging Questions, Multi-Label Classification.**

INTRODUCTION

In internet everywhere we see question forums which make users to ask questions or to answer questions. Some of the question answering sites are Quora, StackOverflow etc.., In these question answering sites users ask many questions or users answer for many questions. The concept of tagging is helpful in these sites to categorize the questions for easy finding of questions. But in these questions answering sites the user needs to tag the questions manually.

But sometimes the task of tagging the questions for users becomes difficult. Also, sometimes the tags chosen by the user based on their opinion becomes inaccurate and sometimes leads to wrong connection between contents. We considered this as a multi-label classification problem. So, this automatic question tagging is very helpful to tag the questions based on the One vs Rest and linear SVM algorithm.

EXISTING SYSTEM

In natural language processing and machine learning, text classification is the process of classifying text documents into classes based on their topic. Text classification is a both multiclass and multi-label classification problem. For tackling multi-label classification we have two common approaches. They are one-vs-rest and using of set of adaptive algorithms. Adaptive algorithms include boosting and random forest. One-vs-rest which decomposes the classification problem into k independent binary classification problems by training a separate classifier for each of the k possible output labels. Adaptive algorithms tries to predict all labels at once along with a confidence ranking associated with each label.

PROPOSED SYSTEM

The dataset we use to develop our classifier is from a past Kaggle competition. The dataset consists of approximately 6,000,000 user questions spanning roughly 40,000 unique tags from the Stack Overflow site. Each post from the data contains a unique identifier, the title of the question, the body of the question, and the associated tags. The large quantity of data in our dataset caused memory to become a bottleneck for computing. Developing models for the full dataset was computationally intractable. Hence before testing any model or set of features, we sub sampled the data, typically restricting it to only those posts with one of the 20 or 100 most common tags seen in the entire data. We also ensured that our sub sampled dataset contains at least 100 training examples for each label. This ensured that we have sufficiently diverse training data so that we can learn statistics for the relevant tags. We hand-engineer features to make use of code portions of questions and improve performance on tags our classifier found most difficult to predict. We aim to enhance the performance on the 20 most common tags, as most examples have one of these tags (over 5 million of the questions had at least one of the top 20 tags). After tuning our system for the top 20 tags, we examined the worst performing tags and manually created simple string matches for commonly occurring syntax related to these tags.

ADVANTAGES OF PROPOSED SYSTEM

It would be useful to automatically infer and tag the topic of a question posted on a forum. We use Term Frequency Inverse Document Frequency, to reweight the count of each term in accordance with how relevant it is to each label.

LITERATURE SURVEY

1. Autonomous Tagging of Stack overflow Questions: The use of Support Vector Machines (SVMs) for learning text classifiers from examples. It analyzes the particular properties of learning with text data and identifies why SVMs are appropriate for this task. Empirical results support the theoretical findings. SVMs achieve substantial improvements over the currently best performing methods and behave robustly over a variety of different learning tasks

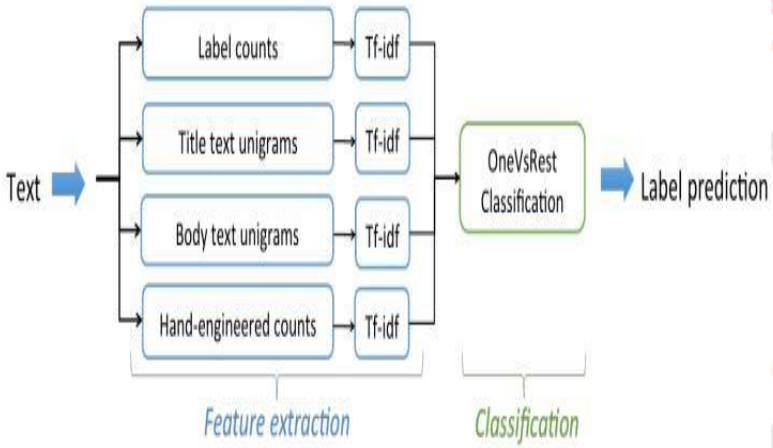
2. Stack Overflow using Statistical Methods: Web directory hierarchy is critical to serve user's search request. Creating and maintaining such directories without human expert's involvement requires good classification of web documents. In this paper, we explore web page classification using keywords from documents as attributes and using the random forest learning methods. Our initially results are promising that the random forests learning method performed better than several other well known learning methods.

3. Prediction of Automation Tags: Algorithmic Programming Language Identification based approach is one such popular technique. Various syntax highlighting tools such as Google Code Prettify will automatically highlight syntax given some code. However, these tools do not actually identify languages; instead, they use heuristics that will make the highlighting work well. In the case of Google Code Prettify, broad grammars (such as Clike, Bash-like, and Xml-like) are preprogrammed. These grammars are then used to scan code, and the best matching grammar is used in highlighting. Clearly, languages that share a grammar cannot be distinguished between. More relevant is Source Classifier, which attempts to identify a programming language given some code. However, it relies on a simple Bayesian classifier. Its strength is therefore limited to the quality of training data, and it can easily be thrown off by strings and comments.

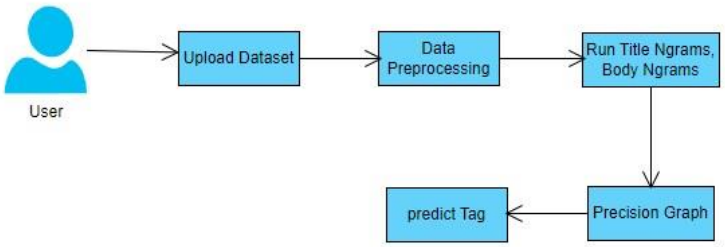
4. #ML NLP Automation Tagging: The other popular approach is automatic content tagging. In one paper [7], Xia et al. propose an automatic tag recommendation algorithm TagCombine. There are three components of Tag Combine, each of which tries to assign the best tags to untagged objects: (1) multi-label ranking component, which predicts tags using a multi-label learning algorithm, (2) similarity based ranking component, which uses similar objects to recommend tags.

ARCHITECTURE

Text to Label Prediction



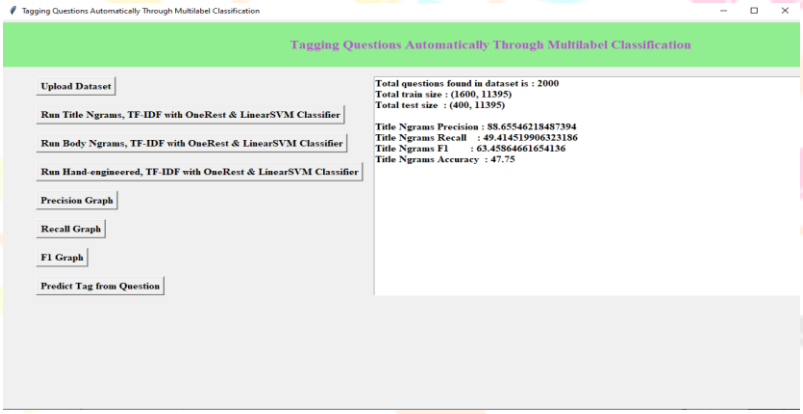
Analysis



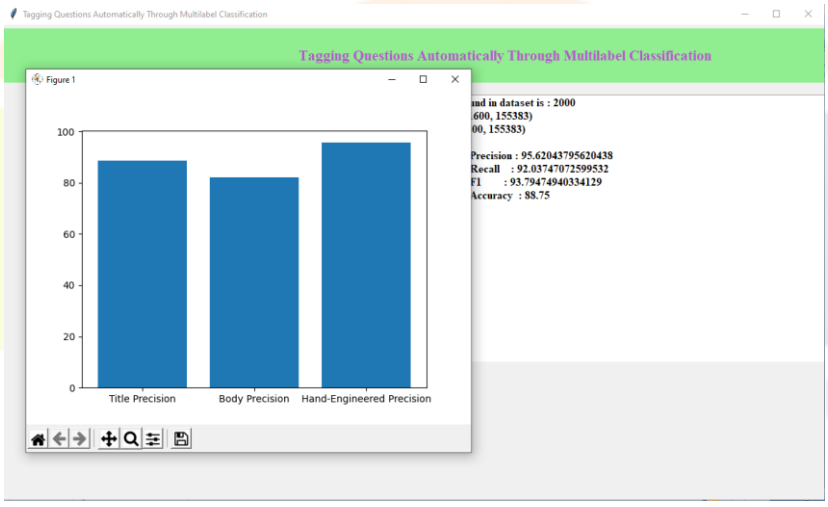
SAMPLE RESULTS

At initial stage after running the project, the first window that appears on the screen.

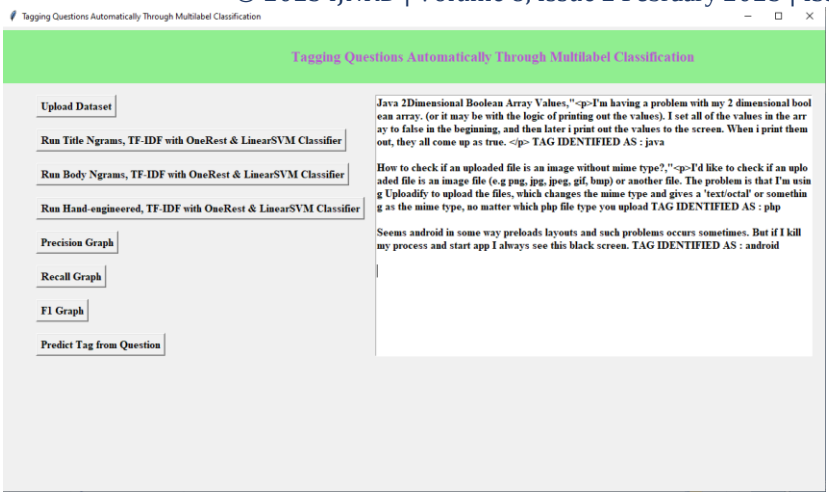
After this we need to upload the dataset for predicting the tag. Then by running the run title Ngrams TF-IDF with one vs rest and linear SVM classifier, run body Ngrams TF-IDF with one vs rest and linear SVM classifier, run hand engineered TF-IDF with one vs rest and linear SVM classifier, we get precision, recall, F1 score and accuracy of the tags.



To visualize the precision, recall, F1 score we added buttons.



For predicting the tag for the question asked by the user we need to upload the file of the question. After uploading the file it will the associated tag.



CONCLUSION

We implement a one-vs-rest classification system to automatically tag online forum question topics. Our classification system, implemented using a linear SVM model and carefully selected features, achieves an optimal F1 score of 62.35% for a sampled portion of the dataset with 100 of the most popular tags and a training set consisting of at least 500 examples of each tag.

REFERENCES

[1] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features.

[2] Kaggle (2013). Facebook recruiting III – keyword extraction. https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction

[3] Klassen, M. and Paturi, N. (2010). Web document classification by keywords using random forests. In Networked Digital Technologies, volume 88 of Communications in Computer and Information Science, pages 256-261. Springer Berlin Heidelberg.

[4] Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In Proceedings of the ACL-02 Workshop on Effective Tools and methodologies for Teaching Natural Language Processing and Computational Linguistics, ETMTNLP '02, pages 63-70, Stroudsburg, PA, USA. Association for Computational Linguistics.

[5] Manning, C. D., Raghavan, P., and Sch¨utze, H. (2008). Introduction to Information Retrieval Cambridge University Press, New York, NY, USA.

[6] McCallum, A. K. (1999). Multi-Label text classification with a mixture model trained by em. In AAAI 99 Workshop on Text Learning.

[7] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013) . Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.

[8] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011).Scikitlearn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830.