Large Scale Hierarchical Text Classification

Y Kushal , Massand Sagar Sunil

IIT Kanpur

Introduction

Today, there is a huge dependence on online sources of informations such as Wikipedia, Quora etc. However, the assignment of categories to documents on Wikipedia, as well as the assignment of question tags on Quora remains a largely manual process. We make an attempt at automation of this process.

Previous work

Previous works include top-down classification approaches which perform classification on each level of the hierarchy. However, these methods are inaccurate due to error propagation at each level of the hierarchy.

Problem Instance

The following are the salient features of the instance:

- **Prior knowledge of the categories which could be assigned as well as the hierarchy.**
- **Structure of the hierarchy is a DAG and the target categories are those nodes which do not have an outgoing edge.**
- **The document is given as a feature vector where the features represent words.i.e. a bag-of-words model is followed.**

Objectives

- **Efficiency: Due to the large number of training(2.2 million) and test documents(0.45 million), it is imperative that the classifier be efficient.**
- **Accuracy**

Practical application

A user posts a question about a topic or writes an article about it on Wikipedia. Our idea is to have automatic "tag suggester" for the document, which gives the user a pool of tags to choose from.

Overview of the algorithm[1]

We use a modified version of the enhanced KNN method given by [1] in our implementation. The algorithm is as follows:

- **Selection of top k documents: Find nearest k neighbors using a BM25 similarity measure for top 3 features of test document.**

$$BM25(S_a, S_e)=\sum_{k=1}^{n} f'(w,Sa) * f'(w,Se)*idf(w)$$

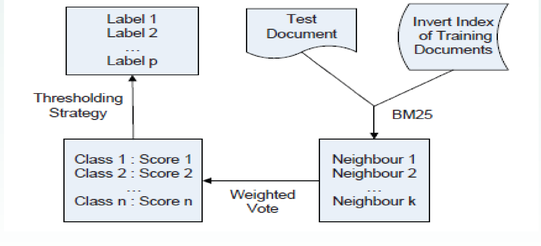
$$f'(w,s)=(k1+1)f(w,s)/(k1+1-b+(b*|s|/La))$$

$$Idf(w)=log(N-n(w)+0.5)/(n(w)+0.5)$$

- **Weighted voting: The top-k documents perform a weighted voting and a score is assigned to each of the labels.**

$$Score(c|Sa)=\sum_{k=0}^{n} \Upsilon(Se,c) * BM25(Sa,Se)\text{^}\alpha$$

- **Thresholding strategy: We employ a DS-cut[1] strategy in which we tune the thresholds for the 2nd label, 3rd label and so on and output a label if it crosses the threshold.**



Observations

For the enhanced KNN, we sampled the test document as well as the validation document from the training set[2] given. We ensured that no category was over-represented.

The parameter values we finally use for processing test documents are-

K=50, α=0.75

The maximum number of labels we accept is 3. The threshold parameters for the 2nd and 3rd labels are 0.6 and 0.4 respectively.

Computational efficiency: For each feature, we store the set of all the training documents containing the feature - (Inverse Document index).

Time taken per test document on average - ~0.276 s

Time taken per test document on average before using the Inverse Document index - ~1.5 s

Results

Using cosine similarity, we obtain an accuracy of 18.88%.

References

[1]-Enhanced K-nearest neighbor for Large Scale Hierarchical Multi-Label Classification, Xiao-Lin Wang, Hai Zhao and Bao-Liang Lu.

[2]-www.kaggle.com/c/lshtc