SEARCH ENGINE TECHNOLOGY

Project:
Topic Based Text Multi-Class Classification for News Articles

Type of Project: Software

UNI nr2483

ABSTRACT

Classification of news articles into categories can be used to develop a content based recommendation application to quickly pull out and suggest relevant news articles that a reader might be interested in.The project’s focus is to develop a classifier based on the supervised learning technique K-Nearest Neighbors to classify news articles into a set of 7 pre-defined and mutually exclusive categories: Sport, Business, U.S., Health, Science & Tech, World and Entertainment
A training set and testing set will be randomly selected from the following annotated Corpus
”News” corpus downloaded from following url
http://acube.di.unipi.it/tmn-dataset/
This contains a list of 32K English news articles extracted from popular newspaper websites such as nyt.com, usatoday.com and reuters.com and have been labelled by human experts.
The model will be trained using the training set.Since the system uses k-nearest neighbors technique it is easy to train, as it is memory based and does not require estimation of any parameters like priors for Naive Bayes and centroids for Rocchio classification.
For evaluation of the system’s performance, accuracy is used as a metric. The accuracy can be calculated as a ratio of correctly classified articles from the test set.

Motivation/Goals:

Classification of news articles into categories can be used to develop a content based recommendation application to quickly pull out and suggest relevant news articles that a reader might be interested in. The project's focus is to develop a classifier based on the supervised learning technique K-Nearest Neighbors to classify news articles into a set of 7 pre-defined and mutually exclusive categories: Sport, Business, U.S., Health, Science & Tech, World and Entertainment. The system also aims to compare it with other popular classifiers which implement machine learning techniques like SVM and Stochastic Gradient Descent.

Unlike commonly used classifiers which use the entire text of an article to learn the model, this system uses only a 2 line summary and yet manages to achieve a respectable accuracy of above 70%. Since it uses only 2 lines of an article it is much faster and training upto 12000 news articles requires only ___ seconds. The possible loss on accuracy is negligible in comparison to the time saved on processing and is a good tradeoff. Due to this lightweight design the future scope for this system includes integrating it on a mobile platform to build or improve mobile apps such as Yahoo News which can classify news articles on the fly.

Data:

A training set and testing set is randomly selected from the following annotated Corpus
"News" corpus downloaded from following url
http://acube.di.unipi.it/tmn-dataset/
This contains a list of 32K English news articles extracted from popular newspaper websites such as nyt.com, usatoday.com and reuters.com and have been labelled by human experts.
Example of one Data Sample:

investing: can you profit in agricultural commodities?
bad weather is one factor behind soaring food prices. can you make hay with farm stocks? possibly: but be prepared to harvest gains on a moment's ...
http://rssfeeds.usatoday.com/~r/usatodaycommoney-
topstories/~3/qbhb22sut9y/2011-05-19-can-you-   make-gains-in- grains_n.htm
1
20 May 2011 15:13:57
ut
business

Preprocessing

The corpus contains various fields such as the Title, Description, URL,Serial Number,Date,Source and Category. The data is contained in a .txt file which is line separated. The system first provides a script for data extraction which takes the input path of the corpus from the user, extracts relevant data and stores it in a CSV.

Console Output for Extraction step:

```
Please enter the Path of the Corpus
news
The corpus contains   32604  articles

Processing the Corpus to extract relevant data from the Corpus and
writing to CSV...

The training data has been loaded into corpus.csv

The time taken to extract data was: 4.91216897964  seconds
```

Tools and Libraries:
The system makes use of the following tools and libraries:
1) Scikit learn Library provides a number of useful library methods for various Machine Learning Techniques using Python

http://scikit-learn.org/stable/install.html

2) Numpy is used to handle multi-dimensional arrays,to store sparse matrices and other useful high level mathematical functions such as permutation to generate a random permutation of numbers in order to randomly split the data into a training and testing set for cross-validation

System Description: (Include formulas and references)
The model will be trained using the training set. Since the system uses k-nearest neighbors technique it is easy to train, as it is memory based and does not require estimation of any parameters like priors for Naive Bayes and centroids for Rocchio classification.

The system takes the text of the news article, generates the TF-IDF matrix and passes this to the classifier.
Most classifiers available in Scikit are binary classifiers.
To extend the binary classifiers for multi-class classification, this system uses a label Binarizer.
With a label binarizer, while learning the system can make use of just one binary classifier for each class “belongs to “ or “does not belong” to convert multi-class
This when combined with transform and inverse_transform methods can assign a single class during prediction per data sample point based on the corresponding model that gives the highest confidence.

The classifier methods return a list of classified output labels for the test data inputs.

Running Time:
Owing to the lightweight design, the entire classification system takes around 25 seconds to perform training and classification operations of a corpus containg more than 32000 news articles

Outline of Experiments:

A comparison is made between KNN classification technique and Linear SVC and Stochastic Gradient Descent for various sizes of randomly selected training datasets. The observations have been summarized in the below table:

Comparison for Accuracy with other Techniques

green=highest accuracy, blue=intermmediate accuracy, red= minimum accuracy

Number	Description	KNN Classifier	SVM Classifier	SGDC Classifier
1	train_ex=1000	53.87%	48.82%	55.54%
2	train_ex=2000	60.10%	58.24%	63.74%
3	train_ex=3000	64.19%	62.20%	66.18%
4	train_ex=4000	66.33%	65.78%	67.72%
5	train_ex=5000	66.80%	67.07%	69.69%
6	train_ex=6000	68.38%	68.53%	69.74%
7	train_ex=7000	68.54%	69.47%	69.93%
8	train_ex=8000	69.61%	70.13%	69.67%
9	train_ex=9000	71.02%	71.59%	70.47%
10	train_ex=10000	70.69%	71.96%	70.29%
11	train_ex=11000	71.73%	73.07%	70.64%
12	train_ex=12000	72.45%	73.68%	71.05%

The above comparison shows that across various training sample sizes, the KNN technique shows medium accuracy when compared to other popular classification techniques such as Linear SVM's or Stochastic Gradient Descent Classifiers. Here number of neighbors k=10 is chosen for the above experiments

Challenges Overcome:

- Finding an annotated corpus of reasonable size along with short summaries of news articles.
- Exploring and analyzing various Machine Learning Techniques and tools to be able to perform the experiments efficiently.
- Design the system and write code in a way that is easy to understand for developers who wish to enhance the system
- Initially the target was to develop a basic KNN classifier. But the implemented project also provides a comparison with other classification techniques along with a confusion matrix in the output for evaluation which were not part of the proposed system.

Evaluation:

The system uses a supervised learning technique.

First the given corpus of 32000 news articles is shuffled and a training set of size specified by the end user is randomly selected to train the model. Predictions are made for the remaining part of the corpus based on the learnt model. The standard cross-validation technique is used for evaluation of the system's performance. Accuracy is used as a metric which can be calculated as a ratio of correctly classified articles from the test set

Along with accuracy, the System also displays the Confusion Matrix which makes it easy for the end user to evaluate the performance of the classifier in a neat summary. Following is a sample output from one run:

Sample Run:

```
Initializing....
Please enter the number of examples that should be used to train the
model
5000
There are  32604  examples in the corpus


Running K Nearest Neighbors Classification
Number of Examples used for Training 5000
Number of Correctly classified 18479
Total number of samples classified in Test data 27604
The resulting accuracy using KNN is  66.9431966382 %

The confusion matrix is [[4000   45   39  138   80   71  157]
[1112 1357   15   18  178   56   42]
[ 710   23  638   11   34   61   77]
[1351   36   22  823   72   72   83]
[ 825   34    8   14 5962   61   33]
[1686   42   49   41  135 1809  302]
[1084   48   12   16   60  172 3890]]

Running SVM Classification
Number of Examples used for Training 5000
Number of Correctly classified 18474
Total number of samples classified in Test data 27604
The resulting accuracy using Linear SVC is  66.9250833213 %

The confusion matrix is [[4141   34   20  121   23   97   94]
[1257 1396    4   10   74   19   18]
[ 786   15  593    7   17   75   61]
[1421   29   22  824   46   68   49]
[1005   20    2    4 5864   26   16]
[1896   15   28   25   41 1904  155]
[1326   27   15   15   12  135 3752]]

Running Stochastic Gradient Descent Classification
Number of Examples used for Training 5000
Number of Correctly classified 19055
Total number of samples classified in Test data 27604
The resulting accuracy using Stochastic Gradient Descent is
69.0298507463   %

The confusion matrix is
[[4076   50   23  140   31  111   99]
 [ 969 1648    5   11   87   36   22]
 [ 736   26  618    6   24   82   62]
 [1310   48   21  898   52   83   47]
 [ 832   39    3    5 5996   44   18]
 [1642   38   31   29   68 2089  167]
 [1247   40   15   18   34  198 3730]]
```

References:

http://scikit-learn.org/stable/
http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
http://en.wikipedia.org/wiki/Confusion_matrix
http://acube.di.unipi.it/tmn-dataset/