COMP90049 Project 2 Report: Geolocation identification of tweets

Introduction

A GEOLOCATION identifier refers to finding out the location from which a given tweet originates. The aim of the developed system is to identify the location of a tweet from a large dataset of tweets. Inorder to make this prediction of location, it is important to identify the factors or features unique to a tweet from a given location. The system attempts at building a model using a classifier to make this predictions for the tweet's location after the generation and subsequent analysis of the required features. This particular paper aims at studying the various systems that were developed as part of the project with different features and classifiers and comparing the models yielded by each of these systems. The end goal of the comparison is to gain an insightful knowledge about which classifiers best fit location identification and to understand the role of proper choice of features and the impact of feature engineering in the performance of the given model.

Literature Review

As part of their study Einstein et al. (2010) identified that the geolocation associated with every tweet can be identified based on lexical analysis[1]. The model proposed uses classifier to develop multiclass model that fits tweet location prediction data better than simple linear regression based models.

Further Rahimi et al. (2018) developed their location identification model using graph Convolutional Network which supersedes the performance of supervised learning models, but is computationally taxing if the dataset has clear lexically rich features[2]. Thus the system this paper is going to discuss is taking into account the use of text classification based models proposed by the former while also using highly supervised condition based modelling as inferred in the later paper.

Data Input and Preprocessing

A. Preliminary Input

The initial input consists of two main files. A text file each for training and testing the developed model. The training dataset provided to the prediction system is a text file consisting of tweet's id, the user's unique identification, the tweet itself and the location from which the given tweet originated. The test file consists of all the above mentioned features. However every value of location corresponding to a tweet is filled in with a question mark (?). In order to develop a model using different classifiers, the given input data had to undergo the following data pre-processing.

B. Feature Selection

Having examined a batch of tweets from the training dataset, some tweets stood out with their use of emoticons, special characters, etc. Hence features maintaining the count of these characters were added to the dataframe. Furthermore every tweet was analysed to identify if the sentiment behind a given tweet is good or bad. Inorder to perform this analysis, the Vader package in python is used. A parameter called sentiment compound is an output of the sentiment analyser which goes from 0 to 1 for most negative to most positive statements. Each of the tweets need to be analysed in the context of the location which can be determined by some special bag of words. To do this the tweets were processed by removing the punctuation and the stop words (Common words).Lastly terms from each tweet where obtained to generate a term vector and each of these vectors where filled with binary values depending on the presence (1) or absence (0) of the term in a given tweet.

C. Final Input

Post the above mentioned pre-processing, the final training dataset is generated with the following columns. Training data < 'tweet-id', 'user-id',

'tweet', 'tweet-puc-less', '@count', '#count', '!count', 'word-count', 'sentiment', '<term frequency>', 'location' >. The testing data and the training data files have same number of features but the shape of both files is different.

METHODOLOGIES

The system needs to predict the location of any given tweet given that a dataset of tweets and their corresponding location is available. Herein a prediction of the value of Y i.e the location is to be made given the value of X i.e the tweets. For this it is important to identify how X and Y are related mathematically. Thus it was understood that it is a classification problem and different classification algorithms were explored. Since the class feature is deterministic, various supervised algorithms were tested. The system was tested against the following supervised algorithms.

A. Decision Tree Algorithm

A decision tree algorithm assigns every single viable option in the class feature a probability corresponding to how likely it is to be the final result[3]. Decision tree algorithms are simple if-else structures bound between probabilistic assignments. The following is an example of a simple decision tree structure.

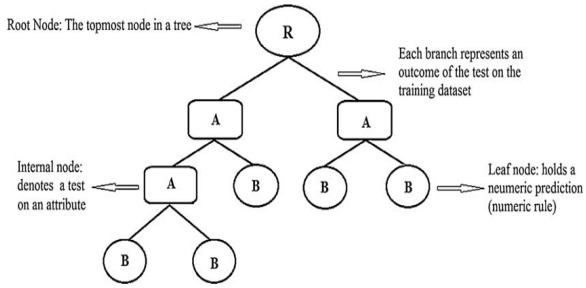

Fig [1]

B. Naive Bayes Algorithm

Naive Bayes determines the best class a particular tuple of data belongs to by determining the conditional probability of the selected feature to its class feature[4]. Simply put Naive Bayes gives the best choice given the right features to the classifier to model. The structure of Naive Bayes is as follows:

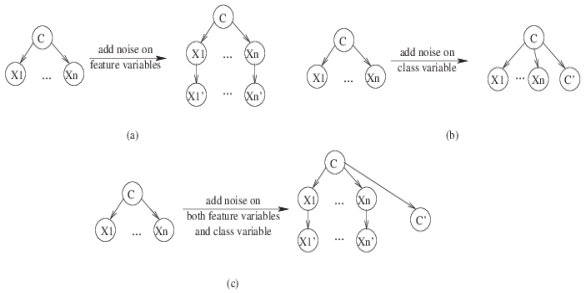

Fig [2]

C. K nearest neighbour Algorithm

A K nearest neighbour classifier or a memory based classification works with the principle that objects of data that are close to each other in similarity are to be classified into the same class[5]. So the algorithm analyses the different features, performs calculation as to which tuple is closest to the unknown sample and classifies it accordingly.

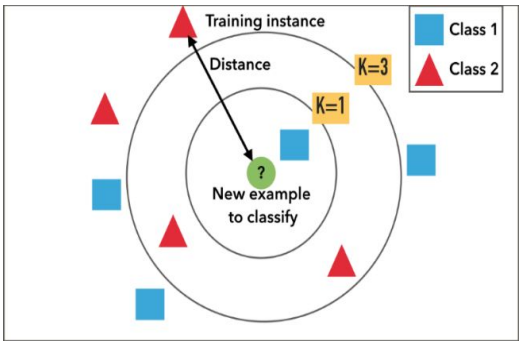

Fig [3]

PROPOSED SYSTEM

A. Business Logic

The system uses the above mentioned supervised learning algorithms to predict the location from which a tweet originated. In order to do that, a choice of important features in the dataset had to be made. The final training dataset was devoid of features such as user-id, original tweet and tweet-id. The training test was vertically divided as X and Y wherein X represents the various features obtained by processing the tweet and Y held the location class. Further the system was divided horizontally in a 70-30 ratio to act as a training and testing set. Division of this manner helps understand how well the developed model is given that we know the locations of the test set as well. Post this split the training data is passed to above mentioned classifiers and a mathematical relation that 'fits' between X and Y is obtained. This fitting is what makes the model. Once the model is

created, the X data of the test set is passed to it to predict the value of Y. once predicted Y array is obtained its compared with original Y of the test set to identify how accurate the created model is.

B. System Implementation

The system is developed using Python as the main programming language. An input training data frame of length 96585 is passed to the classifier. The final test data frame with unknown location was of the length 32977. All the classifiers are imported from the python package scikit-learn. The decision tree classifier is imported as

from sklearn.tree import DecisionTreeClassifier

The Naive Bayes Classifier is imported as

from sklearn.naive_bayes import GaussianNB

And the K nearest neighbour classifier is imported as

from sklearn.neighbors import KNeighborsClassifier

Inorder to measure the correctness of the system various measuring metrics were required which were imported as

from sklearn import metrics

COMPARATIVE ANALYSIS

Each of the above classifiers are compared for the accuracy score and the values are as follows wherein DT is Decision Tree, NB is Naive Bayes and KNN is K nearest neighbours:

Algorithms	**DT**	**NB**	**KNN**
Values	0.6054	0.6001	0.59

Table 2.1

Accuracy is a measure of the ratio of correct answers to total words of input. Meaning it interprets how many values have been classified correctly. The mathematical formula is as follows:

As observed from the above table, the decision tree algorithm performs better than the other two algorithms. This has been observed because the decision tree by default assigns feature importance to all the features provided in a dataset and works using the assumption that most important contributes most towards the relation between X and Y and hence must be used first to classify. The Naive Bayes has lesser accuracy than decision tree because of improper selection of features as it cannot directly handle all the features assigned inside the dataset. K nearest neighbours perform poorly to the decision trees due to improper choice of number of neighbours.

UPDATED SYSTEM

In order to work on the above mentioned aspects and possibly improve the efficiency, an updated system is designed

A. Business Logic

This system is an improvisation of the proposed system. In this system a proper feature selection and subsequent feature engineering is performed. The YAKE package in python is used to find the best terms and their corresponding importance in any given tweet. Furthermore by performing feature importance analysis it was observed that feature 'word-count' actually decremented the accuracy of the classifier and hence the feature is dropped. The feature engineering is performed by taking up probabilistic values of fitting from each classifier and getting the arithmetic mean to find out the final class.

B. System Implementation

The updated system is also implemented using Python. It makes use of the Voting classifier to combine the results of the above three classifiers. The Voting classifier is imported as:

from sklearn.ensemble import VotingClassifier

COMPARATIVE ANALYSIS

Each of the above classifiers are compared for the accuracy score and the values are as follows wherein the acronyms are as former table with VC for Voting Classifier:

Algorithms	**DT**	**NB**	**KNN**	**VC**
Values	0.6054	0.61	0.63	0.6117

Table 2.2

From the above table it is observed that the accuracy value of the k nearest neighbour algorithm is relatively higher than the other classifiers.

Proposed system and Updated System

From tables 2.1 and 2.2 it is clear that the updated system is an improvement on the originally proposed system. An increment is observed in the accuracy of the classifiers namely Naive Bayes and K nearest neighbours. This difference is due to appropriate feature selection. The voting classifier does not significantly change the predicted values as observed by the accuracy metric. The K nearest neighbour algorithm performs better because of proper choice of features and proper choice of number of neighbours. The right number of neighbours is determined by the elbow method. The number which is parallel to the value of X is the ideal number of neighbours. However a point worthy of noting is that although according to above observations K nearest neighbour looks like the best option, it is imperative to note that with the increase in feature and dataset shape, the number of neighbours may increase making it computationally expensive. Thus to infer from the updated system, it would be viable to use Naive Bayes for both lesser computational complexity and high accuracy metric. An important thing to note is that this improvement in the system occurs due to proper feature selection and subsequent engineering. Thus the knowledge obtained from the above analysis is to design a model by identifying the best classifier for the problem statement which in case of geolocation identification is found out by the above system to be Naive Bayes and to make the right choice of features to better fit the model.

Future Scope

The system did not perform a comparative analysis between all available classifiers. A good addition would be to see the model developed by Support Vector Machine using the linear kernel as it performs well in the IRIS dataset which is similar in structure to the geolocation identification dataset. Also a good way of engineering features would be to divide the training set into batches and perform stratified K-fold analysis. The system is currently being updated to incorporate the above mentioned changes. Thus in conclusion the study performs a comparative analysis and identifies best classifier to fit a model for geolocation identification.

References

[1] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In Proceedings of the 2010 conference on empirical methods in natural language processing, pages 1277– 1287. Association for Computational Linguis- tics, 2010.

[2] A. Rahimi, T. Cohn, and T. Baldwin. Semi-supervised user geolocation via graph convolutional networks. arXiv preprint arXiv:1804.08049, 2018.

[3] Magerman, David M. "Statistical decision-tree models for parsing." *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1995

[4] Rish, Irina. "An empirical study of the naive Bayes classifier." *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. No. 22. 2001.

[5] Cunningham, Padraig, and Sarah Jane Delany. "k-Nearest neighbour classifiers." *Multiple Classifier Systems* 34.8 (2007): 1-17.

[6] Retrieved from https://images.app.goo.gl/z7EmJXeirKDvqfdg9

[7] Retrieved from https://images.app.goo.gl/nYwduzaP2m1wWeTD9

[8] Retrieved from https://images.app.goo.gl/BFjWKWM6rGfE9X7S9