Deanonymizing Quora Answers

CS299 Project Report

By

Arunika Yadav - 1601CS56
Harshika - 1601CS14

Submitted on

17th April, 2018

Department of Computer Science and Engineering

Indian Institute of Technology , Patna

Acknowledgement

We would like to extend our gratitude to Dr.Jimson Mathew, Dr. Arijit Mondal and Dr. Raju Halder for having given us an opportunity to explore in depth of certain topics related to Machine Learning and Natural Language processing in particular. It has been a great chance to explore deeper into the course work and understand and apply it better.

Contents

1 Abstract

Quora is a knowledge sharing and gaining website but it does not have a system to keep a track of the answers which are uploaded anonymously.With this feature lacking in quora, this extremely useful platform can prove to be a potential danger making it's way into the world of cyber crime like plagiarism or even exposing malcontent against somebody, to say.Also it is a prequisite to use quora that everybody should have an account on the site and hence we can solve the problem of deanonymizing quora answers using our machine learning models.

2 Introduction

Authorship identification has been a very important and practical problem in Natural Language Processing. It has become even more important with the widespread use of various forums where anonymity is allowed. A good system for authorship identification can handle cases of misuse of anonymity. Such a system can also be used for plagiarism detection. Analysis of authorship of historical texts has already been done many times using techniques for authorship attribution. Various spoofing e-mails can also be detected by using these techniques.We are trying to apply our knowledge of authorship attribution to our project **Deanonymizing Quora Answers** .

3 Previous Work

The problem of authorship identification has been studied extensively. Many hand-designed features have been developed in order to tackle this problem. For solving this problem, conventional text classifiers like naive bayes, logistic regression and SVM have been used, and various deep learning models like LSTM have also been tried. Different models perform differently depending upon the type of training and testing data available. **We have tried to map the problem that we are solving with this already existing model.**

4 Dataset

We have obtained the necessary dataset of size **20MB** from kaggle.com. It contains **50 training documents and 50 test documents** for each of the 50 authors. All of these documents are on the same topic. Every document has been written by one and only one author.We have merged both the datasets into one single directory and then went on to split them into a ratio of 8:2 as training dataset and test dataset, respectively.

5 Our Model

It is a supervised learning task as a label is present for each of the training examples. Since there are multiple authors and we have to assign the document just one of them, it is also a multi-class single label text categorization problem.
We basically used the following models for accomplishing our tasks: naive bayes classifier, support vector machine classifier, decision classifier, random forest classifier, logistic regression classifier, stylometric features or style markers and convolutional neural networks.
Similar ideas have been tried out in the past but not necessarily in this particular combination. We have used Python programming language for implementing our model. We have used logistic regression classifier from the Python SciKit Learn library.We have also used keras library for implementing neural networks.

5.1 Naive Bayes Classifier

Naive Bayes classifier calculates the probabilities for every factor.Then it selects the outcome with highest probability.This classifier assumes the features (in this case we had words as input) are independent. Hence the word naive.

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- P(c || x) is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P(x||c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

A Naive Bayes Classifier is a program which predicts a class value given a set of set of attributes. For each known class value,

- Calculate probabilities for each attribute, conditional on the class value.
- Use the product rule to obtain a joint conditional probability for the attributes.
- Use Bayes rule to derive conditional probabilities for the class variable.

Once this has been done for all class values, output the class with the highest probability.

5.2 Support Vector Machine Classifier

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems.It is based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. For this type of SVM, training involves the minimization of the error function:

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

subject to the constraints:

$$y_i \left(w^T \phi(x_i) + b \right) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, \, i = 1, \ldots, N$$

where C is the capacity constant, w is the vector of coefficients, b is a constant, and represents parameters for handling nonseparable data (inputs). The index i labels the N training cases. Note that represents the class labels and xi represents the independent variables. The kernel ϕ is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized. Thus, C should be chosen with care to avoid over fitting.

We applied SVM to our training set and obtained better results than naive bayes classifier.Also we used the scikit-learn library to apply the same classifier.

5.3 Decision Classifier

Decision Tree Classifier, repetitively divides the working area(plot) into sub part by identifying lines.The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data(training data).The understanding level of Decision Trees algorithm is so easy compared with other classification algorithms. The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

We implemented this classifier to train our dataset but obtained an accuracy which was slightly less than svm.

5.4 Random Forest Classifier

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve

the predictive accuracy and control over-fitting.The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).It is an ensemble type of model, i.e, it works by implementing decision tree classifier model many times.
Even Random Forest Classifier could not give us better accuracy.

5.5 Logistic Regression

Logistic regression is named for the function used at the core of the method, the logistic function.
The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

$$1/(1 + exp(-value)) \tag{1}$$

Where e is the base of the natural logarithms and value is the actual numerical value that you want to transform.
We used the NLTK library and the scikit learn library to implement the same and obtained the best accuracy so far.

5.6 Stylometric Features

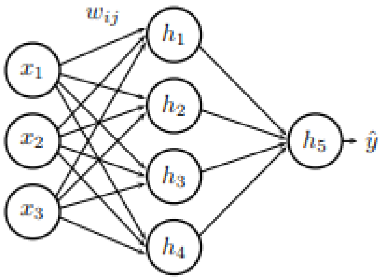
Stylometric features or style markers try to capture the writing style of the author. We have used a large number of stylometric features in our model. The motivation behind these features comes from the survey paper on authorship attribution [4].
The features pertaining to types and token are:number of types (unique words), number of tokens, type-token ratio. Some authors have a tendency to write long sentences whereas some authors prefer shorter sentences.In order to capture this, we have included various features in our model such as average length of a sentence (both in terms of words and characters), standard deviation of sentence lengths. Some authors pose more questions as compared to others. For assimilating such ideas in our model we have introduced features like relative number of declarative,interrogative, imperative and exclamatory sentences. Other style markers include relative number of punctuations,

relative number of digits in the document etc. Natural Language Tool Kit (NLTK) has been used to extract these style markers from the text.

5.7 CNN

A neural network is usually made of several units, also known as neurons, of the form :

$$h_j(\mathbf{x}) = \sigma(w_j + \sum_{i=1}^{n} w_{ij} x_i),$$



The first layer is the input layer, which transmits the input values x = (x1, ..., xp) to the second layer. The second layer is made of activation units hj, taking as inputs the weighted values of the input layer and producing non-linear transformations as outputs. The third layer is made of a single activation unit, taking as inputs the weighted outputs of the second layer and producing the predicted value $\hat{y}$. Assuming that this structure is fixed and that all units from the network make use of the same activation function σ, the hypothesis space H therefore includes all models ϕ of the form :

$$\varphi(\mathbf{x}) = \sigma(w_5 + \sum_{j=1}^{4} w_{j5}\sigma(w_j + \sum_{i=1}^{p} w_{ij} x_i)).$$

In convolutional neural networks the first layer applied to the inputs is of convolution function followed by some other approximation functions.

6 Experimentation and Results

6.1 Main Models used

The implementation was done in Python using the com- monly used ML and NLP libraries like scikit learn, Keras, NLTK etc. The code is available in this github repository.
We started out with the simple naive bayes classifier which gave a decent accuracy. Then we tried various other classifier models of which logistic regression performed the best for us. So we decided to go ahead with it. Then we added various stylometric features to our model and obtained a best accuracy of **84 percent**.

6.2 Other Methods Explored

CNN :
We also tried CNN as they have been proved to be very effective for this problem in the literature. Implementation of CNN was done in keras. The initial accuracies were not good. One problem that we suffered from was the lack of data. We had only 20 MBs of text, used for training and testing of 50 classes. Since neural networks require a large amount of data for getting appreciable results, they did not help in effectively solving our problem. Therefore, we did not further consider the approach of using CNN for solving our problem.We tried to go for the larger dataset of size 1 GB but because it was not labelled, we could not proceed with the same as labelling it would have taken a lot of time.

Lemmatization and POS tagging :
We considered replacing the words with their root forms using Porter Stemmer lemmatization [5]. But it did not result in an increase in accuracy. We also tried the syntactic feature of appending the words with their Part-Of-Speech (POS) tags.Again this idea did not work out in our favour. These features try to hide the actual word information. Perhaps due to this they did not improve the accuracy.

7 Conclusion

The accuracy obtained by our model is close to the accuracy of the baseline model. This shows that stylometric features are crucial to the task of identifying the anonymous authors. Also for the used dataset, traditional classifiers like naive Bayes, SVM and logistic regres- sion are performing better than neural network models like CNN.

8 Future Work

One promising approach is to use CNN with more amount of data and to build on the previous work[1][2] added with all the stylometric features.
The classifiers that we used were linear in nature. Therefore, it seems that the use of non-linear models can increase accuracy as non-linear models will also be able to capture non-linear properties of the documents. Adding semantic features like synonyms and antonyms can also be tried.
Also we have our dataset in the form of text files rather than .csv file, hence we faced a lot of problem in the initial steps of preprocessing our data.Also we could not build a platform because of the same glitches.Hence if the dataset is available in the .csv file, a lot can be done easily.

References

1. Anand Pandey and Ankit Pensia. Cs671: Natural lan- guage processing.

2. Pranav Jindal and Ashwin Paranjape. Deanonymizing quora answers.

3. Efstathios Stamatatos. A survey of modern authorship attribution methods.

4. Vivake Gupta (v@nano.com).porter stemmer.https://github.com/vbuterin/spread/blob/master/spread/porter.py, 2012.