Age Detection from Blog Data

Sanket Kashyap[1]

[1]Department of Computer Science, Ashoka University

Abstract

The Blog Authorship corpus is the set of blogs available on blogger.com in 2004. In this paper, we try to predict the age category of the blog author depending on the textual content and writing style of the author. We first extract features at varying granularity (sentence, word, character) to create a feature set which is then used for classification. The size of the feature set is 88 elements which are a combination of the above discussed features. We observe multiple classifiers (SVM, Random Forest, XGBoost etc.) and finally see that the Multi-Layer Perceptron classifier performs the best, giving an accuracy of 72.83%.

1 Introduction

The internet has connected us all together to create a smaller world where people are always connected. Most websites do not require a formal identification for creating an account and work with the concept of anonymous handles. Anonymity has also given rise to online bullying and the creation of fake accounts to lead massive campaigns without any accountability. In this project, we create a predictive system which can predict the age-group of a blog author by analysing carefully selected textual features from the contents of their blog. The corpus used for training the model is the Blog Authorship corpus which has the data of 19,320 bloggers gathered from blogger.com in August 2004. The variety of content present makes the corpus different from datasets which have contextually similar data like the IMDB Movie reviews dataset. For the task of classification we compare individual classifiers, an ensemble classifier based on maximum voting and a Multi-layer Perceptron classifier.

2 Background

The work is an extension of the work done by Schler and Koppel (2006)[1], where they use a Multi-Class Real Winnow (MCRW) algorithm on 502 features extracted on the style of writing achieving an overall accuracy of 76.2%. Rosenthal and McKeown (2011)[2] use the same dataset as part of their larger dataset to predict age and gender from blogs combined with the online behavior of the users yielding an accuracy of 81.57%. Considering the similarity of neural networks and MCRW, we hypothesize that neural networks should also perform well on age detection.

3 Approach

3.1 Data

The Blog Authorship corpus has the post data for 19,320 bloggers, in an .xml file for each blogger. The data is first cleaned of non-unicode characters and each blogpost is saved as a row of the given dataframe, ['ID', 'Gender', 'Age', 'Industry', 'Sign', 'Date', 'Text']. The models are trained to predict the 'Age' parameter. The problem is posed as a multi-class classification problem not a regression problem where we predict the exact age of the blog author so the 'Age' variable is converted to a categorical label as shown -

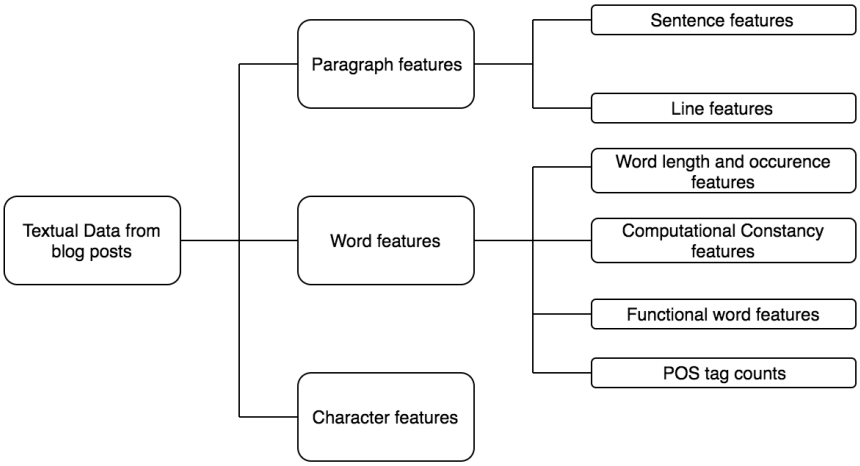

Figure 1: Feature Generation Pipeline

- For ages between 10-20, the label is 10
- For ages between 20-30, the label is 20
- For ages between 30-40, the label is 30

So, we obtain a total of 679319 rows with each row corresponding to a single blogpost.

3.2 Feature Extraction

For the amount of text data present it was inefficient to use sparse representations (eg. n-grams, tf-idf vectors). Instead, to obtain features we parse the text at a different granularity level and obtain stylistic features at a paragraph level, word level and character level.

In all, we compute 88 different features from the above pipeline. The sentence and line level features are counts of words, average word length, average number of punctuation used in a line etc. The word level features capture the functional words groups (Conjunctive words, auxiliary verbs etc.), length of words, number of large words, number of small words, computational constancy measures etc. A constancy measure for a natural language text is defined, in this article, as a computational measure that converges to a value for a certain amount of text and remains invariant for any larger size. Because such a measure exhibits the same value for any size of text larger than a certain amount, its value could be considered as a text characteristic[3, 4]. The constancy measures used are -

- Yule's K measure
- Entropy
- Simpsons' D measure
- Sichel's measure

The character level features account for the presence of upper-case charcters, lower case characters, whitespace charcters, special characters etc. These features are combined to create a feature dataframe which is then normalized for better performance during classification. The final feature vector for each blog entry contains 88 normalized features.

3.3 Classifiers Used

Finally to choose the optimal classifier we try 6 different classifiers, which are -

- Linear Support Vector Machines
- Random Forests
- Logistic Regression
- XGBoost Classifier
- An ensemble classifier that is a combination of Linear SVM, Random Forest and Logistic Regression with majority voting
- A multilayer perceptron

The first 5 classifiers are implemented using the scikit-learn library in python while the multilayer perceptron is implemented using the Keras[5] library.

4 Results

We create a test-train split of 70% - 30% and train the classifiers on the test set to see the performance of the classifier. The results are shown in the table provided below -

Classifier	Accuracy
Linear SVM	54.07%
Random Forest	58.27%
Logistic Regression	53.65%
Ensemble Model	54.22%
XGBoost	56.31%
MLP	**72.83%**

Table 1 : Comparing Classifier Accuracy

The best performing classifier is the multi-layer perceptron which has two hidden layers - one of 64 neurons and another of 20 neurons, finally leading to a 3 neuron output which gives the output as a one hot vector for the class that has been predicted. The structure can be seen in the following image -

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_13 (Dense)             (None, 64)                5696
_________________________________________________________________
dense_14 (Dense)             (None, 20)                1300
_________________________________________________________________
dense_15 (Dense)             (None, 3)                 63
=================================================================
Total params: 7,059
Trainable params: 7,059
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: Neural Network model

The MLP is trained for 250 epochs and is trained with a batch size of 2048 using the adam optimizer. For each classifier we construct the confusion matrix, which helps us analyse the performance better. Sparse features on a small sample of the dataset seem to perform very poorly and we choose not to include sparse features in our feature set as they get muted by the presence of the other features. Using an SVM classifier on the tf-idf vectors over the bigrams of the text creates makes the classifier predict the most frequently occuring class. We do not test this approach over the whole dataset because a computational bottleneck is observed while creating sparse vectors from the text corpus.
The confusion matrices over the testing results are shown in the tables below -

Real-Predicted	**10**	**20**	**30**
10	42258	27754	712
20	21829	71658	2703
30	5161	26799	4922

Table 2 : MLP Confusion Matrix

To compare the classifiers we can see that the MLP classifier has a much stronger concentration of elements on the main diagonal. The random forest classifer is next best performing classifier.

Real-Predicted	**10**	**20**	**30**
10	45142	23576	2006
20	23091	66099	7000
30	6769	22601	7512

Table 3 : Random Forest Confusion Matrix

Real-Predicted	**10**	**20**	**30**
10	31972	38724	28
20	21829	78481	71
30	5161	32474	51

Table 4 : Ensemble Model confusion matrix

The ensemble model despite using all three of its constituent classifiers fails to accurately predict the '30' class while the random forest classifier performs this prediction much better. This happens because of the maximum voting where the Linear SVM and Logistic Regression mute the effect of the correct answer given by the random forest. The [1] paper gives a baseline of 43.8% which is bettered by all our classification approaches. The MLP classifier delivers a result close to the Multi-Class Real Winnow (MCRW) algorithm in [1] despite using a much smaller feature set (88 features compared to 502 features).

References

[1] Jonathan Schler. Moshe Koppel. Shlomo Argamon. James Pennebaker. *Effects of Age and Gender on Blogging.* AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs,2006

[2] McKeown, Kathleen; Rosenthal, Sara *Age Prediction in Blogs: A Study of Style, Content, and Online Behavior in Pre- and Post-Social Media Generations* Association for Computational Linguistics, 2011

[3] Kumiko Tanaka-Ishii, Shunsuke Aihara
`Computational Constancy Measures of TextsYules K and Renyis Entropy` Association for Computational Linguistics, 2015

[4] Anke Ldeling, Merja Kyt
`Corpus Linguistics, Volume:2` De Gruyter, 2009

[5] Francois Chollet
`https://github.com/fchollet/keras,` 2015