Ziynet Nesibe
Computer Engineering Department,
Fatih University, Istanbul
e-mail: admin@ziynetnesibe.com

Author Prediction for Turkish Texts

Abstract

The main idea of authorship categorization is to specify characterization of documents that finds the writing type of authors. The features which we use to make correct predictions are including measurements like counting of word numbers and some specific styles used by authors. These features are used for predicting the author according to given text. In this experiment, 40 articles of 30 Turkish columnists have been analyzed. Results are found by using some machine learning algorithms. Naïve Bayes, Naïve Bayes Multinomial, Neural Networks, SMO, and other algorithms are the classifiers that have been used to classify texts according to their authors. The experiment will also group the given article according to gender and age. This experiment is implemented on WEKA data mining tool.

Keywords: Natural Language Processing, Author Prediction for Turkish, Authorship Categorization

Introduction

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics. NLP has a research area that is used for many different purposes and it is getting more popular day by day. Authorship categorization is an application of NLP.

Authorship categorization can be defined as identifying the author or determining some characteristic features of the author by using a given text with assist of natural language processing methods. Matching a given text and author according to attributes which are known from training texts is called author prediction. Author prediction is a sub-category of text categorization which is also a sub-category of document classification. For Turkish, studies about this issue are done very rarely.

Author identification can be used in a large range of applications. There exist rare authorship cases and applications about Turkish articles. In 2006, determining the identification of a Turkish document's author, classifying documents according to text's genre and identifying a gender of an author, automatically by using some classification methods which are Naive Bayes, Support Vector Machine, C 4.5 and Random Forest; and the success rate in determining the author of the text, genre of the text and gender of the author was calculated as 83%, 93% and 96%, respectively. [1] In 2007, the author of a text can be identified by using 35 style markers that characterize a group of authors which consists of 20 different writers with a success rate of 80% in average. [2] The other work is done for investigating the feasibility of predicting the gender of

a text document's author using linguistic evidence, and 84.2% success is achieved by illustrating the applicability of techniques to gender prediction. [3]

The rest of the paper is organized as follows. Methodology includes detailed description of the methods and techniques which are used in the project. Results and Discussions include the experimental results of the project which is defined and discussed. Summary includes the conclusions.

Methodology

For the authorship categorization system, there exists a program named as Prizma. I have updated some parts of this program, and use it for author predictions.

First of all, I have collected the data for training and testing. The data used in this project is articles gathered from different newspapers. Previous version of this project had 10 authors and made predictions just about who is the author of the given text. I increase the number of author, and attributes which are used for predictions. I have 30 authors to analyze. The genre of these authors is close to each other, they generally write about politics and casual events. I have totally 1200 articles, which means that 40 articles from each of 30 authors. Each data has specified format. Their extensions are txt, the first line of this file contains the title, and every line has a paragraph. The name of each data is the published date of article. Each data is inside a folder which is named as the author of the article.

After that, we have created data sets from these data. There are 3 different data sets. One of the data sets is author dataset which contains folders of 30 authors. These folders' names are author's name, and this each folder have 40 articles from the author. Second data set is gender dataset. There are 2 main folders inside, which are male and female. Male folder has author folders which author is female, and male has male authors' folders. The last data set is age dataset. There are 3 main folders inside. For under the age of 50, we have grouped authors as young. For the age between 50 and 65, we have grouped authors as middle-aged. For over the age of 50, we have put them into old group.

For getting prediction results, we need to create arff files with a specified format. Arff files are created by using Prizma program. User selects a dataset location, gives a name to arff file, and chooses the training percentage of whole data. There are attributes in a list. User chooses

the attributes that s/he wants to have in the arff file. The Prizma Program creates arff file with a given name and contains the chosen attributes.

If user chooses author dataset, the program selects training data of each author folder according to training percentage. After writing attribute values of training part of each article into arff file, it starts to alter testing values into the arff file. If the data set is selected as age dataset or gender dataset, It goes into the folder, it creates arff files according to age or gender. This structure works like that; it looks at the data set location, if there exists folder inside it, it takes folders one by one, and calculate them. If there exists just files inside the data set location, it takes the upper files with given percentage as training, and the rest of the folder is taken as testing.

In the coding part of the Prizma program, there are attributes which is used for creating arff files. There were some attributes which were written for old version of the program. I have added extra features to be able to make better predictions. Because, the program was predicting the author of the given text between 10 authors before, but now, I try to predict also age group and gender of the author of the given text. And also I have 30 authors anymore. When the number of possible output increases, it is more difficult to make correct predictions about the author and understand the writing style.

In Prizma program, there is a list of attributes. User needs to select at least one attribute to create an arff file. After taking data and creating arff files of them according to chosen attributes, it is time to test these data. We have used WEKA to see the results and compare them according to chosen algorithm by using these arff files. WEKA is a popular tool which is used for machine learning and it is free software available. WEKA is used to classify the given text. WEKA takes an arff file, allows the user to select an algorithm, and shows the user the correction rate of results. I have tried Naïve Bayes, Naïve Bayes Multinomial, Sequential Minimal Optimization, and Neural Networks.

There is some extra information like accuracy of the results and confusion matrix. Confusion matrix shows us every detail of data, how many of them is classified correctly and how many of them is wrong classified as what. Every details are shown in this confusion matrix. It is thought that to add this classification part to the Prizma program, but now it is under construction. If it would be available, the Prizma program will be enough to do all these coding, training and testing parts, and the program will use WEKA in the background, but the user will not waste time. According to these confusion matrix and accuracy results, we see which algorithm gives the best

result, and the features are enough or not to make correct prediction with expected result. It is possible to measure the rate of success and comment over the results.

Results and Discussions

Many attributes are analyzed and according to affect on the result they are eliminated or they are added to the program as attribute. At last we have 19 attributes to make the best predictions. These attributes are average paragraph length with/without space, average sentence length, empty paragraph ratio according to total paragraph number, length of title, paragraph count, some punctuation ratios such as asterisk, colon, comma, dash, double quote, elliğsis, exclamation, and semicolon, punctuation count of title, punctuation ratio of whole text, stopword ratio, subtitle ratio and word length variance. For most of the attributes, I have used ratio to determine the characteristics of the author. For the future work, this program would be able to determine the author of given text which is not always a complete article, but also some part of an article. However, for some attributes measures the count, because it is also a characteristic style, for example, writing too long articles, using long words, etc.

After training and testing part, we have percentage of correctly classified instances. We have analyzed our datasets with four different rates:

- 75% training, 25% testing
- 50% training, 50% testing
- 37.5% training, 62.5% testing
- 25% training, 75% testing

While predicting an author, to use the biggest part of data as training can be better. Because, if we have more training, we can know this author better. However, we have tried to use the smallest amount possible to predict the author. Our aim is to know the author with small amount of data, learn his/her characteristics with these data, and test lots of data to see the predictions are true or not. I have tried the datasets with some machine learning algorithms, and according to my structure of job, Naïve Bayes, Multilayer Perceptron, and SMO algorithms are suitable for my experiment. The randomly results are like that:

Author Dataset			
	NaiveBayes	Multilayer Perc.	SMO
30x10	85.33	81	82.67
20x20	83.83	83.33	82.5
15x25	83.07	79.2	80.53
10x30	77.44	78.78	76.56

In this image,amount of data is written as first value is training data number, and second one is testing data number. For example, ‘30x10’ means, 30 training article, and 10 test article which corresponds to 75% training. I want minimum training data to know an author’s writing style, and maximum test to see my predictions’ correctness. And even 37.5% which has 15 training data and 25 testing data, is enough to make good predictions. The difference between 37.5 percentage and 75 percentage is not so big, therefore this shows us that my attributes are working good. And with 10 training I can learn author’s characteristics with a small difference according to 30 training, so 10 articles are almost enough to predict correct author with my attributes.

For authorship, it is important to predict correctly. Our main aim is to predict the author of given text. And also, I want to try to predict gender and age, if it is possible to do with these attributes. I have used my other two datasets, one is for gender and has two groups as male and female inside; other one is for age and has three groups as young, middle-aged, and old inside. When I have looked at the results of them, they are also has approximately 80 percentage of success rate with the most suitable method for my experiment which is Multilayer Perceptron. However, I have created 4 fake gender datasets, and looked at the results.I have tried this experiment three times. At first one, I have preserved the order of instances and analayzed the result. At second and third one, I have randomly selected the articles and analayzed again. The results have showed me that, this gender dataset is just looking at the authors in its group, and tries to know them. It does not matter who the author inside a lot. If having two groups female or male, or having mixed two groups are not has too much affect. However, even with a small amount, the original gender dataset has the biggest percentage of correctly classified rate according to fake gender datasets. And it is also the same for age dataset.

Conclusion

Choosing appropriate attributes is the most important part of classification. While using an attribute, the performance time for prediction is another concept to deal with. Testing attribute before deciding to use for an experiment is an important trade-off.

In this experiment, there exists 3 datasets, and 19 attributes for classifying these datasets. Much attributes are analyzed and these 19 attributes are choosen as best performance. Different machine learning algorithms are tested, and Multinominal Perceptron and Naïve Bayes algorithms are the best algorithms for this experiment.

The project which works just 10 authors is improved and now it predicts the given texts with 30 authors. The prediction is for finding the author of given text, and also gender and age analyzes are added to the experiment. The gender and age datasets are based on the grouping the people inside them, and predicting the possible characteristics inside this group.

For having affective attributes, ratios are used instead of counting the values. For future works, this can be a good affect; and the project can work not just the whole article, but also some part of an article. In the future, gender and age datasets can be improved with some features which are not just based on grouping but also specifically finding the gender of the author.

References

[1] Amasyalı, M.F., Diri, B.: Automatic Turkish Text Categorization in Terms of Author. Genre and Gender, NLDB, Klagenfurt, Austria, 221–226 (2006)

[2] Taş T., Görür A.G., Author Identification for Turkish Texts, 2007

[3] Kucukyilmaz T., Cambazoglu B.B., Aykanat C., Can F, Chat Mining for Gender Prediction