Authorship Attribution

Deepak Munjal
Dalhousie University
Halifax, Canada
deepak.munjal@dal.ca

Mihir Joshi
Dalhousie University
Halifax, Canada
mihir.joshi@dal.ca

Pradeesh Sivakumar
Dalhousie University
Halifax, Canada
pr835572@dal.ca

Abstract–The purpose of this project is to find the most likely author of a specific text by defining an appropriate characterization of documents that capture the writing style. This is also known as Authorship Attribution. In the back-end, we have used Machine Learning and Natural Language processing techniques to achieve this goal and in the front-end, we have used HTML, JavaScript and its D3 library for visualization of the data and results. But before applying the machine learning techniques, pre-processing of the data is also done. Preprocessing includes removing punctuations, numbers, special symbols, stop words etc. Common N-Grams classification method issued for authorship attribution. In this method, all text is converted into n-grams[6] and CNG distance formula[10] is used to find the distance between two text documents. Various values of n are considered as part of our experimentation, but n=3 is found to be the best for this scenario. For visualization, we used many visual elements like a bar chart, labeled force layout, clustering, concept map etc. Django framework is used as middleware to connect front-end and back-end. The execution time of the back-end algorithm is about 5 - 6 minutes and accuracy achieved on the test data is around 90%.

Keywords–Common N-Grams, Visualization, Document-Term Matrix, N-grams, D3 Library, Clustering

I. INTRODUCTION

Authorship Attribution is the task of identifying the most likely author of a text document. Every Author has a unique writing style. Authorship attribution includes finding the most likely author of the text. Since every author's writing style is unique, it is possible to characterize an author's writing by a unique set of features that define the writing style of the author. We used CNG distance to solve this author attribution problem[10] and represented our data & results using visual elements with JavaScript D3 library.

II. MOTIVATION

Over the last few years, there has been a tremendous increase in the use of Internet users and the amount of text data available on the Internet. With this increase in the number of digital resources on Internet, there has been an increase in the issues of plagiarism, copyright infringement etc. Digital copies are easy to reproduce and hence the real author of those resources is sometimes not credited for their work. An author's work is their own novel creation and it belongs to them and hence the author should be credited each time their work is used. With the increase of such issues, a requirement for such tools and solutions for author attribution problem arises.

III. DATA DESCRIPTION

Dataset has been taken from the UCI Machine Learning Repository. The name of the dataset is "Reuter_50_50 Data Set"[11]. This dataset is a collection of various text documents. The downloadable compressed folder name is C50.zip and its size is 7.8MB. Attributes of this dataset are character n-grams. This dataset is a subset of the RCV1 dataset. The rcv1 dataset is a multilingual and Multi-view text categorization test collection dataset. It contains text documents from five different categories namely - original English documents, French documents translated into English, German documents translated into English, Italian documents translated into English and Spanish documents translated into English. When extracted, C50 directory contains 2 folders, namely C50train and C50test. Its size after extraction is about 14.8MB. Each of these folders contains 50 sub-folders. Each sub-folder corresponds to an author and contains 50 text files belonging to the corresponding author. This makes 2500 text files in both training and test corpus. So overall there are 5000 text files in the directory C50. We have used the naming convention as "authors" for the author folders in the training data and

"documents" d1, d2, …., d50 for the unknown authors in test data throughout this report and in the overall project.

IV. MATERIALS AND METHODS (DATASET(S) AND TECHNIQUES)

The raw material for our project is the dataset that we have mentioned above. The dataset name is "Reuter_50_50 Data Set"[11] and is available at the UCI machine learning repository. In terms of techniques, we have considered preprocessing as our first step. Preprocessing is mentioned in detail below. Further, we have considered Common N-Gram distance (CNG Distance) for finding the distance between two text documents. Common n-gram distance converts author's writing style into a list of features of n-grams and it helps in solving the authorship attribution problem. Agglomerative Clustering is another technique that has been considered. This method is used for clustering of authors having similar writing styles to represent them as clusters in the frontend. Details of these techniques are mentioned below.

V. DATA PREPROCESSING

Data preprocessing is an important step in applying machine learning techniques to the data. As part of the data preprocessing, all English stop words are removed, all words are converted into lowercase, special symbols are removed, and all words are stemmed. Stemming is the process of transforming words into their root form. E.g. word "going" will be converted into "go". Apart from applying these techniques, we have also considered dimensionality reduction as well. Originally, we had 4K+ dimensions. The execution time for the backend machine learning algorithm was about 60 minutes and the accuracy achieved was around 92%. We reduced our dimensions to 500, thereby decreasing the execution time of the backend algorithm to just 5 minutes. The accuracy achieved in this case is around 90%. Hence dimensionality reduction leads to decrease in the execution time without making a much negative impact on the accuracy.

VI. EMPLOYED SOLUTION FOR BACKEND MACHINE LEARNING

For solving the authorship attribution problem, there was a requirement of a text classification technique. This technique should be able to characterize author's writing style in a set of features. The importance of this problem lies in the fact that with an increase in online resources, there has been an increase in the cases of copyright infringement and plagiarism. We had multiple options to choose from. One option was cosine similarity method, which tells us the similarity between two text documents. More is the cosine similarity; closer is the two text documents. Another option was to use word embeddings vector representation to represent each document as a vector in space. The distance between vectors defines their closeness to each other. Another technique, which has been used in our solution, is using Common n-gram distance[7][10] (CNG Distance). This technique considers the frequency of character n-grams in a document. CNG distance formula is –

$$\sum_{n \in \text{profile}} \left(\frac{f_1(n) - f_2(n)}{\frac{f_1(n)+f_2(n)}{2}} \right)^2 = \sum_{n \in \text{profile}} \left(\frac{2 \cdot (f_1(n) - f_2(n))}{f_1(n) + f_2(n)} \right)^2$$

Here f1 and f2 are the frequencies of the character n-grams in the two chosen documents. Union of the two sets of character n-grams from the two chosen documents is taken. For each character n-gram, the frequency is considered for both documents. These frequency values are named as f1 and f2 in the formula. A similar process is repeated for each character n-gram from the union and this is summed over all values.

A. Clustering

All observations are divided into subsets, known as clusters. A set of observations are said to be in the same cluster if they are similar in some sense. We have used agglomerative clustering method for clustering of authors having similar writing styles. It is a hierarchical clustering algorithm. This follows a bottom-up approach where each observation begins in its own cluster, and pairs of clusters are merged while moving up the hierarchy. This clustering information is used to represent the authors, having similar writing styles, in the same cluster in the frontend. More is the similarity between writing styles of two authors, less is the distance between them. If the distance is up to a certain threshold, the authors will be in the same cluster. We can choose this threshold from the frontend.

B. Justification for the chosen solution

As discussed above, we had a variety of solutions available for authorship attribution problem. The first solution was cosine similarity method. It is a measure of similarity between two vectors and measures the cosine of the angle between them. The second solution was to use word embeddings vector representation. The third solution was using Common n-gram distance (CNG Distance). Out of all these solutions, the third solution, i.e. CNG distance is found to be best suited to our goals as it is very simple yet accurate. Further, we have chosen n=3, as it is proved to be the best trade-off between accuracy and execution time.

VII. EMPLOYED SOLUTION FOR VISUALIZATION

D3 JavaScript library is used for visualizing documents based on data and it provides full capabilities of modern browsers for combining visualization components[3].

1. Bar chart
2. Labeled Force Layout
3. Clustering
4. Concept Map
5. Visual Elements with Live data

Initially, if you click on evaluate button it takes 5 to 6 minutes to show the visual elements on the web page. From second time, it will take only 5 to 6 seconds to show all elements in the web page. In general, wherever we have included a slider for interactivity clicking doesn't work, only drag and drop option will work. Fig. 1. shows a webpage without any visual component.

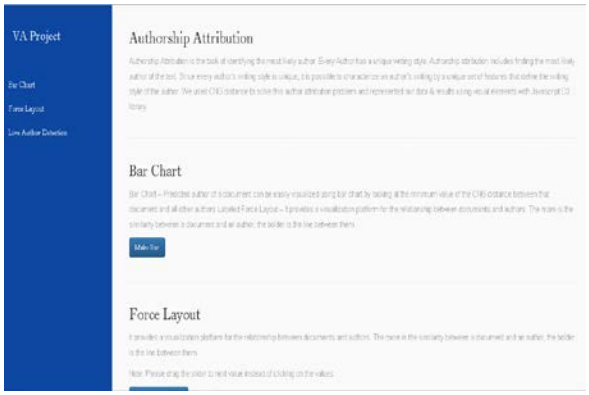

Fig. 1. Frontend Default Homepage

1. Bar chart

The main reason for choosing the technique is that the predicted number of an author can be easily visualized using bar chart technique, and this can be identified by looking at the minimum value of the CNG distance between the documents and all other authors present in the text document. The main advantage of using bar chart is that it shows large dataset in visual form.

In the bar chart, D1, D2, D3…D6 represents different text documents in the graph, if the user selects D1, it will display the list of 50 authors in text document D1 with respect to n-gram distance as shown in Fig. 2. Similarly, if the user selects D1 and D2. It will display the list of all authors corresponding to the text document D1 and D2 with respect to n-gram distance. In the graph we have shown, documents D1, D2, D3…D6 have been selected and this displays all authors corresponding to text document D1, D2, D3…D6 with common n-gram distance. We take the existing grouped bar chart[14] and changes the data along with colors. Checkboxes(Fig. 3.) and radio buttons are added to further increase the clarity of the solution and incorporate different values of n where n the number of grams. Fig. 4. shows the graph when n=4 is selected. We also changed the angle of the labels on the x axis, so they don't overlap.

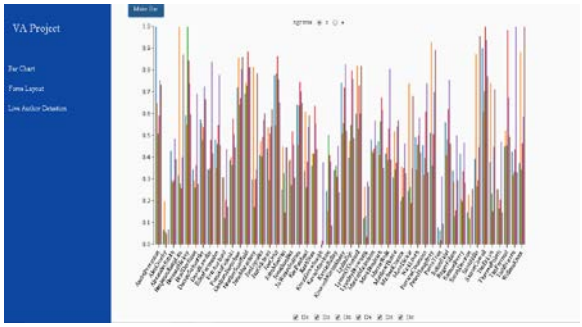

Fig. 2. Bar Chart considering the six random documents and n=3

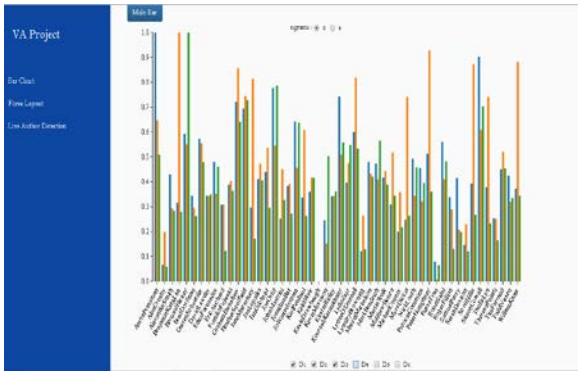

Fig. 3. Bar Chart considering first three documents and n=3

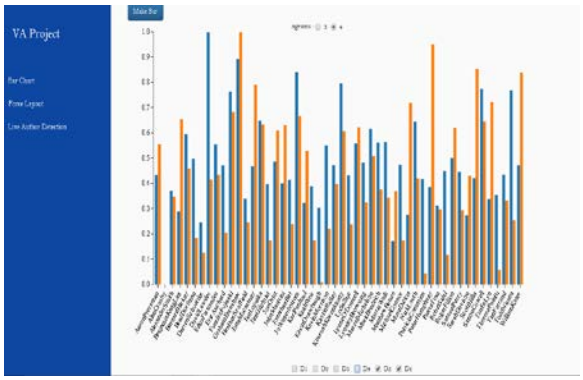

Fig. 4. Bar Chart considering last two documents and n=4

2. Labeled Force Layout

Labeled Force Layout in Fig. 5. provides a visualization platform for finding the relationship between the

documents and authors. The more similarity between a document and an author, the bolder is the line between them. In the force layout graph, each node represents the different authors, and the links represent how strongly the authors are connected to each other based on the n-gram distance. The main advantage of using this technique is that it shows interactivity to the user with flexibility where the user could drag and drop the nodes anywhere.

For example, In the graph, Todd Nissan is connected to Peter Humphrey, where nodes represent the name of the authors and the links represent how strongly they are connected to each other based on the common n-gram distance. On the top of the graph, there is an option called Normalize threshold as shown in Fig. 6., If the user clicks on .6 option it will display all the names of the authors corresponding to n-gram distance of .6 and similarly, if the user clicks on .8 as shown in graph, it will display the list of authors with respect to the n-gram distance of all .6, .7, and .8 and the distance between each node represents how strongly they are connected to each other.

The existing solution[15] is changed in a way that only test authors are connected to all other authors. We also applied changes to the nodes so that test authors are denoted by an image and all others using a circular object. The links are also modified, and the thickness of each link denotes how strong a connection is between two authors. There is a slider[19] as well that changes the number of links based on a normalized value.

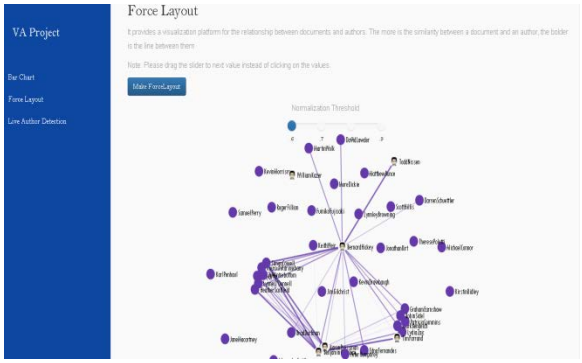

Fig. 5. Labeled Force Layout with normalization threshold=0.6

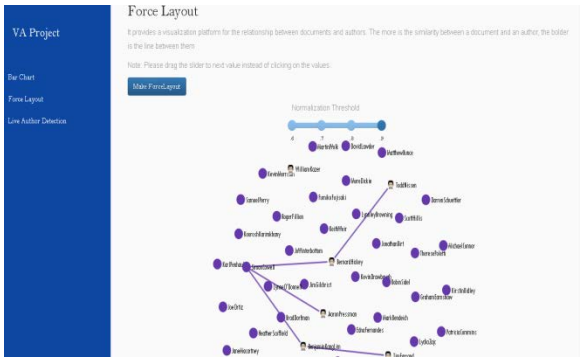

Fig. 6. Labeled Force Layout with normalization threshold=0.9

3. Clustering

Clustering is the "process of organizing data into groups whose elements are similar in some way so that it can be grouped together"[1]. The type of clustering we have used is "Agglomerative Clustering". Agglomerative Clustering is a bottom-up clustering method where each cluster will have sub-clusters which in turn will have sub-clusters and goes on. Some of the advantages of this clustering are that it can produce an ordering of the data's which it may be informative, and the second advantage would be small clusters can be generated which will be helpful for discovering data[2].

Clustering technique we used is used to find some similarity between writing styles of some authors to some extent and this visualization is used to identify clusters of authors which has almost similar writing styles present in the text document. The existing solution[16] is changed such that each item in a cluster will have the same size as compared to the original code where size is based on a value but in our solution, that doesn't make much sense. There are also radio buttons that change the number of clusters to be made.

4. Concept Map[5]

The concept map is a technique which is used to organize data and represents data of the domain.

The main advantage of this technique is to help users with new concepts and key ideas. The main use of Concept Map technique is used to find the visualization of top n-grams of the author and authors corresponding to a list of n-grams. The existing solution[17] is used as is and only colors code is changed along with modification in internal logic to fir our dataset.

5. Visualization Elements with Live data

Visualization with live data gives real-time experience to the users and the most important thing is that is more useful for the data analyst to communicate their findings via popular media in data visualization. In the graph shown, if the user types some text in the text field it will stream live data using different techniques like the bar chart, clustering, and force layout. The most important feature for using this technique is that it helps to visualize live text data using bar chart (Fig. 7.), labeled force layout (Fig. 8.), clustering (Fig. 9.) and concept map (Fig. 10.).

Fig. 7. Bar chart showing normalized scores of top six matching authors with the live data. The simple bar chart[12] and animated bar chart[13] codes are combined in such a way that bar changes with an animation every time a user writes something in the text area.

Fig. 8. Labeled Force Layout showing normalized scores of top six matching authors with the live data

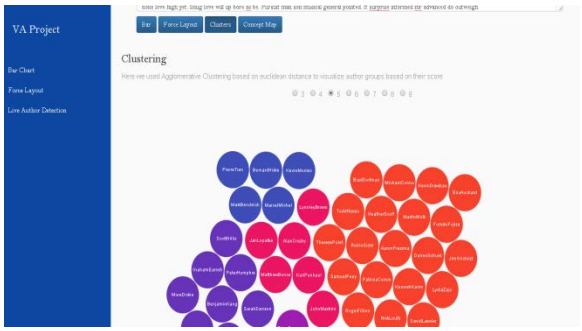

Fig. 9. Agglomerative clustering considering number of clusters as 5

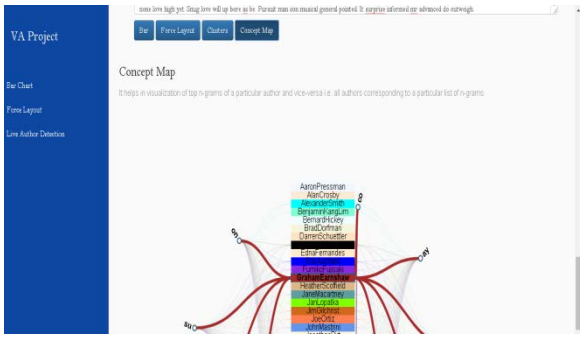

Fig. 10. Concept Map representing top n-grams of an author and vice-versa

VIII.DJANGO FRAMEWORK/ MIDDLEWARE

Django Framework is a middleware component which is responsible for handling specific function and it's helpful to build complex web application simply and quickly. It is written in Python programming language[4].

The Django framework is used to connect Python backend code with a Front end. Some of the benefits of this framework include:

Provides quick web project development

Delivers high-quality code writing

Versatile and has no glaring holes present in the design

IX.CODE ARCHITECTURE

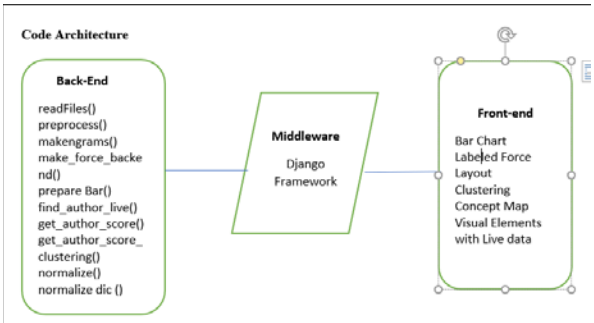

Fig. 11. Code Architecture

Back-End

read Files()- Parameter of reading files Function is the path of the dataset and it returns the file contents and the author names in two different lists.

preprocess()- The attribute of the function is the list containing the contents of all the text files and it removes many special symbols and English stop words. Further, it covers all words into lowercase. After all this preprocessing it returns the same list containing the text data.

make grams()- This function has two parameters - a value of n and list of contents of text files. It returns the list of lists containing unique n-grams (In our case they are trigrams) For each author it also returns the list of lists containing the corresponding normalized count of each unique n-gram for each author.

make_force_backend()[8][9] – This function has many parameters-List of lists containing normalized counts for each author, List of lists containing unique trigrams for each author, list of lists containing the normalized counts for each author in text data, list of lists containing unique trigrams for each author in text data, list of author names, all the features of document-term matrix, list of author names in text data.

It returns a dictionary containing nodes and links for visualization of force layout in the front end.

prepare bar() – It also has many parameters -list of lists containing normalized counts for each author, List of lists containing unique trigrams for each author, list of lists containing the normalized counts for each author in text data, list of lists containing unique trigrams for each author in text data, list of author names, all the features of document-term matrix, list of author names in text data.

It returns a dictionary containing the normalized CNG distances between a list of documents and all authors. This data is used for visualization of the bar chart on the front end.

find_author_live() – This function has also had the same list of parameters as the above-discussed preparer function and it returns a dictionary of author names and their CNG distances.

get_author_score() – It also accepts the same list of parameters as the above two discussed functions. It applies agglomerative clustering technique on the data and returns the clustering scores for visualization of clusters, of authors having the similar writing style, in the front end.

normalize () – As the name says, it is used for normalization of data. This function accepts a list containing the numerical values. It returns the normalization factor and the maximum value.

normalize die () – This function is used for normalization of dictionary values. It accepts of the list containing numerical values. It evaluates the normalization and it returns a normalized dictionary.

Middleware

Django framework- The Django framework is used to connect Python backend code with a Front end.

Front End

Bar chart- The bar chart is a graph which represents data in the form of rectangular bars for each value present in the table. This can be represented in the form of the horizontal and vertical graph.

Labeled Force Layout-This technique provides the platform for finding the relationship between the documents and the authors.

Clustering-This technique is used to identify the cluster of elements which almost have same writing styles with subclusters[1].

Concept Map-This visualization technique is used to find the top n-grams of an author and vice versa present in the text document[5].

Visual elements with the live data-These techniques are used for visualizing live data using live text data in the form of the bar chart, clustering, and force layout.

X. RESULTS, USE CASES, AND EVALUATION METRICS

For the visualization purpose, we have considered six random authors from the author pool. These are – AaronPressman, Benjamin Kang Lim, Bernard Hickey, Tim Farrand, Todd Nissen and William Kazer. We have provided an option to choose the value of n is 3 and 4 in the frontend for bar chart representation. We didn't consider n=2 or n>4 because for n=2, the number of features will be large and there will be a lot of duplicities. Similarly, n>4 will also be not of much use. Hence n=3 and n=4 are considered for our case. For n=3, all authors except Tim Farrand is being predicted correctly. Hence five out of six authors were predicted correctly. So, accuracy can be considered as 5/6, i.e. 83.3%. Similarly, for n=4, first three authors i.e. AaronPressman, Benjamin Kang Lim, Bernard Hickey are predicted correctly. Hence three out of six authors were predicted correctly. So, accuracy can be considered as 3/6, i.e. 50%. But as mentioned above, for n=3, the accuracy of the backend algorithm over the whole test data is around 90%.

XI. CONCLUSION

In Conclusion, we came to know that Author Attribution problem has been successfully resolved using Machine Learning and D3 JavaScript Library and the accuracy we got is 90% with execution time about 5 minutes using backend algorithm. Finally, we would like to say that Visualization elements make understanding of the solution better and much simpler.

XII. FUTURE POSSIBILITIES- BEYOND THIS COURSE PROJECT

The main use will be the deep neural network which keeps into consideration that the prior state will act as an input for the next state. The other future possibility will be two texts which depend on the semantic meaning of the sentence as well. These are the future possibilities that we think could improve.

XIII. PROGRAMMING LANGUAGES AND TECHNOLOGIES USED IN THE PROJECT

We have used Django 1.11.3 and Python 3.6.1 for the project. We have also used nltk and learn library for the backend code. Few other optional dependencies include - Django-hosts, unicorn, numpy, pytz and white noise. In the frontend, we have used bootstrap[18] technology and D3 Javascript library. In the backend, we have used Scikit-learn machine learning library.

XIV. REFERENCES

[1] Anon, (2017). [online] Available at: https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/

[2] Anon, (2017). [online] Available at: http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Agglomerative_Hierarchical_Clustering_Overview.html

[3] Bostock, M. (2017). D3.js - Data-Driven Documents. [online] D3js.org. Available at: https://d3js.org/

[4] Docs.djangoproject.com. (2017). Middleware | Django documentation | Django. [online] Available at: https://docs.djangoproject.com/en/2.0/topics/http/middleware/

[5] Inspiration.com. (2017). How to use a Concept Map to organize and comprehend information | inspiration.com. [online] Available at: http://www.inspiration.com/visual-learning/concept-mapping

[6] H. python, "How to extract character n-gram from sentences? - python", Stackoverflow.com, 2017. [Online]. Available: https://stackoverflow.com/questions/22428020/how-to-extract-character-ngram-from-sentences-python

[7]2017. [Online]. Available: https://web.cs.dal.ca/~vlado/csci6509/notes/nlp10.pdf

[8] S. list?, "Sorting list based on values from another list?", Stackoverflow.com, 2017. [Online]. Available: https://stackoverflow.com/questions/6618515/sorting-list-based-on-values-from-another-list

[9] O. Python, "Ordered intersection of two lists in Python", Stackoverflow.com, 2017. [Online]. Available: https://stackoverflow.com/questions/23529001/ordered-intersection-of-two-lists-in-python

[10] Web.cs.dal.ca, 2017. [Online]. Available: https://web.cs.dal.ca/~vlado/papers/pacling03.pdf

[11]"UCI Machine Learning Repository: Reuter_50_50 Data Set", Archive.ics.uci.edu, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Reuter_50_50

[12]"Bar Chart", Bl.ocks.org, 2017. [Online]. Available: https://bl.ocks.org/mbostock/3885304

[13]"Animated Horizontal Bar Chart with Tooltips", Bl.ocks.org, 2017. [Online]. Available:

http://bl.ocks.org/juan-cb/ab9a30d0e2ace0d2dc8c.
[Accessed: 18- Dec- 2017].

[14]"Grouped Bar Chart", Bl.ocks.org, 2017. [Online]. Available: https://bl.ocks.org/mbostock/3887051. [Accessed: 18- Dec- 2017].

[15]"Force-Directed Graph", Bl.ocks.org, 2017. [Online]. Available: https://bl.ocks.org/mbostock/4062045. [Accessed: 18- Dec- 2017].

[16]"Clustered Force Layout III", Bl.ocks.org, 2017. [Online]. Available: https://bl.ocks.org/mbostock/7881887. [Accessed: 18- Dec- 2017].

[17]"Concept Map playlist visualization generated by Exaile 3.4 beta3", Bl.ocks.org, 2017. [Online]. Available: http://bl.ocks.org/virtuald/ea7438cb8c6913196d8e.
[Accessed: 18- Dec- 2017].

[18]"Simple Sidebar - Bootstrap Sidebar Template", Start Bootstrap, 2017. [Online]. Available: https://startbootstrap.com/template-overviews/simple-sidebar/. [Accessed: 18- Dec- 2017].

[19]"Slider for Bootstrap Examples Page", Seiyria.com, 2017. [Online]. Available: http://seiyria.com/bootstrap-slider/. [Accessed: 18- Dec- 2017].