Report

The project is to build a system to classify the Level of writing samples by English language learners, using a data set gathered from users.

CODE FOR THE CLASSIFICATION OR CLUSTERING ALGORITHM(S)

Code organization (see readme.txt)

See code details on github at https://github.com/tia-e/text-classification-clustering

DESCRIBING YOUR APPROACH

Problem 1 & 2: Classify writings into Levels & Groups.

Data = XML file of writings of EF students at different levels. A level is comprised of different units. There is a linear progression from one level to another.

Class: Level 1 to 16 & Group [A1, A2, B1, B2, C1, C2]

Approach:

1) Data loading and cleaning: The first step was to read and clean data and store it in a dataframe.
   - There were some <br>, <code> balises, that were removed.
   - Empty writings were not included in the dataSet
2) Visualize Data: plot different figures to get a sense of the data (Figure 1, 2)
   - Distribution of class (group/level) in the data
   - Mean and standard deviation of the distribution of some features (word count, average number of words per sentence, grades)
3) Data Sampling
   - For memory and time performance constraints a sample (10% of original data) will be used for the next steps.
4) Data preparation: Pre-processing and features extraction
   - Tokenization
   - Ngram
   - Tf-idf
   - Other features used (word count, average number of words per sentence, number of punctuations)
5) Train/Test Split
   - 80% of data is used for training (and cross validation)
   - 20% is used for Testing of the final model
6) Application of several classifiers to the train dataset using 10-fold Cross-Validation to select the right classifier and the right features.

- Logistic Regression
- Naives Bayes
- Decision Tree

7) Test of the final model using the 20% remaining data
    - Evaluate performance with metrics: Precision, Recall, F1 score, Confusion Matrix.

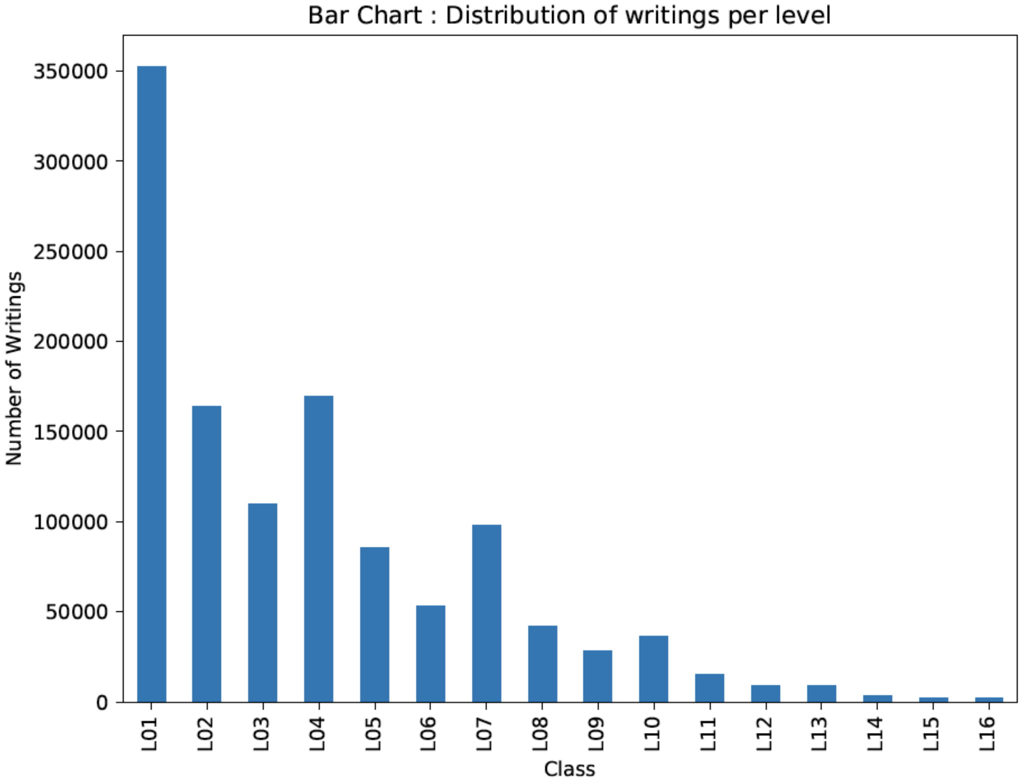

Figure 1 : Bar chart level

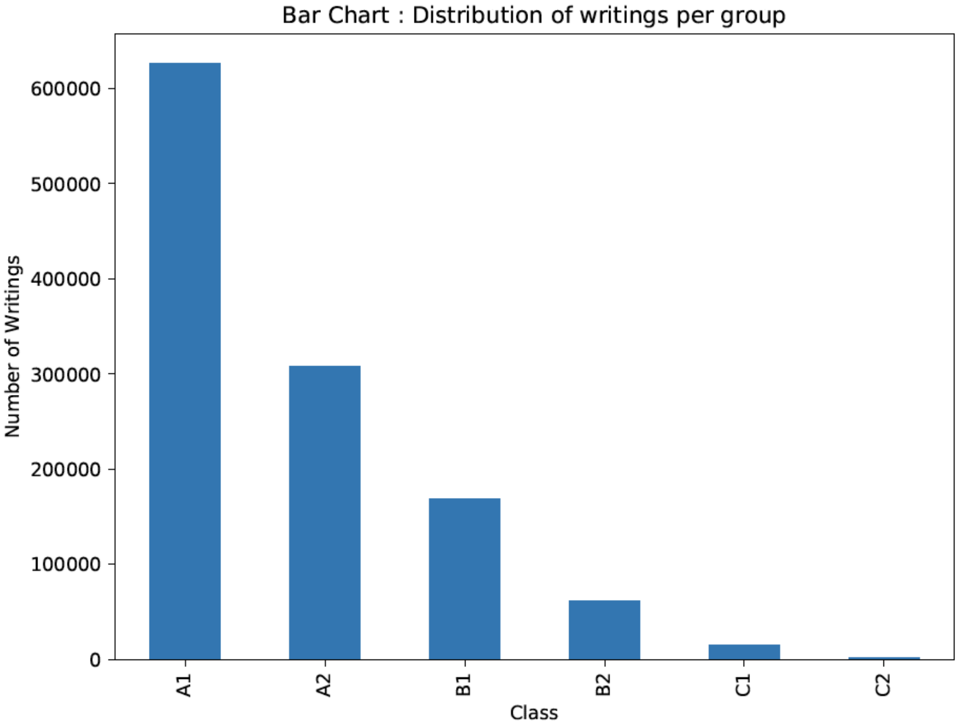

Figure 2 : Bar chart Group

Results Classification Level

Cross-validation shows that Logistic Regression (LR) gives better results than the other models (Naïve Bayes, Decision Tree, KNN). Final results for LR are presented here.

Level	precision	recall	f1-score	support
1	0.89	0.98	0.93	6971
2	0.93	0.88	0.90	3186
3	0.93	0.85	0.89	2219
4	0.91	0.90	0.91	3421
5	0.88	0.90	0.89	1667
6	0.90	0.85	0.87	1070
7	0.87	0.93	0.90	1944
8	0.84	0.76	0.80	862
9	0.90	0.81	0.85	568
10	0.89	0.89	0.89	729
11	0.86	0.67	0.75	320
12	0.92	0.66	0.77	182
13	0.90	0.67	0.77	197
14	0.93	0.35	0.50	75
15	1.00	0.20	0.33	46
16	1.00	0.09	0.16	35
total	0.90	0.90	0.89	23492

Results Classification Group

Test result for Logistic Regression

Group	Precision	recall	f1-score	support
0	0.95	0.98	0.96	12376
1	0.92	0.90	0.91	6158
2	0.90	0.89	0.90	3374
3	0.89	0.79	0.84	1231
4	0.95	0.50	0.65	318
5	1.00	0.11	0.21	35
Total	0.93	0.93	0.93	23492

Problem 3: Find structure in the data / Clustering

Data = XML file of writings of EF students at different levels. A level is comprised of different units. There is a linear progression from one level to another.

Approach:

1) Data loading and cleaning: The first step was to read and clean data and store it in a dataframe.
    - There were some <br>, <code> balises, that were removed.
    - Empty writings were not included in the dataSet
2) Data Sampling
    - For memory and time performance constraints a sample (1% of original data) will be used for the next steps.

3) Text Clustering
    - K-means algorithm with k = 6 was used.
4) Performance Measurement
    - Visualization as performance measurement
        - Multi-dimensionality scaling (PCA)
    - Evaluate performance with metrics: Homogeneity, Completeness, V-measure

Results Clustering - Kmeans

inertia	Homo	compl	v-meas	ARI	AMI	silhouette
1791	0.030	0.022	0.026	-0.033	0.022	0.511

CONCLUSIONS AND FURTHER WORK

- The task of the homework was to build a classification and clustering system for writings data. Overall Logistic Regression performs the best with a precision of 90% f1 score using TF-IDF for Level class and 93% for Group class.

- One aspect about the clustering worth notice is that the similarity measure should look for similarity on text-complexity, not just on text-topic. (We want to know if two texts present the same complexity more than they are on the same topic)

- The classes are ordinal variables, meaning L1 < L2 … < L16 and A1< A2 < …< C2, but the different Classifiers used do not take this information into account. It could be interesting to work in that direction in the future.