Writing Level Classification
-Ritwik Kulkarni

Introduction:

We implement a model that classifies the "writing level" of pupils learning English, into various "levels" based on a text paragraph written as a part of an assignment. Two versions of this task are implemented,
a) 16 levels are defined according to data ('groupby =1' , see code) and
b) the 16 levels are grouped together as sets of three consecutive levels, for e.g. level (1,2,3) = A, level(4, 5, 6) =B and so on ('groupby=3', see code).

Given the context of the task, we take the approach of defining features that can possibly capture the proficiency of English language in the text. To that effect, we hypothesise the following features as indicators of level of proficiency:
1. total length of the paragraph (number of words)
    -longer paragraphs indicating advance level
2. median length of words in the paragraph
    -longer words indicating advanced level
3. number of unique words excluding
    -higher number indicates better vocabulary hence an advance level
4. entropy of the paragraph:
    -higher the entropy of the paragraph, indicates more information content in the paragraph as opposed to repetitive words and hence an advance level
5. number of spelling mistakes made
    -higher the mistakes lower the level
6. rank value, sum of the ranks of top five highest ranked words in the paragraph. Rank is defined based on the frequency of occurrence of the word in the wiki-corpus. The less frequent the word the higher the rank
    -if rarer words are used in the text then possibly an advanced level of English

*features 5 and 6 have the code infrastructure ready, however due to high computational time and limited personal resources they were omitted. They can easily be included by replacing the 'get_features()' method by 'get_features_extra()' method. We encourage performing that experiment.

Another feature of high interest is the number of grammatical errors in a sentence. This can be done in principle using language_tool in python, however it needs a server connection which slows things down considerably and requires elaborate settings and hence omitted.

*It was decided not to do the 10 fold cross validation due to lack of personal resources and high computational time on large data.

Model:

The model itself is a very simple one, with an input feature layer, two hidden layers of size 20 and 10 respectively with rectified linear units and a softmax non linear final output layer. The model is trained using the ADAM optimiser for 50 epochs.

Results:

Task	Accuracy	Precision	Recall	F-score
A) Un-grouped level labels	0.53	0.12	0.18	0.14
B) Grouped level labels	0.84	0.49	0.48	0.48

*Precision, Recall and F-score metrics are macro averaged.

Discussion:

The model does exhibit some level of learning, however the performance is very poor. The discrepancy in the accuracy and precision metrics indicate a possibility of an unbalanced dataset, either unbalanced batch wise or unbalanced statistically. Grouping the output labels does have a considerable improvement in the performance of the model. This can most likely be attributed to the fact that large portions of the variance are mapped onto a single label.

Overall, even though there is a hint that the problem of writing level and text features can be mapped to each other, there is much scope of improvement in the model. Several possible directions can be explored.
a) Starting with the obvious one that feature engineering can be expanded and refined.
b) Model design can be made more robust including hyper parameter tuning
c) The dataset itself needs to be carefully observed for balance and other statistics which can help in feature engineering
d) The “topic” of the paragraph can possibly be an indicator of the level, however this maybe considered as cheating
e) class-wise f-score needs to be investigated for a detailed analysis and possible future insights

Unsupervised Clustering:

We attempted unsupervised clustering on 10k data points of the same data based on the 4 features described above. The DBSCAN algorithm was used to “discover” clusters in the data which yielded a total of 12 clusters with 80% of the mass contained in a single cluster. Visual inspection of the clusters based on their “level”, “units” or “text” did not reveal any specific pattern. However this was a mini trial of the experiment and further exploration of the algorithm parameters as well as data features is required.