# SPEEDLABS QUESTION CLASSIFCIATION USING NLP

Work by: Abhinav Tiwari
Work Duration: 06/12/21 – 06/01/22

## PROBLEM STATEMENT

SpeEdLabs has many questions in their database and the size of this will continuously increase as more questions are being added, therefore there is a need to develop machine learning pipelines to atomate the process of classifying the questions to their respective categories without the need of manual work. The aim is to develop an NLP pipeline and train it on preexisting data and then test how it performs in classification of new data.

## OVERVIEW OF AVAILABLE DATA

The training data consisted of questions from subjects of physics, chemistry, mathematics, and biology, with data from each subject divided into test and train data randomly as follows –

Physics – Example data- 836, Trial data - 250

Chemistry – Example data- 1500, Trial data - 500

Mathematics – Example data- 1500, Trial data - 500

Biology – Example data- 1517, Trial data - 411

Each question has information about the Chapter, Subject, Topic, KSC and Question Text

## PHASE- 1

During this period, I analyzed the data and did preprocessing, as described below –

- Combined the csv question data with the latex data
- Filtered images with image data and checked if there is any problem with conversion of latex into the image
- Visualized the amount of information available for each subject, in terms of chapter, KSC and topic and analyzed the proportion of each in our dataset
- Preprocessed the question text by removing noise in the data such as latex keywords and stop words. Used stemmer to convert words into their root form and did data cleaning.
- Implemented TF-IDF vectorization in our dataset to represent question is vector format so that they are compactable with the machine learning models

## RUNNING K-MEANS AGLORITHM ON THE DATASET

Model - After preprocessing the data, I ran a K- Means algorithm on the biology dataset to classify the questions according to the number of chapters in the dataset. I only took chapters having > 20 questions in the testing and training data (n = 25). The resulting clusters were labelled according to the chapter occurring with maximum frequency in that cluster

Results – The above model was able to represent 9/15 chapters in the clusters, as a matter of fact only chapters having >60 questions were represented dominantly in at least 1 cluster. Some chapters were represented dominantly in more than 1 cluster. On running with the test data, we got a classification accuracy of 56 %.

Inference – The dataset in the analysis is unbalanced, some chapters are represented much more dominantly than others. Certain chapters have < 20 questions whereas some have > 200. Data at the KSC level is not sufficient to perform a informative cluster analysis. We have 500 KSC classes in 7000 questions dataset. In fact, most of the incorrect predictions in K- means belong to a single chapter which has the highest question. This shows that unsupervised algorithms are not currently suitable for our dataset and suffer from imbalanced class representation.

## PHASE -2

Having discovered the nature of dataset and problems posed due to it, I tried deep learning methods to classify the questions. Broadly I describe three models used and their results

## CNN MODEL

Model -

| Layer (type) | Output Shape | Param # |
|---|---|---|
| embedding_2 (Embedding) | (None, 140, 300) | 1500000 |
| conv1d_2 (Conv1D) | (None, 138, 100) | 90100 |
| max_pooling1d_2 (MaxPooling 1D) | (None, 69, 100) | 0 |
| flatten_2 (Flatten) | (None, 6900) | 0 |
| dropout_4 (Dropout) | (None, 6900) | 0 |
| dense_4 (Dense) | (None, 10) | 69010 |
| dropout_5 (Dropout) | (None, 10) | 0 |

```
 dense_5 (Dense)             (None, 3)                 33

=================================================================
Total params: 1,659,143
Trainable params: 1,659,143
Non-trainable params: 0
```

Results - The above CNN model fails to give appreciable accuracy for KSC prediction which is about 20% even after extensive training. The accuracy for subject classification is 99%. The chapter wise accuracy for subjects are as follows –

Physics – 62.5%  Chemistry – 82.26%  Biology – 74.4%  Total – 76.6 %

Inference – From the results it seems that CNN model is capable of correctly classifying questions but it being a relatively large model, requires a large amount of training data. The KSC level data is not sufficient to train such model, also the low representation of physics data may be the reason for its low accuracy.

## 1 LAYER NEURAL NETWORK

Model –

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_42 (Dense)             (None, 100)               10100

 dropout_34 (Dropout)         (None, 100)               0

 dense_43 (Dense)             (None, 10)                1010

=================================================================
Total params: 11,110
Trainable params: 11,110
Non-trainable params: 0
_________________________________________________________________
```

Results – The model gives 46% accuracy on total KSC prediction which is a significant improvement from the CNN model also the subject accuracy is still 99%. The chapter level accuracy is –

Math - 88%.  Physics – 60.80%  Chemistry – 82.00%  Biology – 82.4%  Total – 80 %

Inference – While this model is a very simple one, it still can predict better than CNN, this may be because of the smaller number of parameters so less data required for training. Here also accuracy for physics is low. This may be because physis data has lots of common features, on closer look many chapters are very similar to each other and it is difficult to differentiate.

## KSC CLUSTER ANALYSIS

Currently there are very view KSC cluster data at the question level, thus doing any analysis at the cluster level significantly reduces the number of datapoints. Using the marked cluster information and by comparing the predicted KSC cluster instead of Text an increase of 5% is observed in the test data. To form more KSC cluster, some metric can be used. One such which was tested is the Levenshtein distance between the KS Text labels. This helped to increase the accuracy by just 2-3 %. Its is evident that currently the size of questions under a KSC label are very small and we need better KSC cluster to classify at this level. Other methods of making KSC clusters can be investigated.

## DELIVERABLES

All the code and data files used for the above analysis can be found at the following GitHub repository - https://github.com/matrix101A/SpeedLabsNLP_project . The google Colab workspace for the jupyter notebook can be accessed via https://colab.research.google.com/drive/1CsGOwVRswLqo1T_ePLfZAtGK0rV5OzkB?usp=sharing