International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Concept Detection of Multiple Choice Questions using Transformer Based Models

Tejas Karkera[1], Amruta Sankhe[2], Prakhar Agrawal[3]

[1]Student , Dept. of Information Technology, Atharva College of Engineering
[2]Assistant Professor Dept. of Information Technology , Atharva College of Engineering
[3]Co-Founder Exam Lounge

Abstract - *It is true that almost all the competitive exams in India are MCQ based CBT's (Computer Based Tests) right from the Engineering Joint Entrance Exam (JEE) to the CAT (Common admission Test) or government exams like SSC CGL. Students devote a lot of their time studying important chapters , topics while preparing for these MCQ based exams. While evaluating their performance it is paramount for all students to understand the topics and concepts which they are weak in and should work on. It is a tedious job for students to label the topic or the concept, the question belongs to and hence manual labeling could be a time consuming task.. This paper presents an automated approach for labeling of the concepts and topics for a given MCQ question using two transformer models namely BERT* [1] *and DISTILBERT* [2] *and also doing a comparative study for contemplating on their performance. The approach used in this paper is augmented in terms of performance as it showcases the use of thresholding methodologies on the Dateset with removal of irrelevant data points if the associated probabilistic confidence is lesser for a given threshold. Overall the paper presents a pipeline which can be used for an automated concept labeling of MCQ questions and thus lessening the manual task by achieving an accuracy of around 90%.*

Key Words: Natural Language Processing (NLP) , Machine Learning , Transformer models , Thresholding

1. INTRODUCTION

It is very easy to contemplate that most of the well-known examinations for entering professional courses or degrees in colleges are multiple choice based examinations, specifically in India. The preparatory phase for some of these exams like the JEE are quite long as the syllabus is too vast . If we take the example of the JEE (JOINT ENTRANCE EXAMINATION), there are around 82 topics a candidate has to cover added with the numerous subtopics inside them. Moreover the examination demands a lot of time in practicing various MCQ questions with students giving almost a year or two for this preparation.

Most of the Ed-Tech classes nowadays have faculties or student interns who manually label the topics or subtopics for each of these MCQ questions which can be used later for their reference. Ironically if one tries to take a rough estimate of the number of questions a student can manually label , it should not be more than 600-700 questions a day which is still a huge number.

The various drawbacks of this approach is :

1. Firstly it is very time consuming.
2. It is prone to Human error while labeling.
3. It will use labor which can be utilized elsewhere.

Hence for all these reasons human implemented labeling process of MCQ questions are not as efficient as one wants it to be and so it is evident that in this age of automation with the help of NLP based pipelines for easy processing and classification of Text, it is highly demanding to have a similar pipeline which would take in an MCQ question and predict the concept the question belongs to. This paper wishes to propose a pipeline which would take in an MCQ question, understands the context and its semantics and on these parameters predicts the concept Label . Apart from building the architecture, it is very important to evaluate the performance of the model which will be judged on various evaluation metrics like precision , recall and the F1 score for the Testing set.

2. PIPELINE ARCHITECTURAL FLOW

2.1 Understanding Data

The Dataset for this task was provided by an Ed-Tech Startup Exam Lounge [3]. The startup provides services for various examinations like CAT (Common Admission Test) , SSC_CGL , IBPS (Institute of Banking Personnel Selection.) etc and hence the questions were too a coalition of all these examinations . The Data had numerous features like the level of the question , the question with its options , the solution of the question , the

Hindi translation and it's solution and finally the concept label. Apart from this we are also provided with ' Concept-Topic-Exam ' mapping for our dataset which can be used further to add in more features for our question like the Topic and the Exam in which it occurred. MCQ questions have a variety of words and to get a basic overview of the context of these questions we can plot a word cloud. We can have some of the familiar words like 'marked price' relating to profit and loss topic or something like 'complete series' for numbers. The size of the Dataset is approximately 18000 rows and for predicting on undetermined set we have around 8000 question . Our final goal is to use the content and leverage the concept and topic hierarchy of the question for getting the final concept the question belongs to.

2.1.1 Pre-Processing Dataset

The Data which we have is in its raw form and hence usage of it directly in the model will not yield desired results and also the questions which we have currently is having a topic mapping which is a crucial part for our model hence we need to process that too. The frequent words which the MCQ questions contain is also important to contemplate to better understand the context the model would work on. For this purpose we have a word cloud which will make us visualize the frequent words.

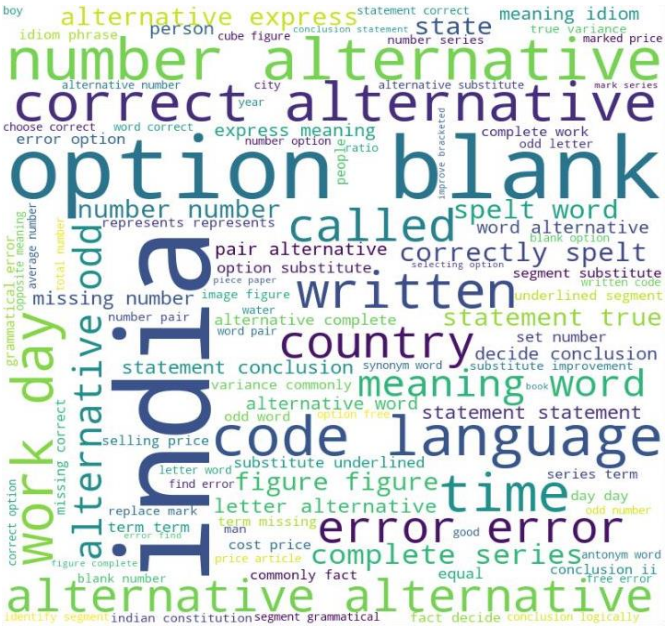

Fig. 2.1 Understanding frequent words in MCQ Questions

2.1.2 Getting the Topic and Exam Mapping for All Questions:

As we have the associated question ID's for each question, we will iterate through all the questions and for each question which we have we will extract the corresponding concept. After extracting the concept we can use the Topic to Concept mapping and find out under which topic the concept comes under. After inferring this we can assign this topic to the given question. After this process we will have topics for all the questions.

2.1.3 Splitting of the Multi label Concepts and Topics:

After extracting the topics for all questions what we inferred was there were questions which belonged to more than one concept and sometimes even more then one topic. This was a crucial point as the model should be able to understand and predict multi-label concepts if needed for a question. So we split the concept labels and feed the questions separately for each concept class. This can be done by iterating over the rows and splitting the multi label concept classes into separate rows.

2.1.4 Text Pre-processing before Feeding to the Model:

The text from the MCQ questions was very raw with HTML tags , punctuation's , abstract numbers , and also included stop-words and was unlammetized. For this purpose we had used the regex for cleaning and NLTK [4] corpus and stop-words. For removal of the HTML tags , punctuation's and the numbers, the regex expressions were "<[^<>]*>" , "[^\w+]" , "[0-9]" respectively. For stop words we can download it using nltk.download("stopwords").

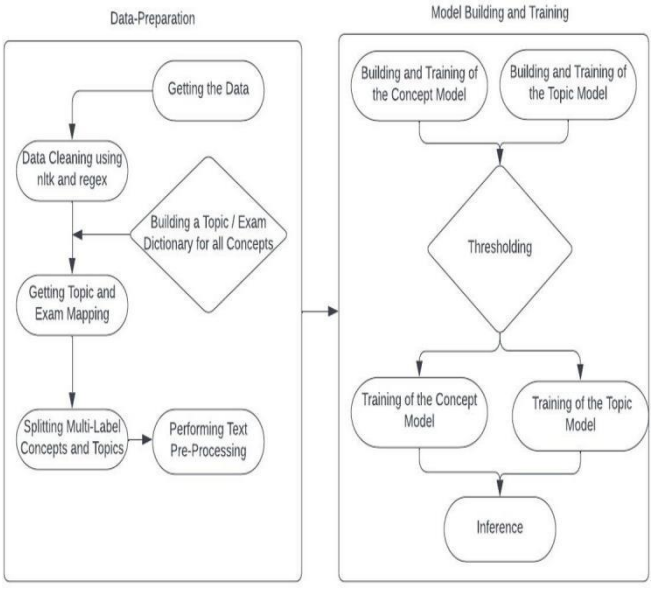

Fig 2.1 Pipeline Architectural Flow

2.2 Building the Model and Tuning Hyper-parameters

For this project we have used the Transformer based NLP models for text classification, specifically Bert [1] (Bidirectional Encoder Representation from Transformer) Model and the DistilBert [2] Model.

2.2.1 Bert Model :

Bert [1] (Bidirectional Encoder Representation from Transformer) model is a transformer based deep learning model which outperforms the previous language models as it has the ability to read and understand input text sequentially from both left to right or right to left and hence showing bidirectionality in its working. It was proposed in 2018 by google researchers and was able to achieve state of the art accuracy on various NLP and NLU tasks.

2.2.2 DistilBert Model :

DistilBert [2] as the name suggests is a distilled version of the Bert Model and is a smaller , cheaper computationally , faster in executions as it has lesser parameters and light weight transformer model which is able to achieve or preserve 90% of the Bert's accuracy and has 40% lesser parameters and is 60% faster.

For implementation purposes we would be using the Ktrain [5] library which provides the ability to set up both the models for any text classification task. The following library can be conveniently installed using pip.

2.2.3 Parameters to be passed inside the model architecture :

A. Initially while creating the model instance we need to pass in the model name in our case either "Bert" or "Distilbert" then we have to mention the Texts column and the Label column with other hyper-parameters like learning rate which is one of the following "1e-4 , 2e-5 , 3e-4" as fine tuning on them has shown good results and accordingly tuning the number of epochs.

B. We have to also mention the Maximum length Bert would be using to understand the context. It is better to use 128, 256 or 512 in this case as it has shown good results. Though if one is using longer length on a cloud based architecture like Google Colab then reduction in the number of batches is paramount to avoid out of memory error.

C. For batch sizes if one is running on cloud based architectures like Google Colab for GPU purposes it is advisable to use smaller batch sizes like 16 or 32 as larger values are susceptible to out of memory.

D. For the number of Epochs it would be different for both Bert [1] model and Distillbert [2] model because of their architectures. It is advised to go with number of epochs as 4 in Bert [1] and 6 in DistillBert [2].

2.3 Training Before Thresholding :

The results after training models before implementing thresholding for Concepts and for Topics for Bert [1] Model were 68.2 % and 89.4 % respectively.

The results after training models before implementing thresholding for Concepts and for Topics for DistilBert [2] Model were 68.7 % and 84.9 % respectively.

2.4 Inferring on Model Performance :

The Concept Models performance is not good enough to be used for evaluation purposes. The reason for this were there were a lot of classes which were having irrelevant data points showing low probabilistic confidence scores due to human errors while designing questions.

2.5 Thresholding :

Thresholding Methodology is the use of Probabilistic Confidence Scores for every data point and eliminating those data points which have a confidence score lesser then the current set threshold. The data points are passed through the model and the output is taken from the layer before final layer which has the confidence scores and then comparing it with the threshold if it is lesser then it is discarded. Thresholding helps in enhancing the precision of the model as the new set of Data points will be such that the model will be able to generalize the decision boundary in a much better way. It can though reduce the recall if the threshold is too high. Hence we have used 0.70 as the threshold for Concept model and 0.85 for the topic model.

2.6 Model Training After Thresholding :

The results after training models before implementing thresholding for Concepts and for Topics for Bert Model were 88.7 % and 91.8 % respectively.

The results after training models before implementing thresholding for Concepts and for Topics for DistilBert Model were 87.4 % and 92.6 % respectively.

BERT MODEL			
	PRECISION	RECALL	F1-SCORE
BEFORE THRESHOLDING			
CONCEPT MODEL	0.682	0.702	0.685
TOPIC MODEL	0.894	0.896	0.890
AFTER THRESHOLDING			
CONCEPT MODEL	0.887	0.909	0.895
TOPIC MODEL	0.918	0.919	0.917

2.5.1 Bert Model Performance

DISTILBERT MODEL			
	PRECISION	RECALL	F1-SCORE
BEFORE THRESHOLDING			
CONCEPT MODEL	0.687	0.697	0.684
TOPIC MODEL	0.849	0.862	0.843
AFTER THRESHOLDING			
CONCEPT MODEL	0.874	0.896	0.884
TOPIC MODEL	0.926	0.932	0.929

2.5.2 DistilBert Model Performance

3. Performance Evaluation :

After training both the models it is very important for us to evaluate the model and get predictions for unseen data . The model is able to achieve more then 85% precision for both the models after thresholding and hence can be used for evaluation. For inferring on the unseen data and extracting the concept label we would be blending the learning's from both the models and hence we should have an algorithmic flow of exactly how we would infer on it. Before the actual inference we need to pre-process this data too.

4. Label Assignment on Unseen Data :

The final part or the main motive of this paper is to infer the final concept labels for all the testing dateset we have and check the results of the model with the Exam-Lounge content Team.

4.1 Pre-processing Testing Data :

Pre-Processing would again be removing all kinds of HTML tags , removing the punctuation's inside the text and also removing all kinds of abstract numerical data in the text.

4.2 Algorithm For Label Assignment :

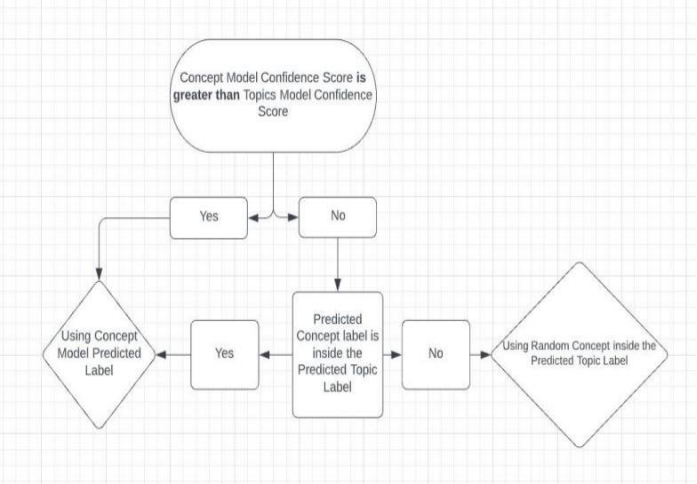

Fig 4.1 Algorithm for Concept Labeling

We would leverage both the models for predicting the concept label as follows :

1. Compare the Probabilistic confidence scores of the Concept model with the topic model.

2. If the Concept models score is high then the final Concept label is from the Concept model.

3. If the Topic models score is high then we have two cases :

A. If the Concept label is a concept inside this topic then because of this hierarchy concept label should be preferred and hence this is the predicted label.

B. If the Concept is not inside this topic then the choice boils down to predicting a random concept inside this topic.

5. Conclusion:

The Transformer Based pipeline for Concept detection after being substantiated with the thresholding methodology was able to produce decent results. This model was built and deployed in an IIT Kanpur based Ed-Tech Startup Exam Lounge [3] and reduced their content team strength by 40% . The results were verified manually by the content team at Exam lounge [3] and about 90% of the results were found to be correct. The model dwindled in performance in some areas of questions related to mathematics like 'Complete the Series of Numbers' or 'Number Analogy'. The reason being the question context for them lack in terms of words and classifying them using language models may not be the best choice. In all the performance is decent enough and can reduce the work of

the content team (manual labeling team) to a great extent.

6 . Future Work :

Firstly the foremost issue related to the model as mentioned earlier were for questions related to Mathematics and some times in Physics as textual language is sometimes less and stands more on equational flow .Thus detecting concepts using language models is not the best choice and more research is needed to classify such types of questions. Secondly trying to leverage topics and concept in a more better way is necessary because there is an unseen hierarchy between them and hence using some kind of hierarchical clustering based model is paramount.

References

[1] D. Jacob, C. Ming-Wei, L. Kenton and T. Kristina, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.

[2] S. Victor, D. Lysandre, C. Julien and W. Thomas, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2019.

[3] A. Prakhar, K. Nikhil and C. Vikram, "Exam Lounge," Lounge Innovative Education Pvt Ltd, [Online]. Available: https://www.examlounge.com/.

[4] S. Bird, L. Edward and K. Ewan, Natural Language Processing with Python., O'Reilly Media Inc., 2009.

[5] M. Arun S., "ktrain: A Low-Code Library for Augmented Machine Learning," 2020.