At first we thought to combine 4 datasets as they contain the same columns (context, question, answer, context id) and these datasets are Arabic SQUAD, ARCD, mlqa, tydiqa.

We tried some preprocessing techniques first on the Arabic squad data by luck, and this data showed unpredictable context data, some context text contains strange alone characters which can’t be understood like ا after some words and some other characters, after some time we discovered that this problem was caused by removing the diacritics from the data which is good thing but the problem was the replaceable character which is ‘ ‘ instead of ‘‘ there was white space in the place of every diacritization in the whole dataset context column.

We thought we figured out the problem and we could solve it by removing every ا and combine it the previous word but there wasn’t only the ا , there were many other characters that has no meaning even if u try to combine it the next or previous word, there were some strange characters in the contexts, but we figures out that the diacritization was in ا in the end of word and ى in the beginning of the word, so we may handle those characters, but what about the other characters? And also is there any context that may containا as it is without diacritics? What should we do about these problems?

We agreed on taking the risk in the ا and ى ways to remove the additional spaces before or after despite it may make some wrong words but the majority was of the mis replaceable space, and for the other additional character that has no meaning, we took a quick look at the data at some Arabic characters and found out, there was small occurrence when the separated character really has meaning and been separated due to diacritization, so we decided to remove all characters EXCEPT for the conjunction (حروف العطف) and preposition (حروف الجر) as they do have a meaning when they are separated like (ك،و،ب).

Why didn’t we choose another dataset instead of this one?

We looked into that solution but the data we wanted to combine as mentioned above were small, and they together less than 2000 rows while Arabid Squad’s rows more than 48 thousand, and this is a big difference between the datasets so we need it for its bid data, but we thought about something …

We will take two approaches first approach is working on Arabic squad data with its problem and second approach is to work on the other dataset combined together, it is small but it may work good and give better accuracy, and also first approach may take a lot time and in the end it may give lower accuracy and it will be wasted time, now we will work on these approaches and see what will happen …….

After some searching we made a function that search for the ا and if there space before and after, it joins the ا with the previous word and for the أ and ي if there spaces before and after them, it joins them with the next word and it worked successfully, now the data is good and ready for the preprocessing.

For the preprocessing technique used in the squad data are the function the fix the problem, checking on diacritization, remove extra spaces, remove different variation of أ and replace it with ا and removing punctuation.

For the another preprocessing techniques, we will see how to use it but after trying to get an output form the model

For the tokenizer used to tokenize we tried t5 tokenizer, but It wasn’t good as it was tokenizing spaces and some words into two tokenization, so we tried another tokenizer which is arabert tokenizer and it was better than t5.

Now we are trying to use t5 model small, but we found that the vocab size for the model is smaller than arabert vocab size, vocab size for model is around 24 000 and arabert vocab size is around 64 000, the model limit is so small for the tokenizer, we tried to use t5 tokenizer as it suitable for the model, but its vocab size is much bigger than arabert, around 250 100 which is not good with model also.

We now are trying to try 3 approaches which are try t5 medium, try arabert model, try to resize the model t5 small to the tokenizer vocab size and we will see what the result is will be ………

We tried to run the model t5 with arabert tokenizer with resizing the model to the vocab size of the tokenizer, but this was a great mistake the output of the model was garbage because the model runs with different sequence from the tokenizer, so I tried to try t5 model with its tokenizer, but there isn’t good also as the vocab size was low for the words and the output were null, there was no output.

The way now is to try mt5 model with its tokenizer but the vocab size it too large with 25100, so we are trying to use it and w will see…..

We tried to train the model and after many tries, we made it and ran the model and it runs in about 4 hours and the model was generating good question but there was some wrong thing in the output, some of the output are without answers and some with answers

We tried to calculate the accuracy of this with bleu but as bleu compute if the word does exist in the prediction and true label, so the average accuracy is about 30 %, we tried rouge metric but this one got 0 in every metric in rouge metrics

Before trying the rouge metric we decided to try more approaches like trying arat5 with the same data and this model was running the same data in about 1 hour and 20 minutes comparing to mt5 it is much faster and also its vocab size it more smaller about 110000 and also this model is much faster by 10 times and the other approach is to try to make another model with encoder and decoder each one is alone and the encoder that will generate the question will be from t5 family most well one will be araT5 as it faster than the others and also have bigger size than t5, and the decoder will be AraBert as it is good in generating answer for the questions

There is also another approach but it require a model to run it one which is to use techniques like top\_p, top\_k, temperature to generate diversity in the output and also to make the question more random to solve the problem where the mode was generating the same question (with answer if found) for the same context which is not good, so we are also trying to generate with theses techniques on mT5 and araT5 and we will see next time ….

We tried the combination with arat5 and the questions were so good and diversity, also we discovered that arat5 is better in generating the questions and also can handle English words, so we decided to continue with arat5, and we begin to train the model with the whole dataset but the model couldn’t handle training 37 thousand rows, so we thought to train the model 10 by 10 and now we are training the model with the 2nd 10 thousand rows.

And about the metrics that we will use to calculate the accuracy of generated questions, the bleu and rouge wouldn’t be efficient with our model, so we searched for different algorithms that could measure the questions quality and also the question relation with the context. We found different things like coherence that measure how much the prediction relate to the true questions and another algorithm that measure if the question have Interrogative words and question mark, there was also another algorithm that measure if the question answerable but this one is what we are going to do with the decoder model that should generate answers based on the questions. These algorithms should be find for now till we found another algorithm that fit for what we are trying to do.

Next time, I will talk more about arat5 when trained on the whole train data and about arabert when trained also on context and questions in the train data and we will measure then everything like if the question is good enough as quality and diversity and also if answerable or not and for the answers if good for the question or not ……

We have played in diversity and quality of the predicted questions; we used temperature and top\_p with many values and each parameter was responsible for something like temperature for diversity and top\_p for diversity to generate random questions with different qualities despite generating low quality and high quality.

We couldn’t test the predicted questions with rouge and bleu as these metrices are comparing between the words in predicted and true questions, which was not helpful for us, we generate different quality questions and these questions are not likely to be similar to be the exact questions as the true, we tried to search for another metrices that can test the accuracy of generated questions, a metrics that test if the question is good or not, not based on the true questions.

After a lot of searching for techniques, we found that cosine similarity is good for us, we used it to test the relation between questions and context, but this is not enough, we will try to test the generated questions with the true questions, each question will all true questions and get the best score

As this step is not enough, as the accuracy might be low, so we thought of another technique of 3 steps, first step is getting the relation between context, true with the generated questions, if low, test the answerability if the accuracy is low also, move to the third step which is giving this question to LLM model to test the questions if the question is good or not and this will be the final step to test the accuracy of generated questions, we are now in first step, let’s see what will happen next time…

When we were trying the first step which is test the accuracy only with cosine similarity and Bert score, we tried to generate 500 questions for every context of the test data that is around 10000 rows with approximately 2000 contexts, but that was a lot and took a lot of time around 12 hours in getting the accuracy and wasn’t enough the session closed auto, we tried to make it just 20 questions, as bigger than that will take much time and I just wanted to see what will happen, also 20 generated questions were not bad, after like 2 and half hours the accuracy was 0.9203 and this is so good, first time we tried to test the accuracy with cosine similarity was between 60 and 70 % and this was acceptable as we tried to get one generated questions with only one true questions even if they are not similar, but what we did was to get every generated question with all true questions and choose the max accuracy among them and so, after that get the mean of the total accuracies for each generated question for each context, what helped us more that we tried to get the accuracy using cosine and bert and after some search bert score is better than cosine similarity that is the main reason behind the last accuracy we got, bert score was the better solution for generated questions.

As we finally did the last step in making the model which is testing the model, don’t know what is the next step but we made the prompt that we will use to generate the mcqs and true & false from deepseek or chat gpt, also the accuracy of answer model is not bad at all, might be low but not that much, also had started in the backend and first design of the web, so we will see next time what we have done….

We tried something to see if bert score is really efficient or not, we tried to get a context and generate for it 20 questions as we do, but this time we revised every generated question with the true questions and their accuracy from bert and cosine to see if bert is really overrating the accuracy or not, and we found that there is some differences but in the end bert is really trying to get the best accuracy which is always high but generated questions are from the context as the true questions, so there will be similarities between the questions, we tried also the answerability, and the answers were a lot alike, so agreed that bert is not overrated, it is just see that we can’t see, the relation between the questions and it is good to use.

Till next time……….

Dr. Manar saw that we should try another dataset that mentioned in the papers with accuracy as to see if the model is really good with the bert score or not, and this dataset is tydiqa, so now I have done the preprocessing, not like the first one that applied on Arabic squad but a lot similar to it, and for the training I will do it twice, one with the local model and another train with out trained model on Arabic squad to see the difference alongside to the accuracy to compare it…..

After training the model in both ways which didn’t take a long time due to the small size of the data, the local and trained, first there was not much difference between both of them, but I observed that the training and validation errors while training was lower in the trained one than the local with around 0.6, trained model got 63 % on BLEU and the local model got 61%, that is close to that mentioned in the papers as the tydiqa dataset accuracy.

Also, the local model when predicting multiple questions on the same context, it doesn’t generate different questions, it generate the same questions with different words but all with the same meaning and the same question word like (متي), all questions have this word with different words in the question, on the trained model, there was some diversity in the questions generated, like there was different questions words like (متي، اين، كم) and other things, they were are alike to each other but there was some diversity in the questions as the model has trained on different data.

We also tried to use ARCD data to compare the accuracy with the papers, its training and validation errors were lower than tydiqa, with 0.2 with on train and validation after 10 epochs, and also there were duplicates contexts with different questions which is better than tydiqa in the diversity of the questions, generating questions with the local model the accuracy of BLEU was 0.34 and the generated question was lower quality than tydiqa dataset, with the trained model the accuracy was around 0.15 for train and validation and on the 5th epoch which is a lot better than the local model and the BELU accuracy was 0.2382 which is lower than the local model but this is because the model can generate more random questions than the local model and also the data itself contain different questions for the same context unlike tydiqa and similar to Arabic squad and that return us to the first problem why we can’t get accuracy for Arabic squad.

In the end, we got similar accuracy to that mentioned in the paper for tydiqa and arcd datasets also Arabic squad bleu metric was around 30 which is similar to a paper who mentioned that dataset, and the questions from the trained model are better than the local modle and with some diversity and randomness, and that is what we wanted some diversity that gets us better quality questions even if there are some low-quality questions among there, they may be needed in some contexts who know!