ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

ΙΕΛ
ILSP

ATHENA
Research & Innovation
Information Technologies

M915 - NLG & NLU

BERT QnA

Report

Christina-Theano (Theatina) Kylafi

LT1200012

Data Exploration

The **SQuAD 1.1** dataset was used for model training(fine tuning) and evaluation. Due to the format of the dataset (json dictionaries), in order to perform the experiments and extract the essential information (questions, answers, question id s, etc.), Pandas Dataframes were created for both saving and organizing the information in a practical structure.

For this task (**question answering**), the maximum context length is crucial to the training process and the results depend on that value, therefore it should be chosen carefully. Ideally, we could keep the maximum value with which the BERT model could work, however our computational power and time are limited, leading to a need for confinement modifications.

For this matter, we **tokenize** the sample context for both sets, **count** the samples the context of which comprise tokens below a certain sum (50, 100, 150, 200, 250, 300, 512 – on which the tests were focused), calculate the unanswerable questions that might have emerged and present those **percentages** below.

Max Context Length	Train Context < Length	Train Questions Unanswerable	Dev Context < Length	Dev Questions Unanswerable
50	1.77	54.01	1.79	54.56
100	12.26	23.97	10.98	24.72
150	53.36	8.08	50.40	8.52
200	79.48	2.65	78.16	3.07
256	92.30	0.78	91.45	1.12
300	96.53	0.35	95.86	0.55
512	99.87	0.01	99.54	0.07

As it is derived from the data, **256** tokens is an adequate value of the maximum context length parameter. More specifically, it seems that a great percentage of samples are 256 tokens long at most (figure 1), which means that not much information will be lost and not many questions will be cast unanswerable due to loss of the answer start and end points in the respective context.

According to the results, only ~8% of the training samples have context longer than 256 and ~9% of the development samples respectively. Furthermore, only 0.78% of the training and 1.12% of the dev questions were cast unanswerable (rather insignificant sum).

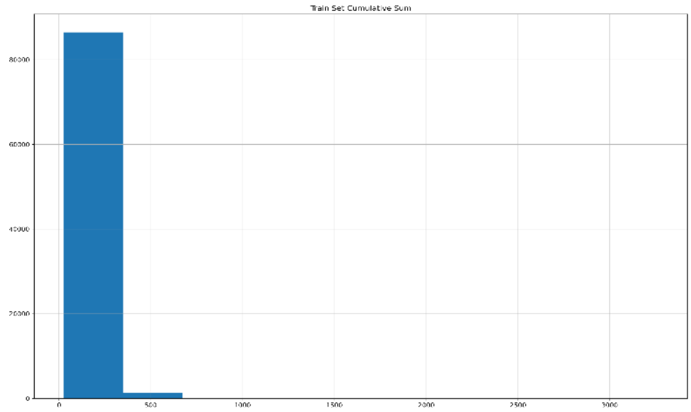

Figure 1. Cumulative Sum of samples under context length

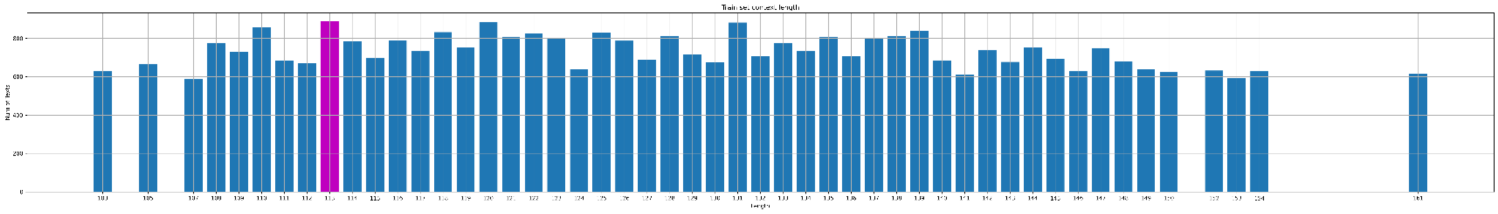

Figure 2. 50 most populated length values

Additionally, in some cases it was imperative that the total **training sample number** was reduced as well, due to computational restraints.

BERT

The most important parameters chosen for the BERT fine-tuning were chosen as follows:

1.	Epochs	→	5	
2.	Optimizer	→	AdamW	(optimized Adam with weight decay)
3.	Learning Rate	→	0.00005	
4.	Weight Decay	→	0.999	(overfitting confinement)
5.	Batch Size	→	32	

Results

BERT was fine-tuned using the training set of SQuADv1.1. After the training process completion, evaluation was carried out using training, dev and user input data (answering questions 2 & 3). In Question 3, 200 random questions were selected 5 times as well as 500, 1000, 10000 and the total of dev dataset (1 time), the mean values of which experiments are noted in the score table. The results are shown below :

Scores (mean %)	200 Train Qs	200 Dev Qs	500 Dev Qs	1K Dev Qs	10K Dev Qs	Total Dev Set
F1	**77.27**	**71.36**	65.74	75.98	69.76	**69.79**
EM	**63.50**	**57.10**	49.00	66.38	55.13	**55.20**

As it was expected, evaluation on unseen data (dev set), resulted in lower scores. However, the difference observed between the dev and training scores (200 question experiment) is very small (~ 6%), indicating a successfully fine-tuned BERT model on this specific task (question-answering).

This confirms the appropriateness of the **dataset**, the data **preprocessing** methods performed as well as the **parameters** chosen for the model training. Slight modifications (such as answer format configurations and special case handling (spaces between ranges e.g. 20-23 vs 20 – 23, different decimal number format e.g. 3,14 vs 3.14 , etc.) might have optimized the model's performance further, which can be left as future work.

Notes

1. **Logfiles** are included in the submitted file, where information about the model fine-tuning, evaluation results (question 3) and user input QnA s (question 2) are stored.
2. **Data visualization** figures and **statistics** were also produced and included in the submission (such as the figures presented in Data Exploration section, stored in better quality).