Spelling Correction Evaluation

Token Classification (xlm-roberta-base)		Note	Dataset	Error
Baseline	Fine tune	0: Normal	Total: 100.000 sentences	error rate for each error: 0.015
Loss: 0.8915	Loss: 0.0257	1: Error	Train: 90.000 sentences	replace_char_in_word
F1: 0.2641	F1: 0.9738	TP (True Positive): Prediction 1 and Ground Truth 1	Test: 10.000 sentences	upper_case_word
Precision: 0.1521	Precision: 0.9877	FP (True Positive): Prediction 1 and Ground Truth 0	Sentence_length: 15 ~ 50 tokens	insert_char_in_word
Recall: 1.0000	Recall: 0.9602	FN (True Positive): Prediction 0 and Ground Truth 1		delete_char_in_word
				swap_char_in_word
Masked LM (xlm-roberta-base)				
Baseline	Fine tune			
Acc1: 0.5789	Acc1: 0.6005			
Acc5: 0.8056	Acc5: 0.8347			
Acc10: 8664	Acc10: 0.8909			
Acc20: 0.9156	Acc20: 0.9308			
The objective of task MLM is to maximize the Acc@10 metric.				
Infer Pipeline				

`If model_confidence > confidence_threshold:`
`  Return Top-1 candidate`
 	
`Else If len(word) < word_length_threshold:`
`  Return candidate with highest Trigram Score`
 	
`Else:`
`  For each candidate:`
`  trigram_s = Normalize(Trigram Score)`
`  lev_s = 1 - Levenshtein Distance / Max Length`
`  combined_s = trigram_weight * trigram_s + levenshtein_weight * lev_s`
 	
`  Return candidate with highest combined_s`
	confidence_threshold = 0.8
	word_length_threshold = 5 words
	trigram_weight = 0.6
	levenshtein_weight = 0.4			