**ASSESSMENT FOR DATA SCIENCE**

**Name: Sajan Poudel**

**[Sajanpoudel32@gmail.com](mailto:Sajanpoudel32@gmail.com)**

**May 28, 2021**

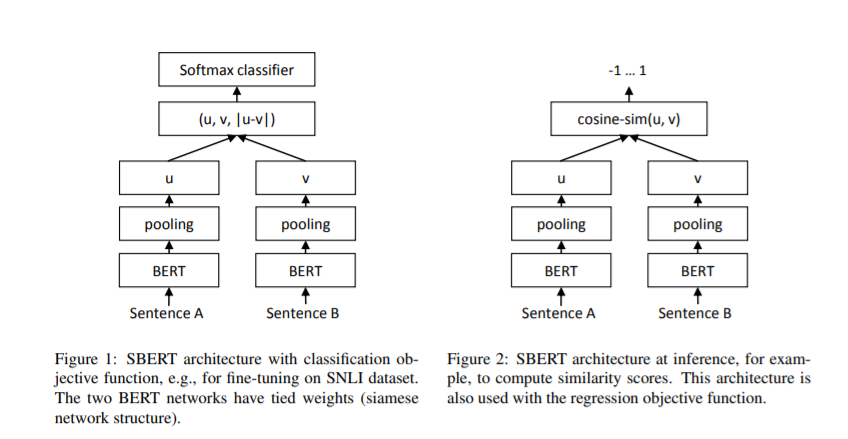
**Task- Finding Semantic Textual Similarity**

**PROBLEM STATEMENT**

Given two paragraphs, quantify the degree of similarity between the two text-based on Semantic similarity. Semantic Textual Similarity (STS) assesses the degree to which two sentences are semantically equivalent to each other. The STS task is motivated by the observation that accurately modelling the meaning similarity of sentences is a foundational language understanding problem relevant to numerous applications including machine translation (MT), summarization, generation, question-answering (QA), short answer grading, semantic search. STS is the assessment of pairs of sentences according to their degree of semantic similarity. The task involves producing real-valued similarity scores for sentence pairs.

**My approach:**

There have been many traditional and modern approaches for semantic textual similarity. After “Attention Is All You Need” paper(original paper of transformer) Transformer had optimized the whole nlp field. So In my approach even I have used the Transformer Architecture. As of my research I found Sentence transformer. ( <https://arxiv.org/pdf/1908.10084.pdf> ). Sentence-BERT (SBERT), a modification of the pretrained BERT network that use siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This reduces the effort for finding the most similar pair from 65 hours with BERT / RoBERTa to about 5 seconds with SBERT, while maintaining the accuracy from BERT.



As we feed sentence separately in model so comparatively this will be computationally cheap. As creating bert model and training takes a long time and takes good computational power so I have imported pretrained model from hugging face. Its simple one line installation. I install sentence-transformers using “pip install sentence-transformers”. This library uses HuggingFace’s transformers behind the scenes. “BERT pretraining, which includes a careful evaluation of the effects of hyperparmeter tuning and training set size. BERT was significantly undertrained and propose an improved recipe for training BERT models, which we call RoBERTa, that can match or exceed the performance of all of the post-BERT methods. Modifications are simple, they include: training the model longer, with bigger batches, over more data;

removing the next sentence prediction objective;

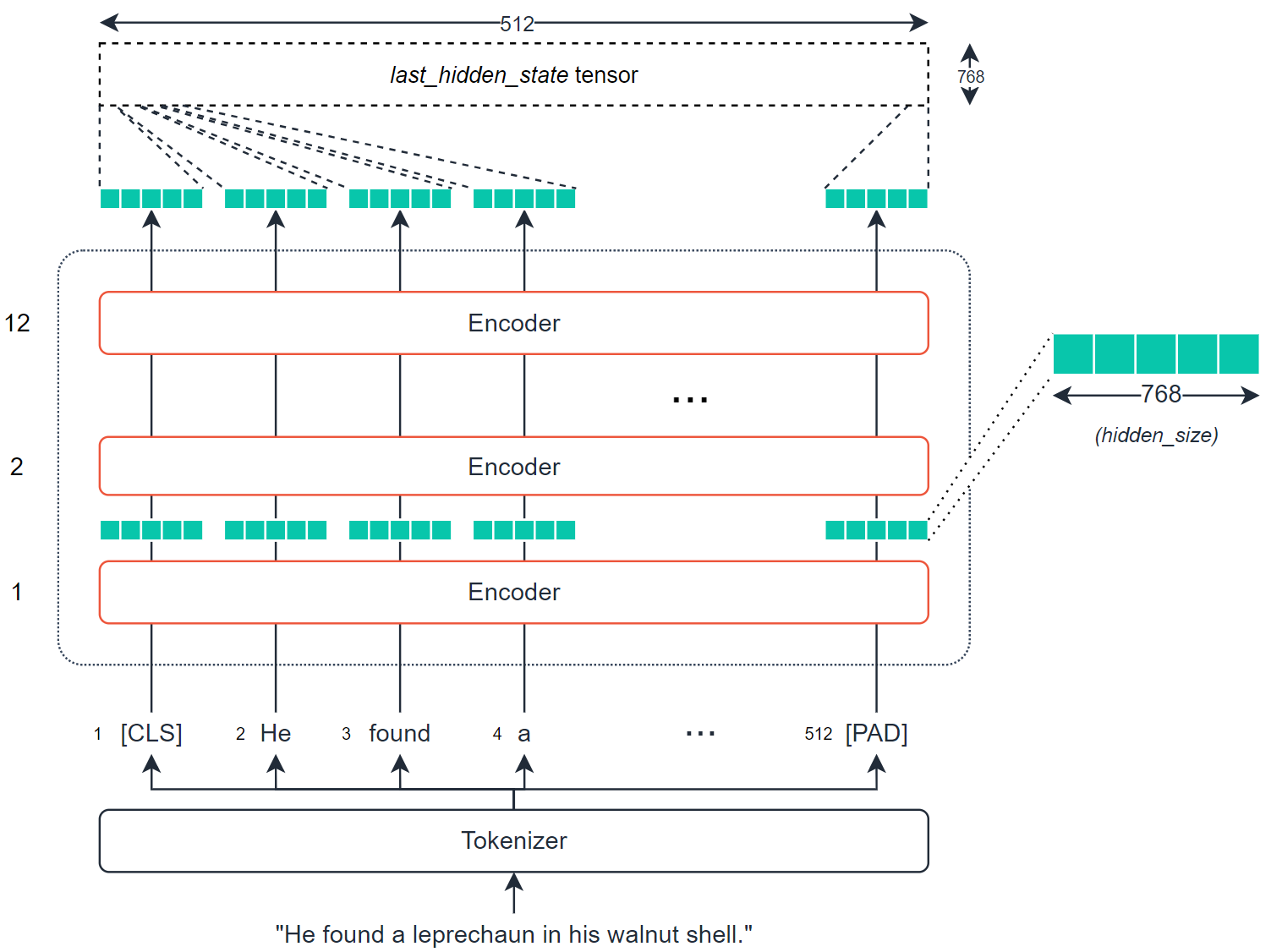
training on longer sequences;

and dynamically changing the masking pattern applied to the training data.

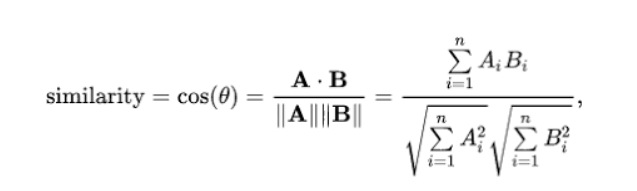
RoBERTa for Robustly optimized BERT approach. Specifically, RoBERTa is trained with dynamic masking, FULL SENTENCES without NSP loss, large mini-batches and a larger byte-level BPE.”

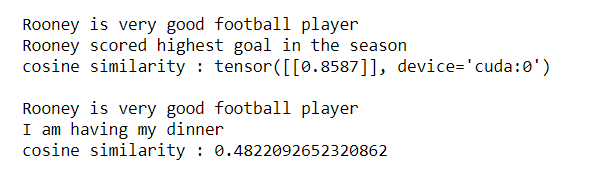
( <https://arxiv.org/pdf/1907.11692.pdf> )

RoBERTa model was trained on two datasets (NLI+STSb). So we assume we can great summarize our paragraphs to convert it to relevant embeddings.



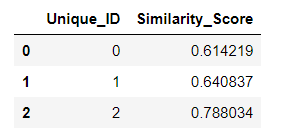
The output from the BERT model is converted to vector using mean pooling operation. This pooling operation will take the mean of all token embeddings and compress them into a single 768 vector space. Now the final embedding vetor is compared with each other for finding the similarity using COSINE similarity. Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. The predicted output we re-scale in the range of 0 to 1. 0 for dissimilarity and 1 for similarity.





Testing on some sentence I seems to give a very good result.  
  
From the given datasets the result we get is quite satisfying.

1St 3 result from the datasets :



Thank you !