……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Name:** ABHIMANYU YADAV

**Email address:** [abhimanyuyadavnsit@gmail.com](mailto:abhimanyuyadavnsit@gmail.com)

**Contact number:** 8860088322

**Anydesk address:**

**Years of Work Experience: 1**

**Date: 2nd September 2022**

……………………………………………...……………………………………………...……………………………………………...……………………………………………………………...………………

**Self Case Study -2: Google QUEST Q&A Labeling**

“After you have completed the document, please submit it in the classroom in the pdf format.”

Please check this video before you get started: <https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg>

# Overview

\*\*\* Write an overview of the case study that you are working on. ***(MINIMUM 200 words)*** \*\*\*

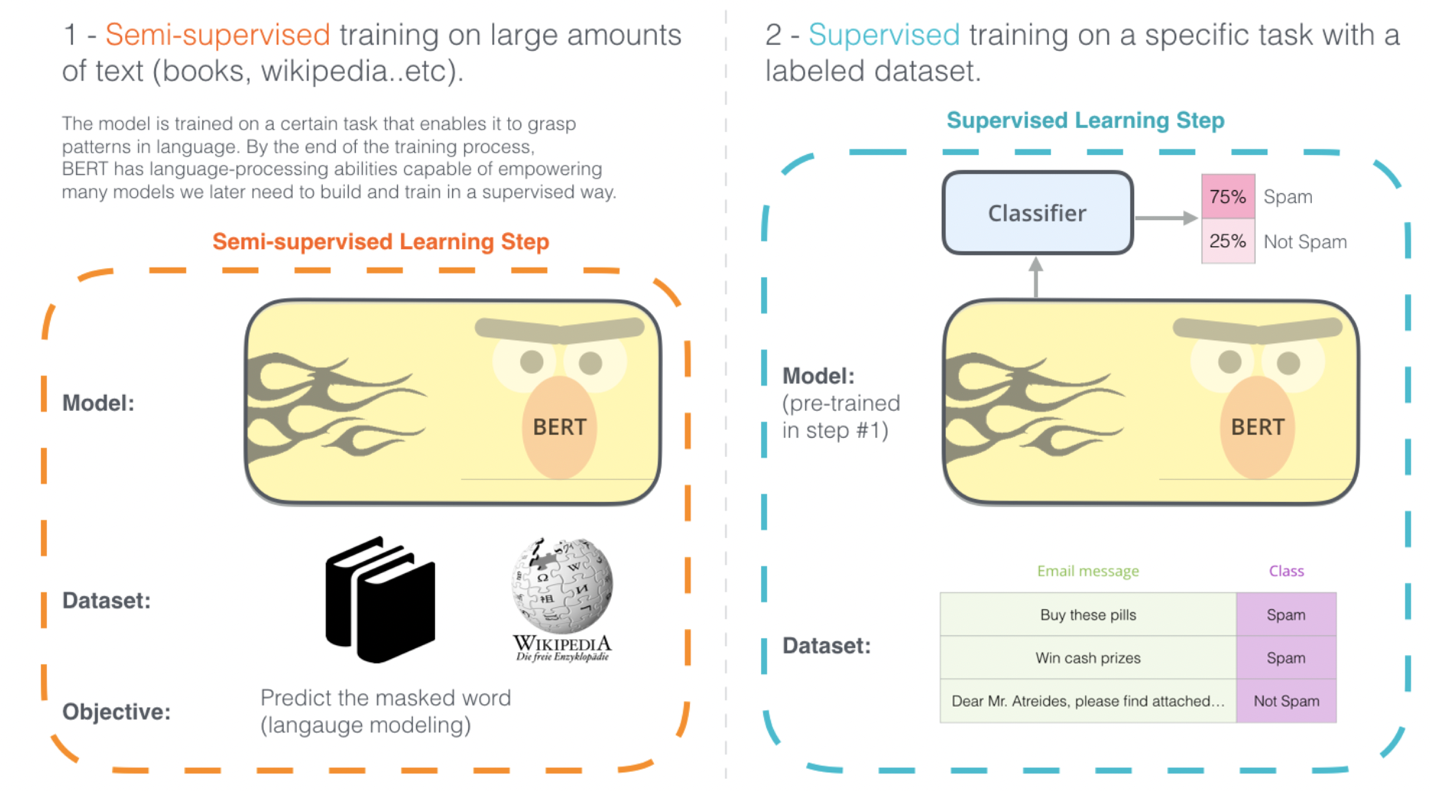
1. Computers are generally considered good at answering questions that are of a factual nature meaning questions for which there exists only single, verifiable answer unlike subjective questions where a deeper, multidimensional understanding of the context is required. Answers to such questions usually are not definitive in nature and changes based on how an individual comprehends it.
2. Google research team has collected data from 70 different websites used raters to answer each of the 30 prompts where 21 prompts are related to questions asked and 9 related to answers corresponding to it.
3. Our task as an ML Engineer is to design a model that can reliably predict each of these 30 different subjective labels.

# 

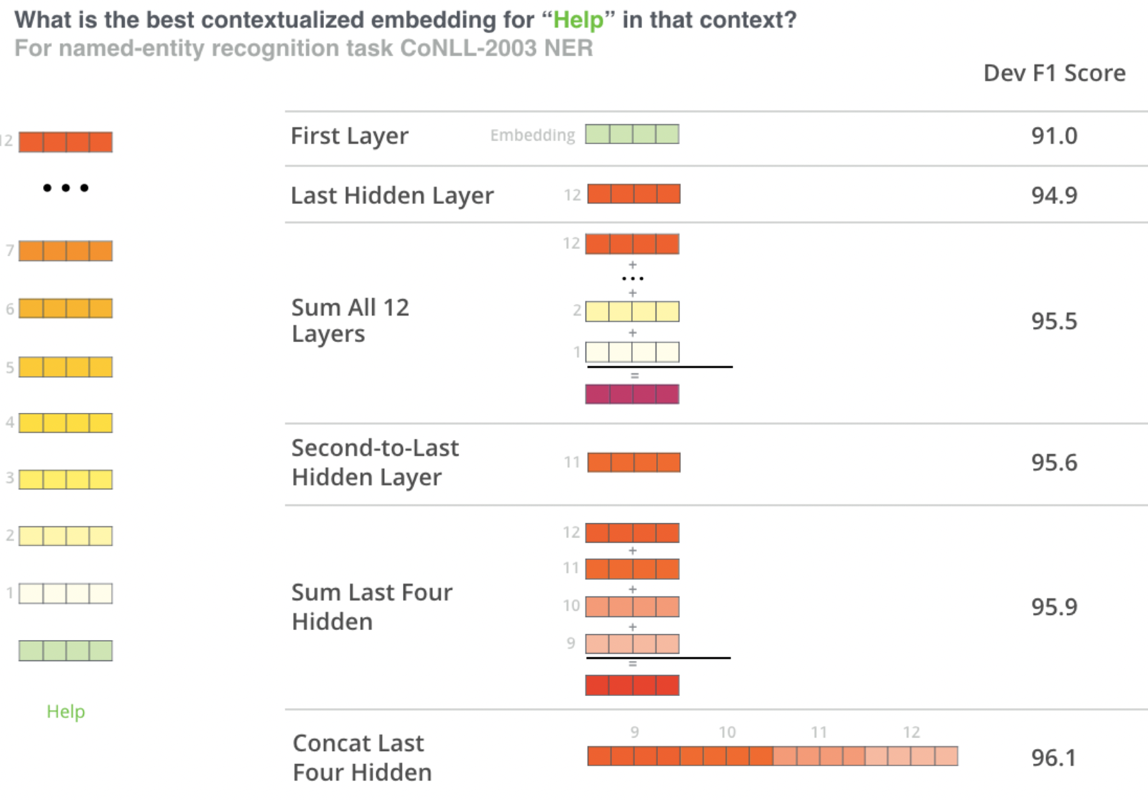
# Research-Papers/Solutions/Architectures/Kernels

\*\*\* Mention the URL’s of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. \*\*\*

1. <https://jalammar.github.io/illustrated-bert/>
   1. BERT is basically a trained Transformer Encoder stack which has been trained using Semi Supervised Learning which can be further fine-tuned in order to achieve specific tasks.



* 1. Model Inputs – It takes as input an input sequence of length not greater than 512 with [CLS] as the first input token.
  2. Model Outputs – Each of the outputs corresponding to each input tokens are 768-dimension vector which depicts the contextualized word embedding. The output corresponding to the [CLS] token is further used as an input for to a FFNN for classification tasks.
  3. Experiments have shown that Embeddings from the last 4 hidden layers when concatenated works as a best contextualized embedding



1. <https://towardsdatascience.com/hands-on-transformers-kaggle-google-quest-q-a-labeling-affd3dad7bcb>
   1. In addition to the first cut approach explained below they have prepared to 2 datasets one that prepares input sequences using question title and body and other using question title , question body and answer.
   2. Since the initial 21 target variables out of 30 are related to questions only. So, they only require question title and body input sequences for predicting these 21 target variables.
   3. So they have created an ensemble of models to make the final predictions.

Diagram

Description automatically generated

# 

# First Cut Approach

\*\*\* Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** \*\*\*

1. Will start with defining the problem statement for a given business problem description followed by defining objectives that we wish to achieve but these objectives must be fulfilled such that certain constraints are taken care of while achieving the objectives. The constraints for the given problem are as follows:
   1. Reliable prediction of these subjective label is desired as it would taking us one step closer to our desire of making feature models more and more human like.
   2. Since each of the prompts being answered takes a value between 0 and 1 and value corresponding to each prompt depicts the chances of presence of that subjective aspect in our question and answer.so, we need to have a model that would give us the probability scores for each of the target labels.
   3. No strict latency requirement is there for this particular problem.
   4. Higher interpretability of the model is desired to understand the decision making process of the model which can help us ascertain the sanity of the model.
2. The second task involves shaping up the business problem into a ML problem. The steps to achieve it are as follows:
   1. Take a basic overview of data which involves what all data have we been provided with, what all files have we been provided with to perform the task, what all information have been provided in those files.
   2. Identifying the type of ML problem here in our case It's a REGRESSION problem where the task is to predict 30 Target Labels for each question answer pair in the given dataset.
   3. Deciding the performance metrics, we will be working with to compare the performance of models we have built and choosing the one which performs the best on the data provided.
   4. Strategy for construction of train, cross validation and test need to be finalized based on the nature of data. Since the data in our case if of NON-TEMPORAL nature we would be sticking to RANDOM SPLITTING.
3. EDA
   1. Looked at what all input variables and target variables have been provided to us in the dataset.
   2. Tried figuring out the number of unique questions we have in the dataset just to have an idea about the amount of repetitiveness of questions in the dataset.
   3. Plotted the distribution of lengths of question title, question body and answer.
   4. Plotted the distribution of questions asked by each user and answer answered by each user.
   5. Plotted pie chart explaining the distribution of questions among various categories.
   6. Plotted the distribution of source from where the questions and answers have been picked up from.
   7. Plotted distribution and correlation map for all the target variables in the given dataset.
4. TEXT PREPROCESSING – performed preprocessing on questions title, question body, and answer by converting the text for all of them into lowercase, removing html tags, performing decontractions, removing all the special characters from each of them.
5. PREPARING DATASET – Because we are using BERT as our model which can work with input sequences that are not more than 512 in length and since our target variables are dependent both on the question and answer so have taken first half of the input sequence coming from question title and question body and the rest of it coming from answer. Corresponding to each input sequence a masking and segment sequence is also prepared.
6. BUILDING MODEL
   1. Once the input sequence has been prepared the same is being fed into BERT to obtain the embedding corresponding to [CLS] token from the last 4 hidden layers. The concatenated embedding thus obtained is further fed into a dense layer with 30 sigmoid activations to obtain the output.
   2. The model is then used to fit on our given dataset using Binary Cross Entropy and Mean Squared Error as loss functions.

Notes when you build your final notebook:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument “X” (a single data points i.e 1\*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
   1. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
   2. so in your final notebook, you need to pass only those two values
   3. def final(X):

preprocess data i.e data cleaning, filling missing values etc

compute features based on this X

use pre trained model

return predicted outputs

final([time, location])

* 1. in the instructions, we have mentioned two functions one with original values and one without it
  2. final([time, location]) # in this function you need to return the predictions, no need to compute the metric
  3. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y\_predict)

1. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data
2. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
3. Check this live session: <https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models>