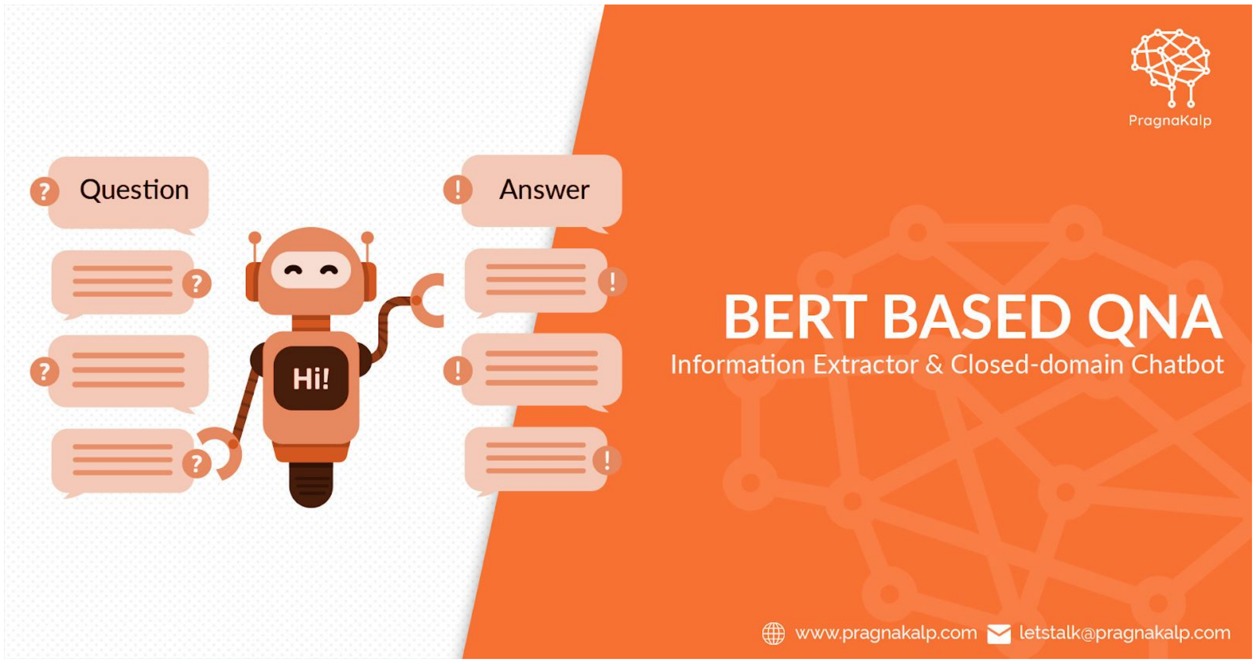
Udacity Machine Learning Nanodegree Capstone Project Report

Google Quest Q&A labelling

By Manav Nitin Kapadnis



Introduction

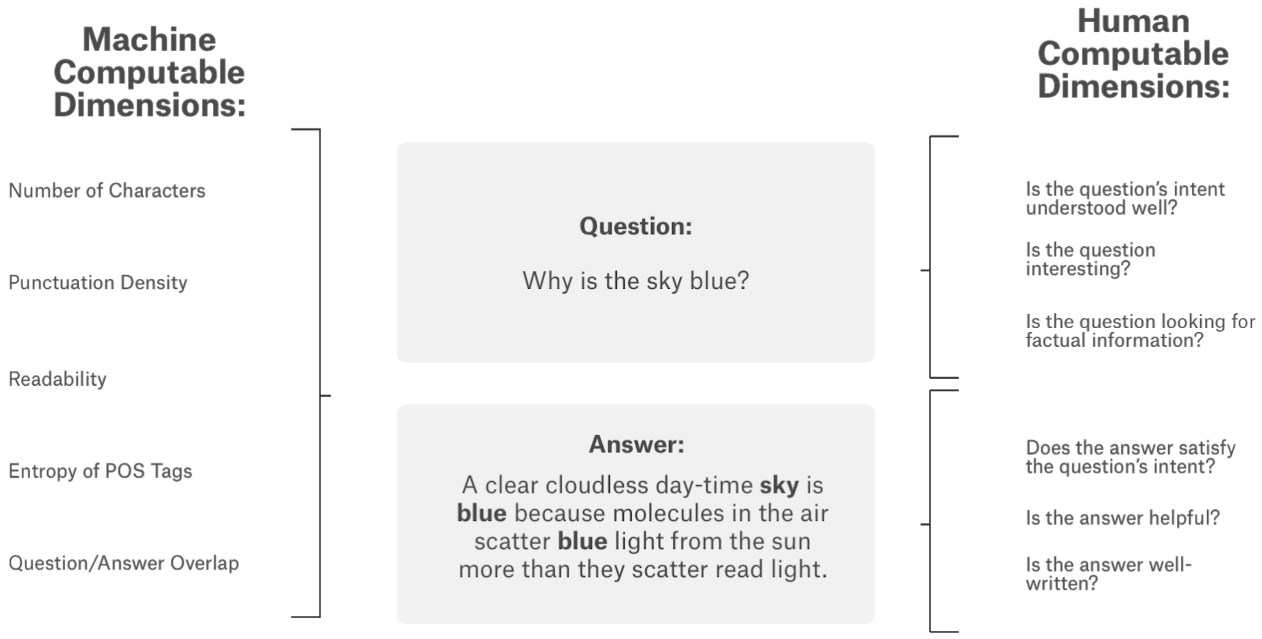
The main aim behind taking this project was to expand my knowledge in the Natural Language Processing Field,mainly in the Question and Answering system,so that I can create more projects such as Chatbots using NLU and Rasa.Moreover,I hadn't touched this field before,but now after completing such a complex project,I feel like I have accomplished something at the end of my Nanodegree. Let's begin the report,it consists of five Parts:

1. **Defining the Problem** - A formal definition and an explanatory one
2. **Analysing the Dataset** - Observations from EDA
3. **Implementation of Bert -** Explaining the functioning of Bert in this task
4. **Explanation Of Evaluation Metric -** It's a relatively complex metric so requires an explanation
5. **Results and Conclusion -** Final outcome of the report

Defining The Problem

Computers are really good at answering questions with single, verifiable answers. But, humans are often still better at answering questions about opinions, recommendations, or personal experiences.

Humans are better at addressing subjective questions that require a deeper, multidimensional understanding of context - something computers aren't trained to do well...yet.. Questions can take many forms - some have multi-sentence elaborations, others may be simple curiosity or a fully developed problem. They can have multiple intents, or seek advice and opinions. Some may be helpful and others interesting. Some are simple right or wrong.



Machine Computable Dimensions:
Number of Characters
Punctuation Density
Readability
Entropy of POS Tags
Question/Answer Overlap
Question:
Why is the sky blue?
Answer:
A clear cloudless day-time sky is blue because molecules in the air scatter blue light from the sun more than they scatter read light.
Human Computable Dimensions:
Is the question's intent understood well?
Is the question interesting?
Is the question looking for factual information?
Does the answer satisfy the question's intent?
Is the answer helpful?
Is the answer well-written?

Unfortunately, it's hard to build better subjective question-answering algorithms because of a lack of data and predictive models. That's why the CrowdSource team at Google Research, a group dedicated to advancing NLP and other types of ML science via crowdsourcing, has collected data on a number of these quality scoring aspects.

In this dataset, you're challenged to use this new dataset to build predictive algorithms for different subjective aspects of question-answering. The question-answer pairs were gathered from nearly 70 different websites, in a "common-sense" fashion.

The what you read was a formal explanation of the problem,roughly the problem translates to this,you are given a question title,a question ,an answer,the source of answer,and a question id,which was not useful and so was dropped while preprocessing.On the basis of these information,you have to predict the probabilities of thirty different labels,which are namely -

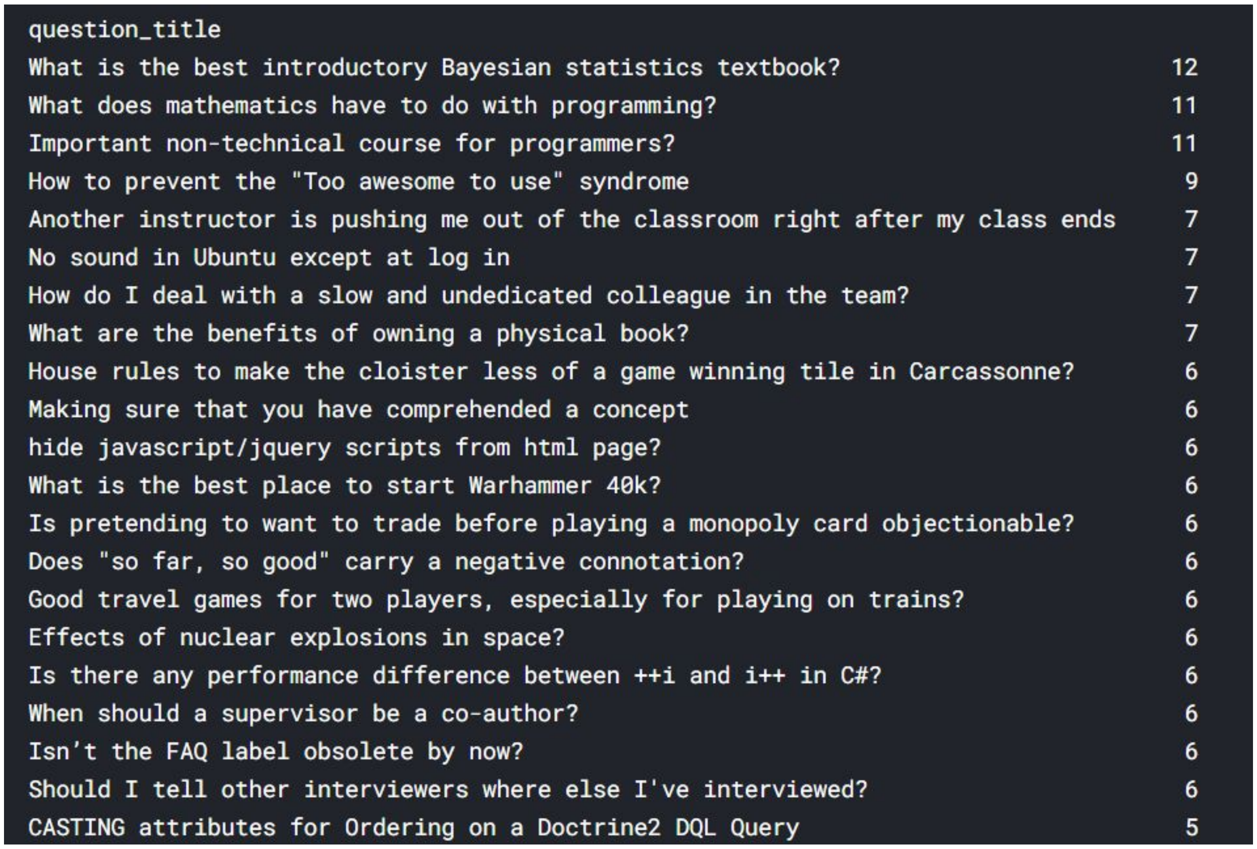
```
output categories:
         ['question_asker_intent_understanding', 'question_body_critical', 'question_co
nversational', 'question_expect_short_answer', 'question_fact_seeking', 'question_has_c
ommonly_accepted_answer', 'question_interestingness_others', 'question_interestingness_
self', 'question_multi_intent', 'question_not_really_a_question', 'question_opinion_see
king', 'question_type_choice', 'question_type_compare', 'question_type_consequence', 'q
uestion_type_definition', 'question_type_entity', 'question_type_instructions', 'questi
on_type_procedure', 'question_type_reason_explanation', 'question_type_spelling', 'ques
tion_well_written', 'answer_helpful', 'answer_level_of_information', 'answer_plausibl
e', 'answer_relevance', 'answer_satisfaction', 'answer_type_instructions', 'answer_type
_procedure', 'answer_type_reason_explanation', 'answer_well_written']
```

This could be easily(not so easily) implemented by the help of Bidirectional Encoder Representations from Transformers or BERT .

Analysing the Dataset

Some of the Observations made by analyzing the Data are as follows:

- The train and test size are fairly small enough,(which was one of the reasons why the competition organizers allowed the use of external data)
- The dataset contained zero missing values,which was very impressive
- The train as well as test dataset hosts were mainly dominated by two websites namely, stackoverflow and stackexchange
- The questions were mainly categorised into five different categories namely Technology , stackoverflow,culture science and life arts.
- There were a lot of duplicate questions,their order is given in the picture below:



```
question_title
What is the best introductory Bayesian statistics textbook?                        12
What does mathematics have to do with programming?                                 11
Important non-technical course for programmers?                                    11
How to prevent the "Too awesome to use" syndrome                                    9
Another instructor is pushing me out of the classroom right after my class ends     7
No sound in Ubuntu except at log in                                                 7
How do I deal with a slow and undedicated colleague in the team?                    7
What are the benefits of owning a physical book?                                    7
House rules to make the cloister less of a game winning tile in Carcassonne?        6
Making sure that you have comprehended a concept                                    6
hide javascript/jquery scripts from html page?                                      6
What is the best place to start Warhammer 40k?                                      6
Is pretending to want to trade before playing a monopoly card objectionable?        6
Does "so far, so good" carry a negative connotation?                                6
Good travel games for two players, especially for playing on trains?                6
Effects of nuclear explosions in space?                                             6
Is there any performance difference between ++i and i++ in C#?                      6
When should a supervisor be a co-author?                                            6
Isn't the FAQ label obsolete by now?                                                6
Should I tell other interviewers where else I've interviewed?                       6
CASTING attributes for Ordering on a Doctrine2 DQL Query                            5
```

- There were a lot of repeating differently ordered questions too,those can be seen in the EDA notebook named as Quest EDA.

Implementation of BERT

The implementation of Bert for this task was similar to that for a normal text classification,but here instead of the three,inputs,mask and segment vectors,the BERT had to be given six inputs mainly two sets of (input, mask and segment) ,one for question and one for answer text.

BERT can handle a maximum sequence length of 512,but we only required a max len of 384 numbers of vector representation for a single input.

Now,in order to create model,a bert-base-uncased model was used for the tokenizer as well as for the the pre trained model,which was later fine tuned by changing the final layer to having 30 outputs with a softmax activation,apart from these Dropout of 20% and two more layers of GLobal average pooling 1 D were also used in the fine tuning of model.

The cross validation splits was of five folds but was trained on only three of them to save time.

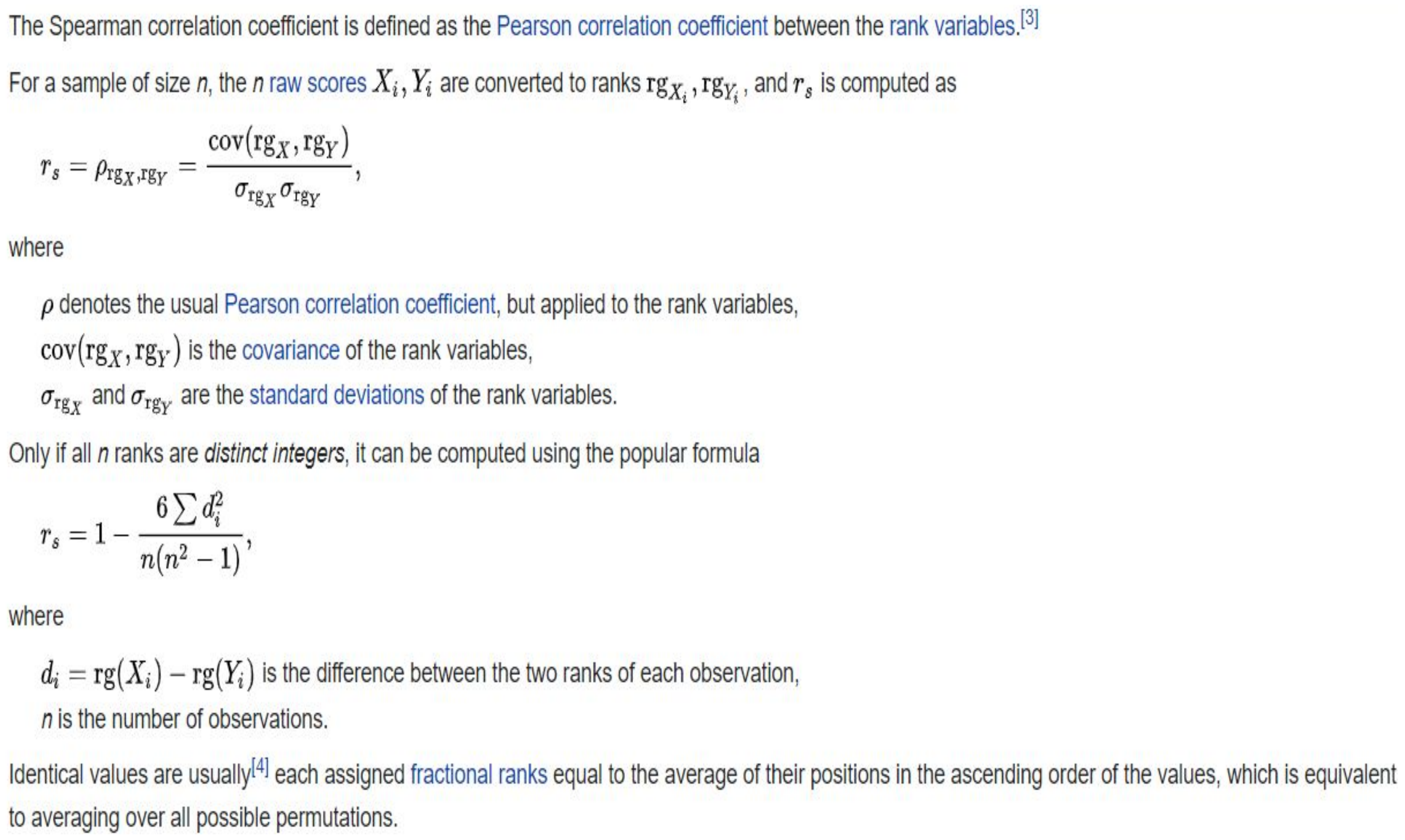
The optimizer used was Adam optimizer with a learning rate of 2*10^(-5).The model was trained for two epochs,even then the training time was 1 hr,which was pretty big for a beginner like me.

Explanation of Evaluation Metric

Submissions were evaluated on the mean column-wise Spearman's correlation coefficient. The Spearman's rank correlation is computed for each target column, and the mean of these values is calculated for the submission score.

Spearman's Correlation between two variables is equal to the Pearson correlation between the rank values of those two variables; while Pearson's correlation assesses linear relationships, Spearman's correlation assesses monotonic relationships (whether linear or not). If there are no repeated data values, a perfect Spearman correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other.

Intuitively, the Spearman correlation between two variables will be high when observations have a similar (or identical for a correlation of 1) rank (i.e. relative position label of the observations within the variable: 1st, 2nd, 3rd, etc.) between the two variables, and low when observations have a dissimilar (or fully opposed for a correlation of −1) rank between the two variables.



The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the rank variables.[3]

For a sample of size *n*, the *n* raw scores X_i, Y_i are converted to ranks $\mathrm{rg}_{X_i}, \mathrm{rg}_{Y_i}$, and r_s is computed as

$$r_s = \rho_{\mathrm{rg}_X, \mathrm{rg}_Y} = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X} \sigma_{\mathrm{rg}_Y}},$$

where

ρ denotes the usual Pearson correlation coefficient, but applied to the rank variables,
$\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)$ is the covariance of the rank variables,
σ_{rg_X} and σ_{rg_Y} are the standard deviations of the rank variables.

Only if all *n* ranks are *distinct integers*, it can be computed using the popular formula

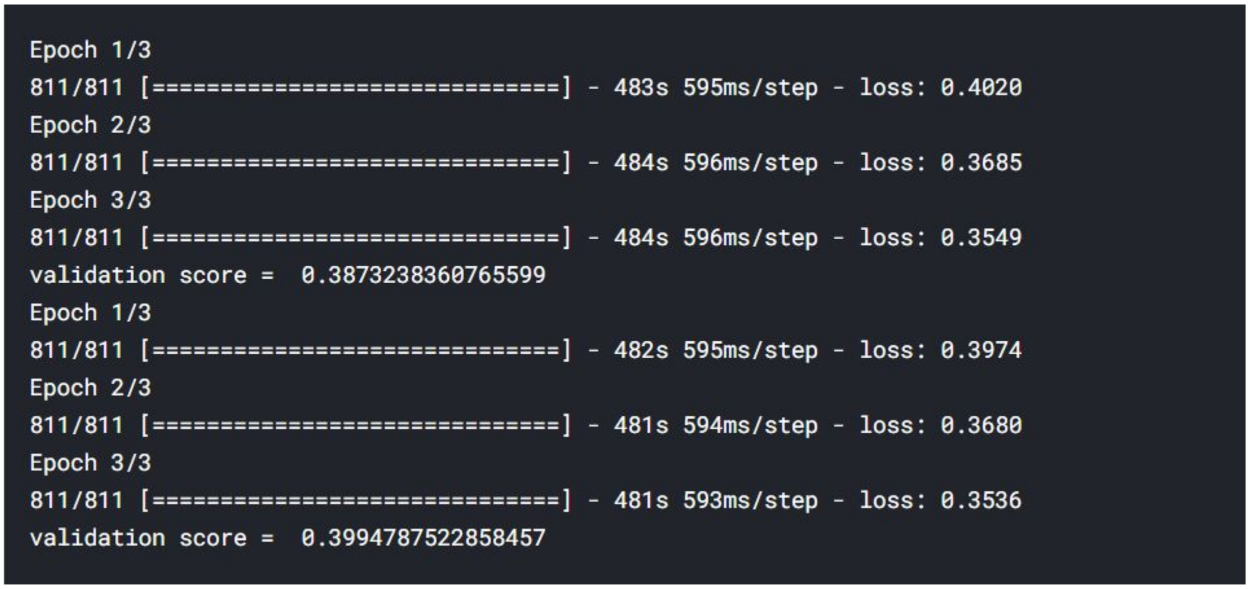
$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)},$$

where

$d_i = \mathrm{rg}(X_i) - \mathrm{rg}(Y_i)$ is the difference between the two ranks of each observation,
n is the number of observations.

Identical values are usually[4] each assigned fractional ranks equal to the average of their positions in the ascending order of the values, which is equivalent to averaging over all possible permutations.

Results and Conclusion



```
Epoch 1/3
811/811 [==============================] - 483s 595ms/step - loss: 0.4020
Epoch 2/3
811/811 [==============================] - 484s 596ms/step - loss: 0.3685
Epoch 3/3
811/811 [==============================] - 484s 596ms/step - loss: 0.3549
validation score =  0.3873238360765599
Epoch 1/3
811/811 [==============================] - 482s 595ms/step - loss: 0.3974
Epoch 2/3
811/811 [==============================] - 481s 594ms/step - loss: 0.3680
Epoch 3/3
811/811 [==============================] - 481s 593ms/step - loss: 0.3536
validation score =  0.3994787522858457
```

The picture you see above is the the validation score after running the code for two epochs and for three cross validation score each.In the output the score shown is 0.399 in the validation set,but while submission I guess the score might be between 0.39 to 0.41, I forgot to submit my predictions,because here in this competition you have to submit via the kernel itself rather than manual submission and I forgot to do that and closed my kernel,only to realise later the submission wasn't done,but no problem,as this competition was already over ,I guess there was no loss in my mistake.I also wanted to implement RoberTa and DistillBert,but due to college assignments and homeworks,I got time only for BERT Implementation.There was no point in training for more than 2 epochs since the bert already has such complex calculations,running for more than two epochs would lead to overfitting and that would lead to a low test score

In Conclusion,I would like to say that the score can be increased a bit more by using a different model namely BART or AlBert,this was seen in the first position solution.I will try this in future.I would like to thank Udacity for giving me this opportunity to learn and implement a full fledged project on NLP on my own,which has increased my knowledge in this sector many folds.Thankyou Udacity and Amazon for this Nanodegree Program!!!