Function will take two lists of words as argument;

list\_of\_words\_1 = [‘ask, ‘everybody]

list\_of\_words\_2 = [‘full’, ‘complement’, ‘not’, ‘quite’]

Parallel process to list of word in a way that;

Processing list\_of\_words\_1;

* Find the list of words in the JSON (sample.json) provided
* If words are found in JSON
  + Find if words if the list # 1occurred in sequence and in specific time interval
  + If words are present in JSON in sequence check if they are within 10 seconds interval
  + If words are within 10 seconds interval; capture the “offset of the last word” and convert them into seconds by dividing the offset value with 10,000,000

{

"Word": "ask",

"Offset": 90500000,

"Duration": 3500000

},

{

"Word": "everybody",

"Offset": 94000000,

"Duration": 4400000

},

Processing list\_of\_words\_2;

* Find the list of words in the JSON (sample.json) provided
* If words are found in JSON
  + Find if words if the list # 1occurred in sequence and in specific time interval
  + If words are present in JSON in sequence check if they are within 10 seconds interval
  + If words are within 10 seconds interval; capture the “offset of the last word” and convert them into seconds by dividing the offset value with 10,000,000

full: 307200000/10,000,000 = 30.72 seconds

complement: 311800000/10,000,000 =31.18 seconds

Duration of “full” = 4600000/ 10,000,000 = 0.46 seconds

{

"Word": "full",

"Offset": 307200000,

"Duration": 4600000

},

{

"Word": "complement",

"Offset": 311800000,

"Duration": 6600000

},

{

"Word": "not",

"Offset": 318400000,

"Duration": 3000000

},

{

"Word": "quite",

"Offset": 321400000,

"Duration": 5300000

Output of function will be:

Final out of the function will be calculated duration from list\_of\_words\_1 to list\_of\_words\_2;

Offset of last word in the list\_of\_words\_2 (minus) Offset of last word in the list\_of\_words\_1

}