**CS372 Natural Language Processing with Python**

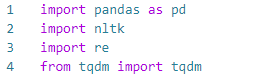
**Homework 3**

Python version : 3.8.10

Nltk version : 3.5

This homework is kind of open-ended problem. So instead of writing the final result of my program, I will show how I improved my model, step by step. There are some ideas that turned out to be a failure, and some of successful ideas remained to the next level.

For external modules, I imported pandas, to handle tsv file easily, tqdm, to estimate the remaining running time of the program, and nltk.



I didn’t use url link at all, so snippet output and page output will be same.

|  |  |
| --- | --- |
| **Design 0 tf(former word)** | 2 |
| **Design 1 closer word** | 3 |
| **Design 2 Subject pronoun** | 4 |
| **Design 3 Specific cases** | 5 |
| **Design 3.1 parenthesis** | 5 |
| **Design 3.2 word coming after pronoun** | 6 |
| **Design 3.3 always true** | 7 |
| **Design 4 both false cases** | 8 |
| **Design 5 Subject word** | **10** |
| **Design 5-1 Only subject word** | **10** |
| **Design 5-2 Both subject word** | **11** |
| **Design 6 Word appearance** | **12** |
| **Design 6-1 More appearance** | **13** |
| **Design 6-2 Both most appearance** | **14** |
| **Design 7 begin of sentence** | **15** |
| **Design 8 Subject(Final model)** | **16** |
| **How to improve** | **18** |
| **Appendix** | **19** |

**Design 0 – tf(former word)**

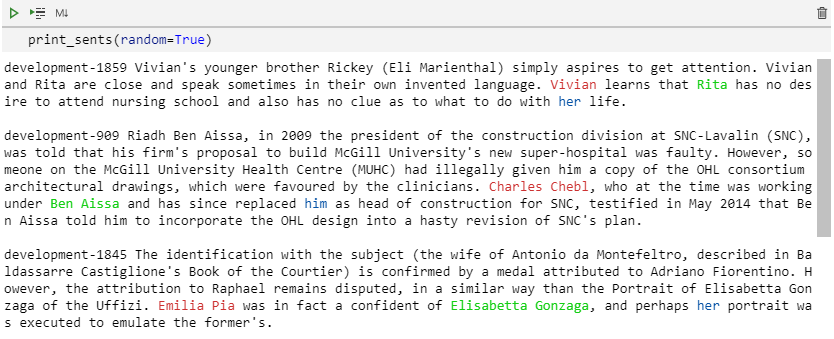
For two given words, I will always predict first word as True, and second word as False.



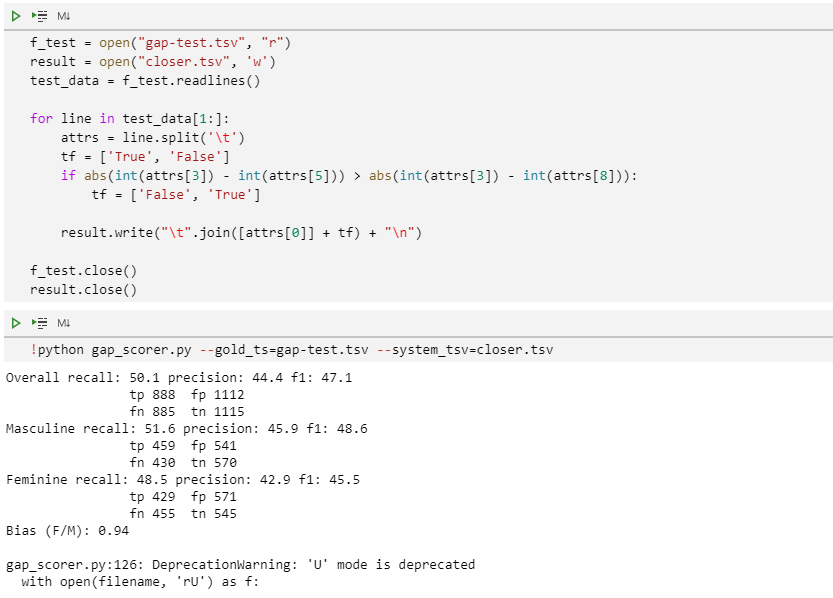
Overall f1-score is 48.7, which means almost nothing for such a binary classification problem.

**Design 1 – closer word**

In order to find any characteristics of the sentences, I had to take a closer look at them. But I found it too difficult to read them with just ‘offset’. So I made a program that prints the sentences with colors so that I could identify which one is the target word/pronoun.



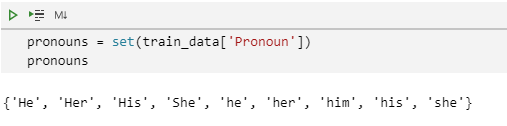
What I realized from doing this was that the word that appears closer with the pronoun are more likely to be true. So I used the idea to my model.



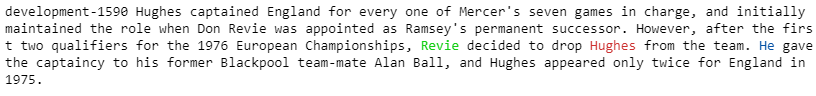
But unfortunately, f1-score was even lower.

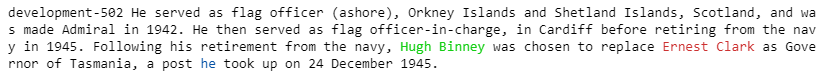
**Design 2 – Subject pronoun**

I learned that there are three different kinds of pronoun : subject pronoun(he, she), object pronoun(him, her), possessive pronoun(his, her). And they showed different characteristics of different kinds.



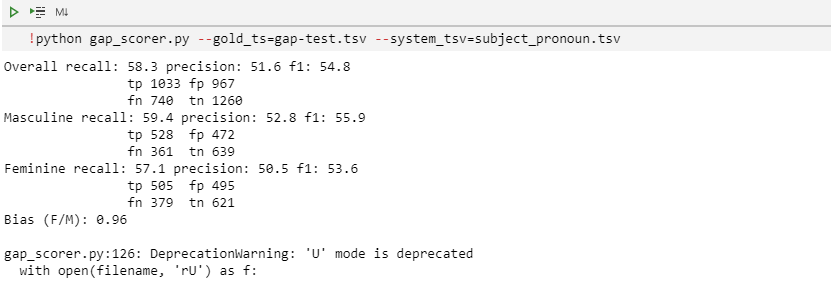
For example, unlike object and possessive pronouns, subject pronouns are more likely to refer farther word.





So I divided pronouns into two groups : subject pronouns & others. My second model predicts farther word as true, and closer word as false when the pronoun is subject pronoun, and do the opposite when the pronoun is object or possessive pronoun.





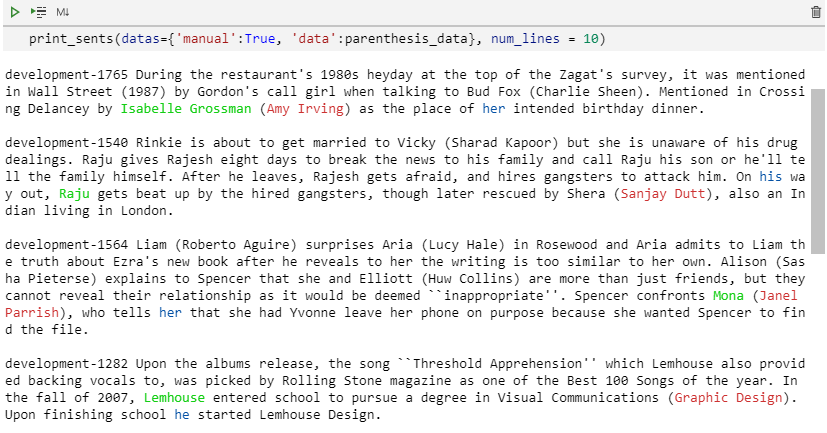
Now the f1-score is higher than Design0 and 1.

**Design 3 Specific cases**

In design 3, I tried various ideas from observation without thinking deeply.

**Design 3.1 – parenthesis**

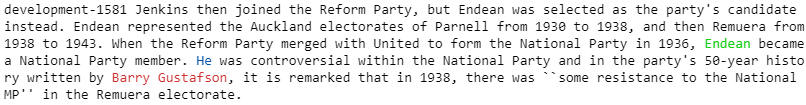
I found some words are inside the parenthesis, whose coreference in most of the cases were false.

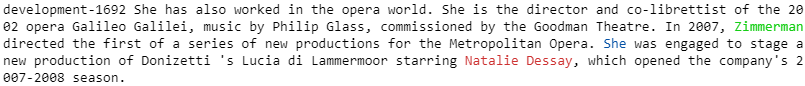


When I predict the word inside the parenthesis as false, 22 out of 31 sentences were predicted correct, which means accuracy is 0.71.(pretty good)

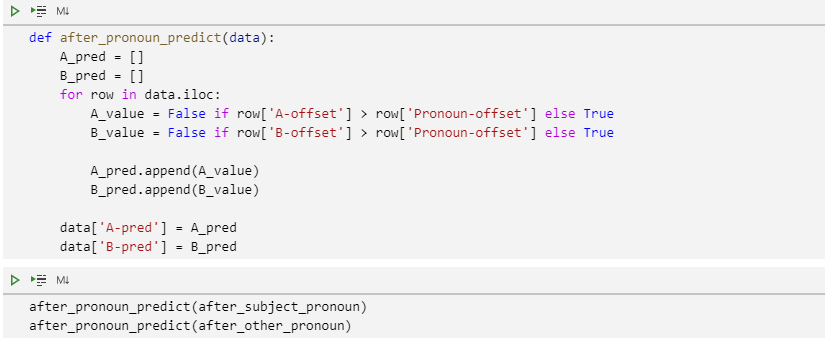
**Design 3.2 – word coming after pronoun**

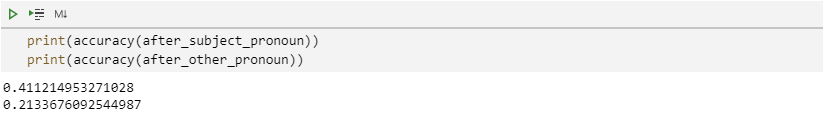
Sometimes one of the words comes after the pronoun.





And the word coming after the pronoun is more likely to be false. I tested this idea in two divided datas: when the pronoun is subject pronoun & others.

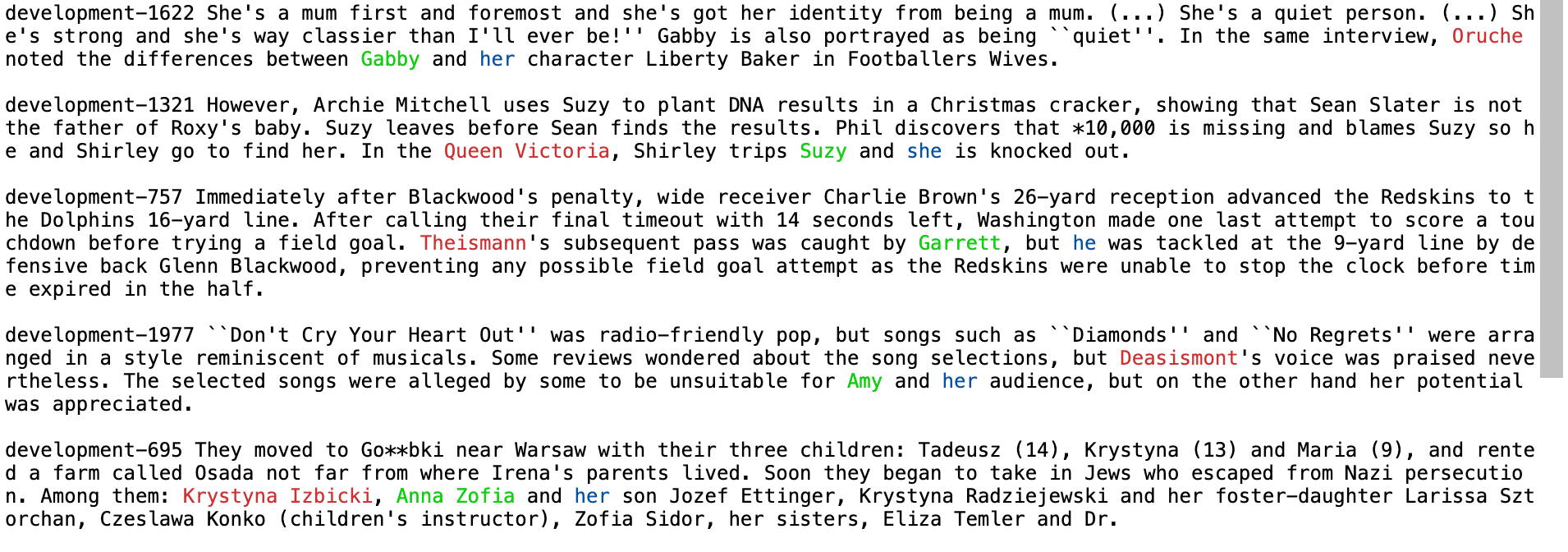




But the accuracy wasn’t so good. I think the hypothesis I made was too general.

**Design 3.3 – always true**

There are patterns that are ‘always true’. For example, patterns like <(word) and (pronoun) …>, or <(born) (word)> is always true. There are not many cases satisfying this condition, however, it will help to raise the accuracy of total cases because their accuracy is almost 1. This time, I only tried <(word) and (his|her)…> pattern, but it will be good to find another patterns that are always true.

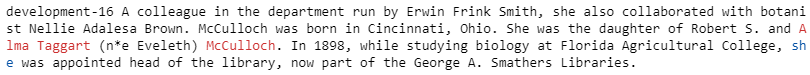


**Design 4 – both false cases**

I realized that there are 200 cases, which is 10% of total cases that the both coreference of words are false.



While watching ‘both false cases’, I found some characteristics they have in common. First, another pronoun already appears before the words.

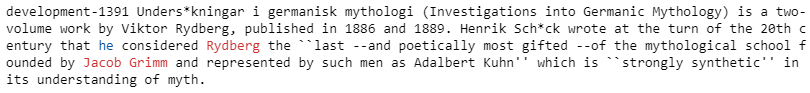


These pronouns are likely to refer to the same words as each other, while most of the words that are referenced appear much earlier. So when another pronoun in same gender appears before the words, I predicted them as both false.

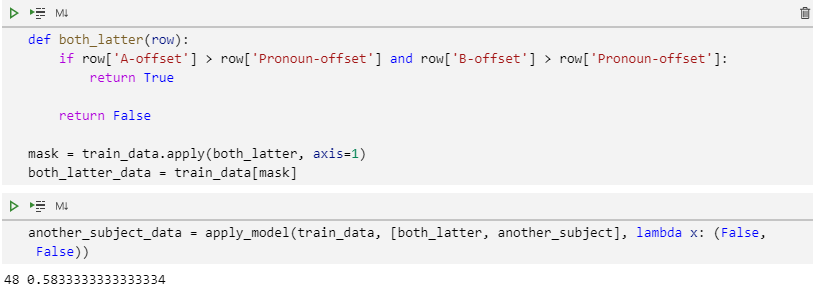


However, the performance was really bad.

Second idea is that when the words both come after the pronoun, and there is another subject exist in the text, predict the both coreference of the words false. (Another subject means that proper noun that comes in the beginning of the sentence)







**\*apply\_model() : apply the conditions with condition functions to given data, and predict with prediction function(Appendix)**

The accuracy is not bad, but size of total cases is too small(48 out of 2000 cases).

**Design 5 Subject word**

I defined subject word like this : Subject word is a word that comes right before verbs, modals, or adverbs.



**Design 5-1 Only subject word**

When one of two words are subject word, and other is not, I will predict the subject word as true, and other is false.





The accuracy of my model that meets the condition(only subject), was 0.75(Accuracy0 in the figure above). Interesting thing is that for objective pronoun ‘him’, accuracy of the model was lower than simple prediction model(latter word is true, Acuracy1).

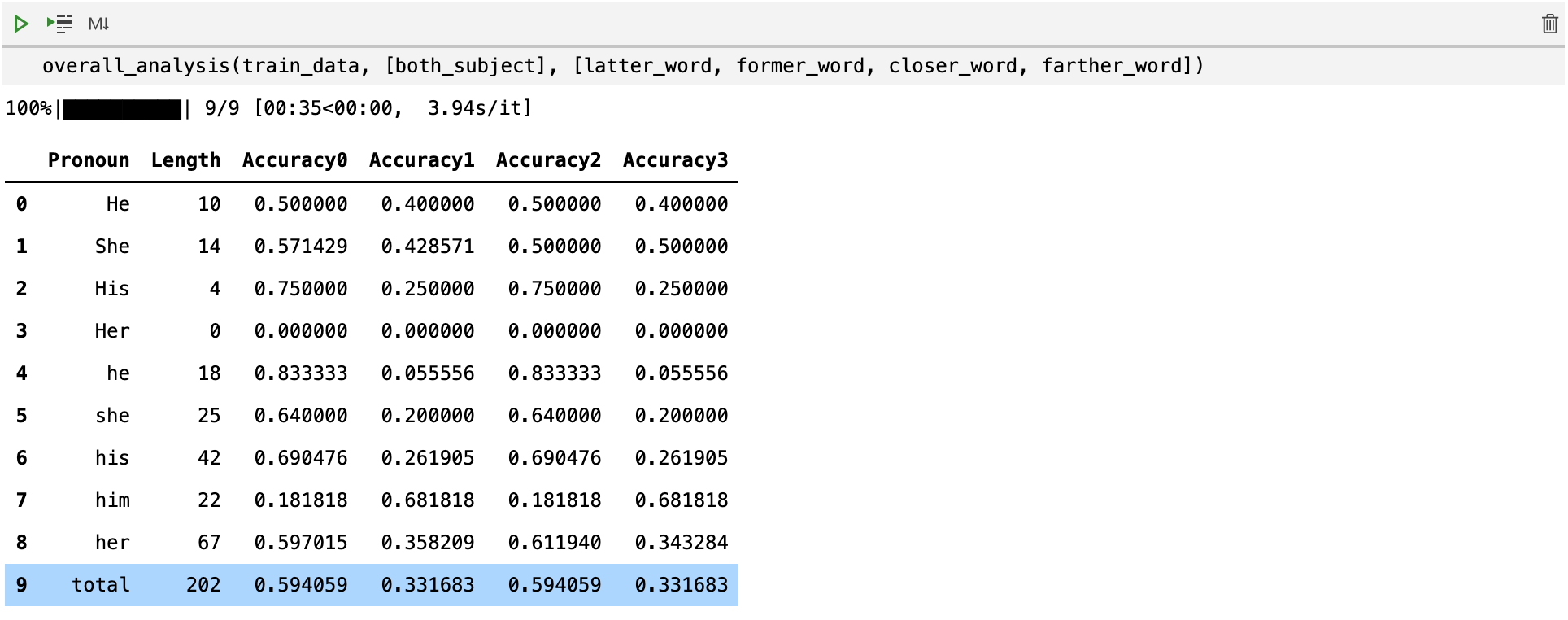
There were 1150 cases that meet this condition, so I had to make another models to deal with other 850 cases.

**Design 5-2 Both subject word**

When both words are subject word, it is difficult to choose which one will be the referenced word by pronoun.



So I first applied 4 different simple models(former word, latter word, closer word, farther word) to see which model will be the best.



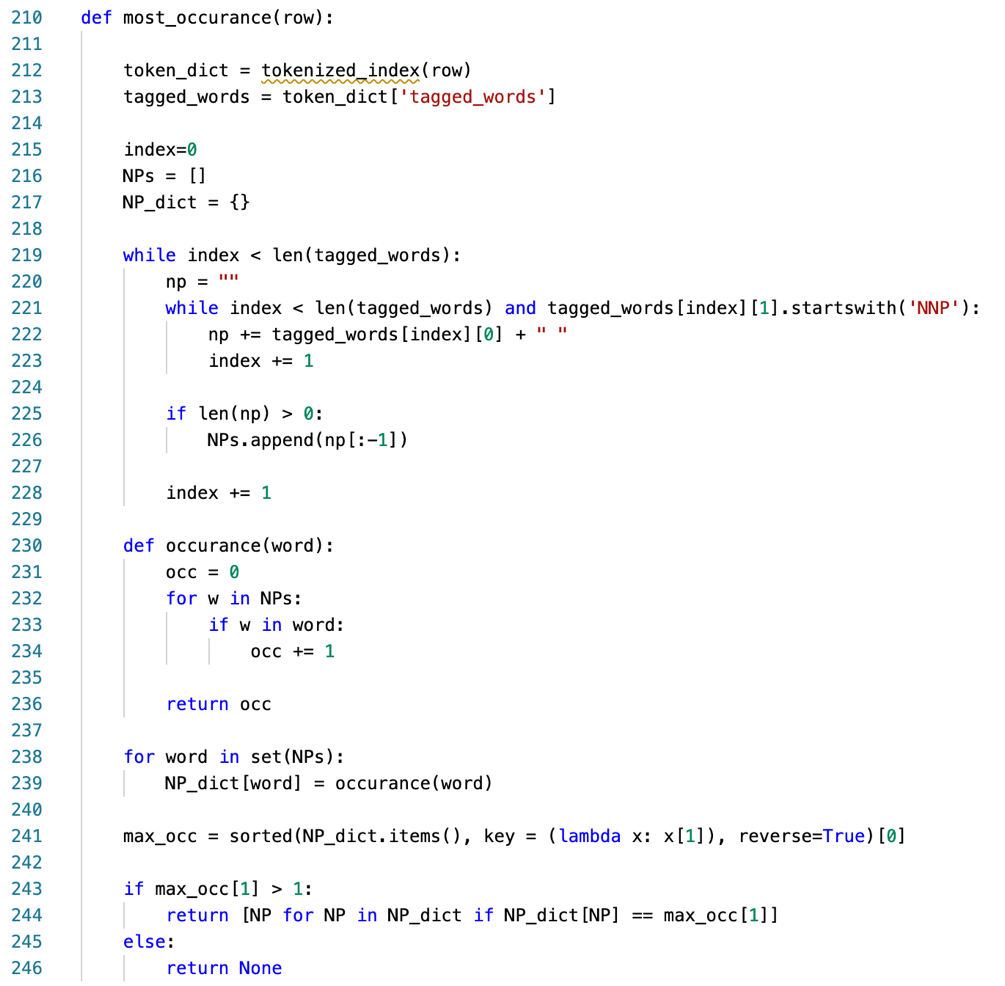
There are 202 cases that both words are subject, and except for ‘him’, predicting latter word as true shows average 60% of accuracy.

**Design 6 Word appearance**

There are about 600 cases that none of the words are subject word. And I couldn’t find any characteristics they have in common. So I need another model to deal with them.

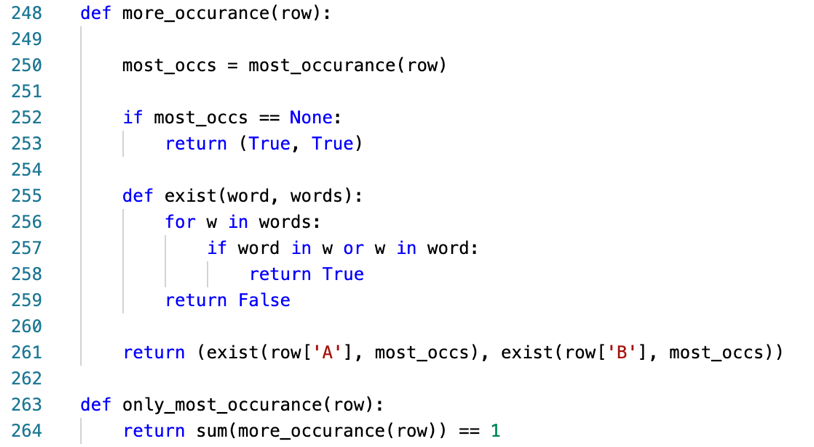
When observing the sentences, I found out that some words appear multiple times in the text. Appearing multiple times means that the word play an important role in the text, so it is more likely to be referenced by pronoun. So I counted the frequency of given words in the text.





**Design 6-1 More appearance**

If the frequency of one of the words in the text is the most, then predict the word as true, and other is false.





There are 269 cases in training set that one of the word is most frequently appearing, and total accuracy of the model is 0.58.

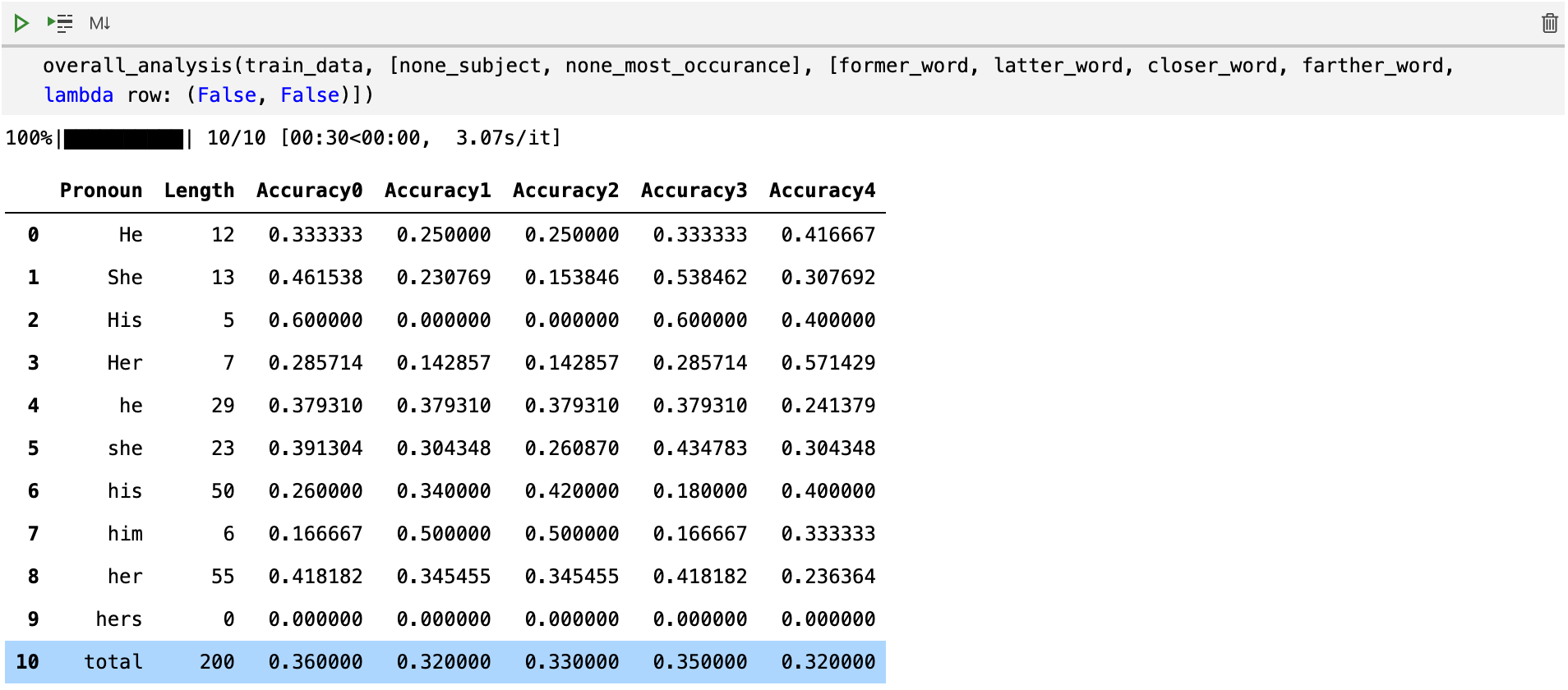
**Design 6-2 Both most appearance**

When the frequency of both words are the same (and the most), predict latter word as true when the pronoun is object/possessive pronoun, and predict former word as true when the pronoun is subject pronoun. This prediction is based on comparison between 5 simple predictions (former, latter, closer, farther, both false model)

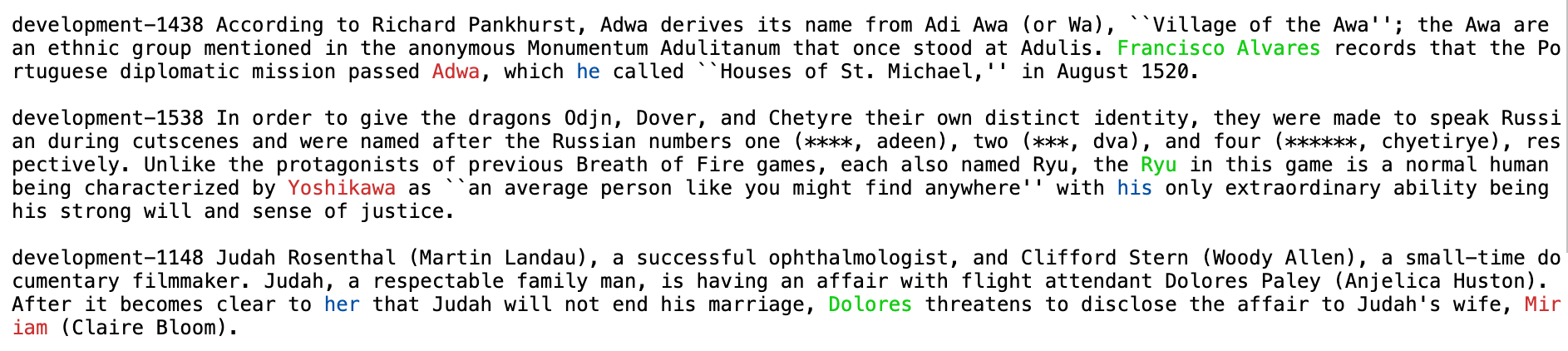


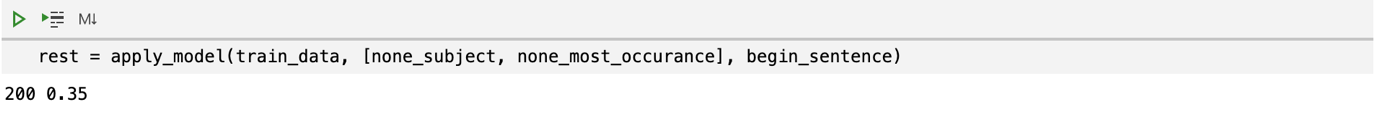
**Design 7 begin of sentence**

Now there are about 200 cases remaining that both words are not subject, and both words appear only once during the text. Even though I tried every existing simple models(former, latter, closer, farther, both false model), total accuracy was no more than 0.4.



Then I found out that referenced words are likely to appear in front of a sentence, or clause. Which means, it appears right after a dot, or comma. So I build a new model that predict word as true if the word is part of noun phrase(NP) appearing right after a dot or comma. (I will post the code in Appendix, because it is quite long.)

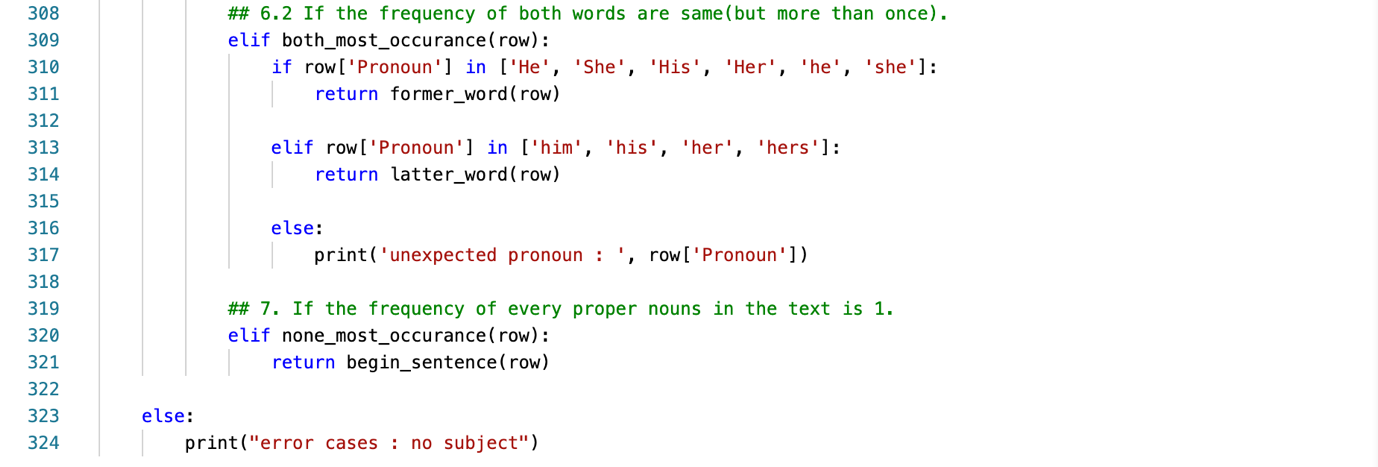
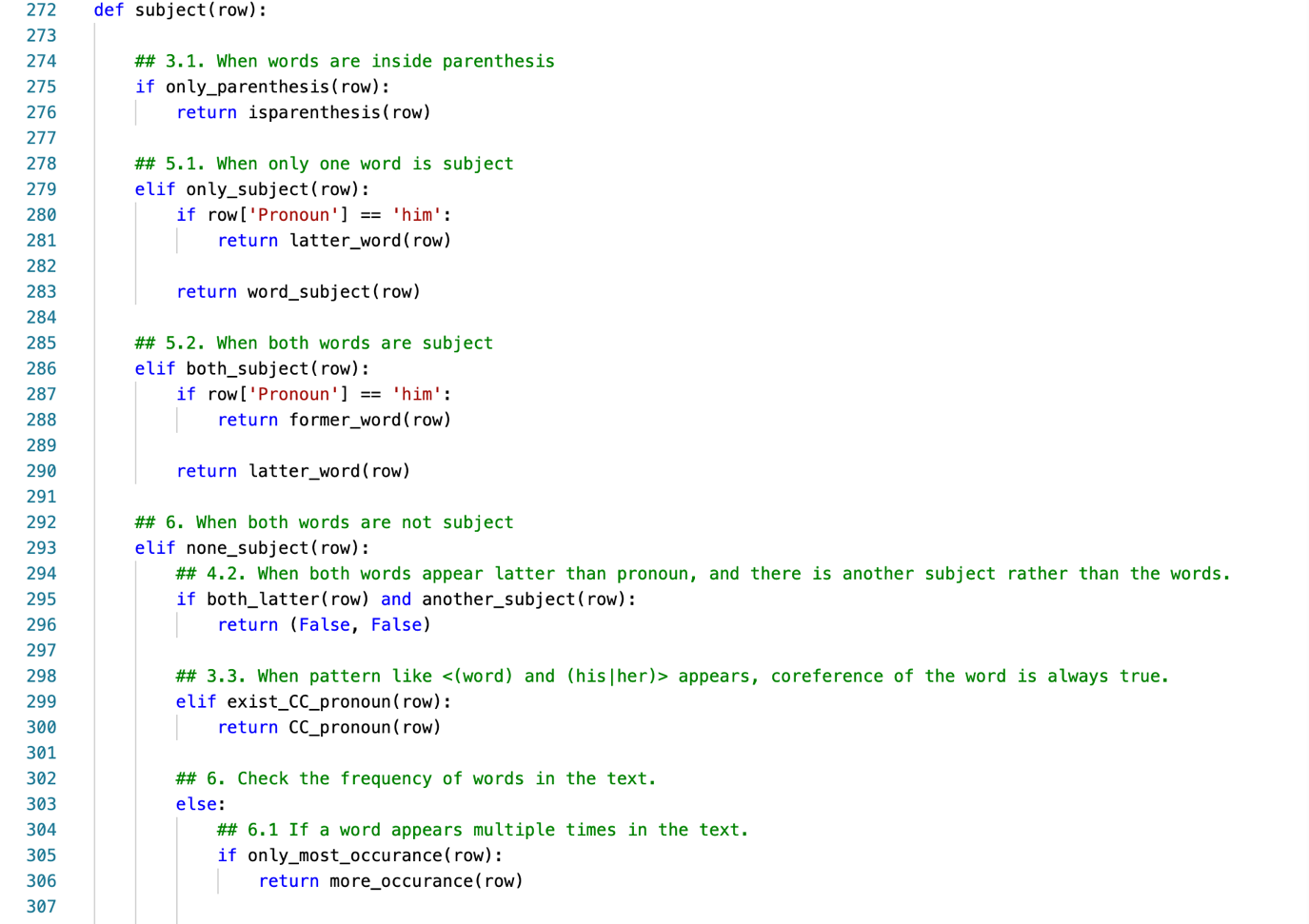




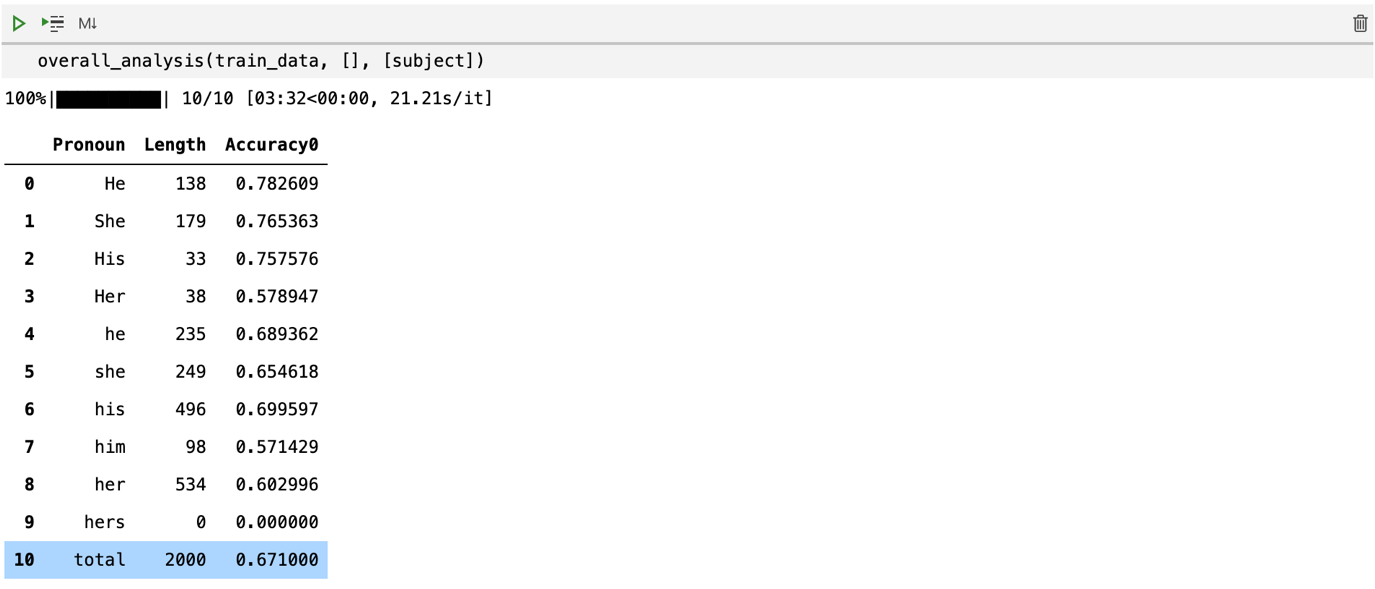
The final accuracy of this model is 0.35, which is not higher than simple prediction models at all. But still, I decided to use it as a last model.

**Design 8 Subject(Final model)**

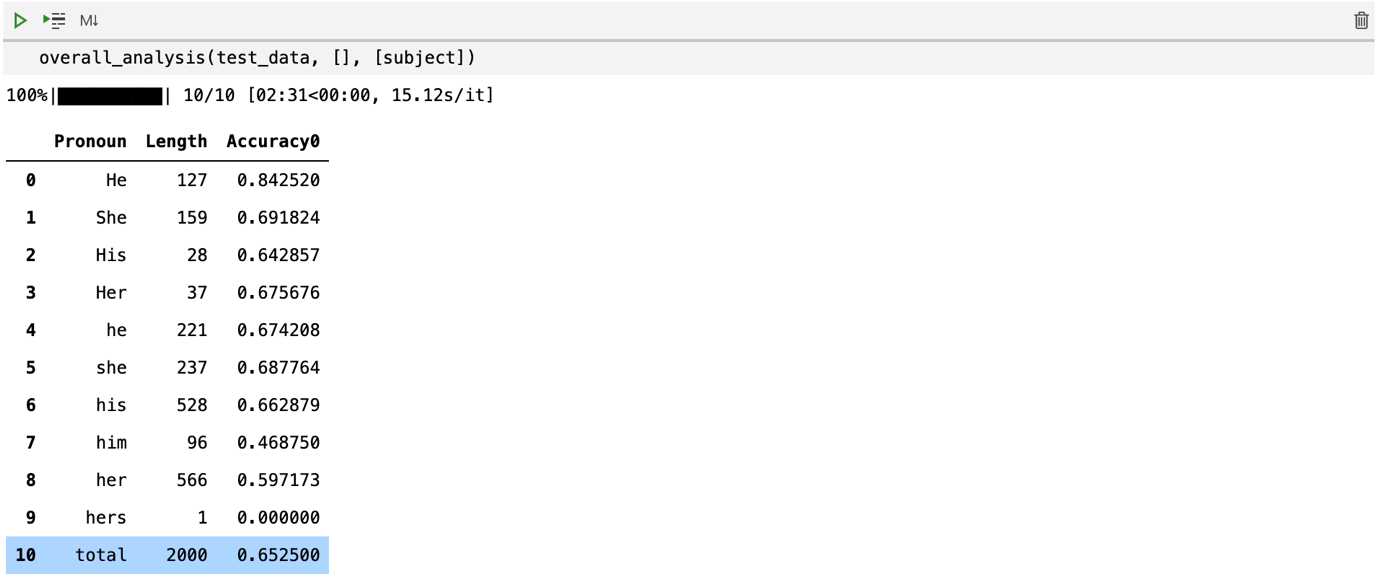
In the final step, I will gather every model I made so far, divide cases into different conditions and apply different prediction model by each case.



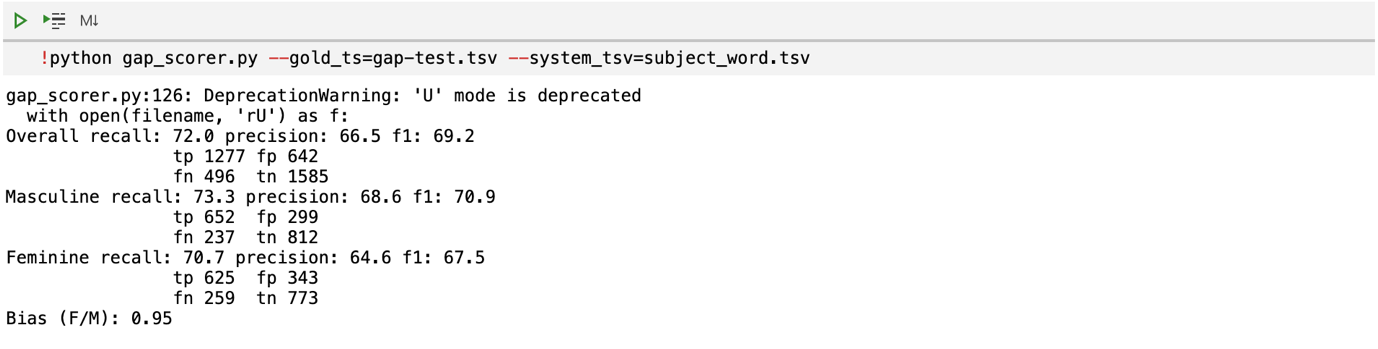
The numbers in the annotation refer to the design number of the applied model.



Total accuracy of train data(gap-development) was 0.67. I was satisfied because this was higher than I expected(0.65).



This is total accuracy of test data(gap-test). I was worried that overfitting had happened. But the difference of accuracy between training data and test data is only 0.02, so that I can say that no overfitting had happened.



f1-score of the final model is 69.2.

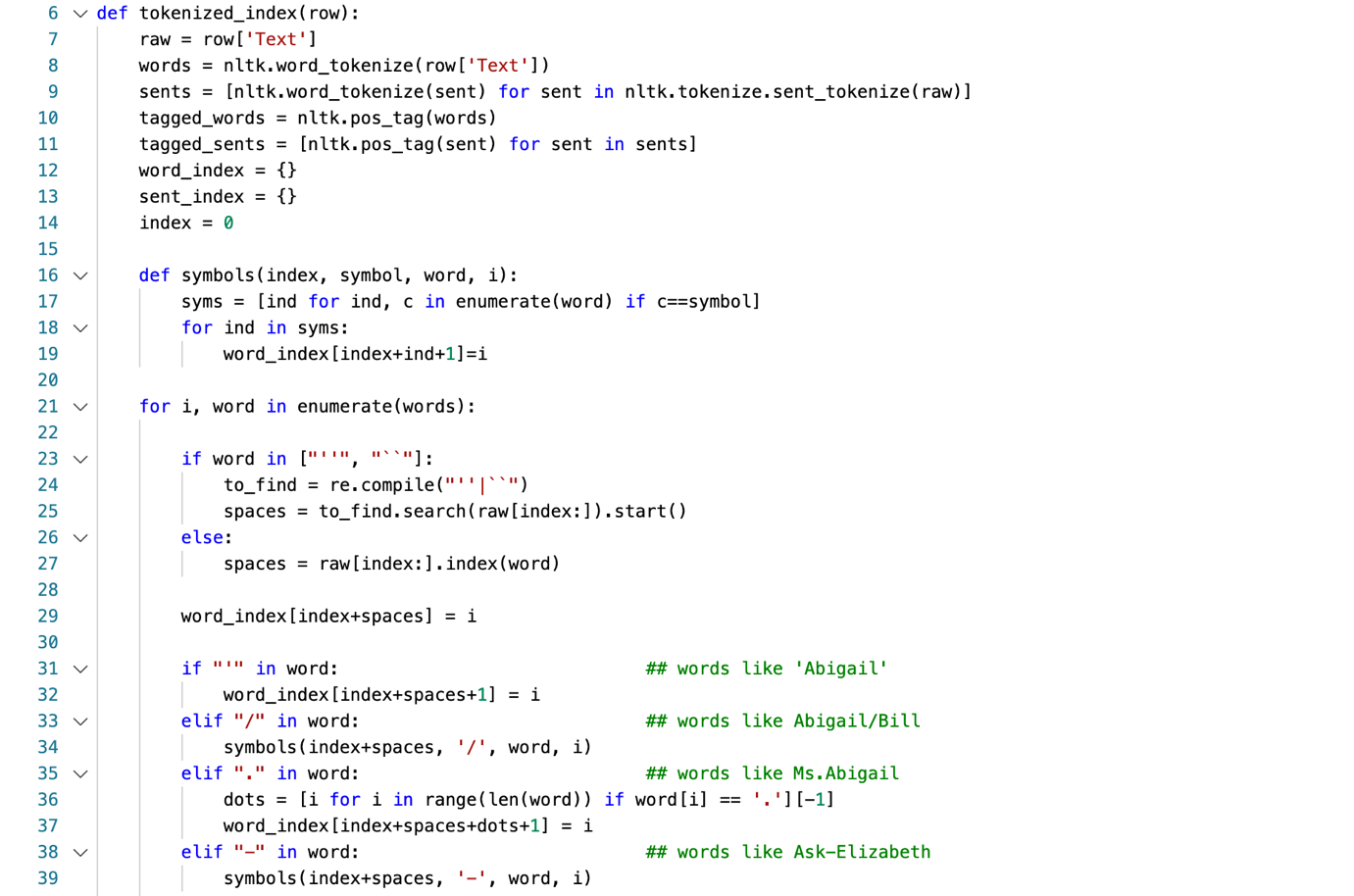
**How to improve**

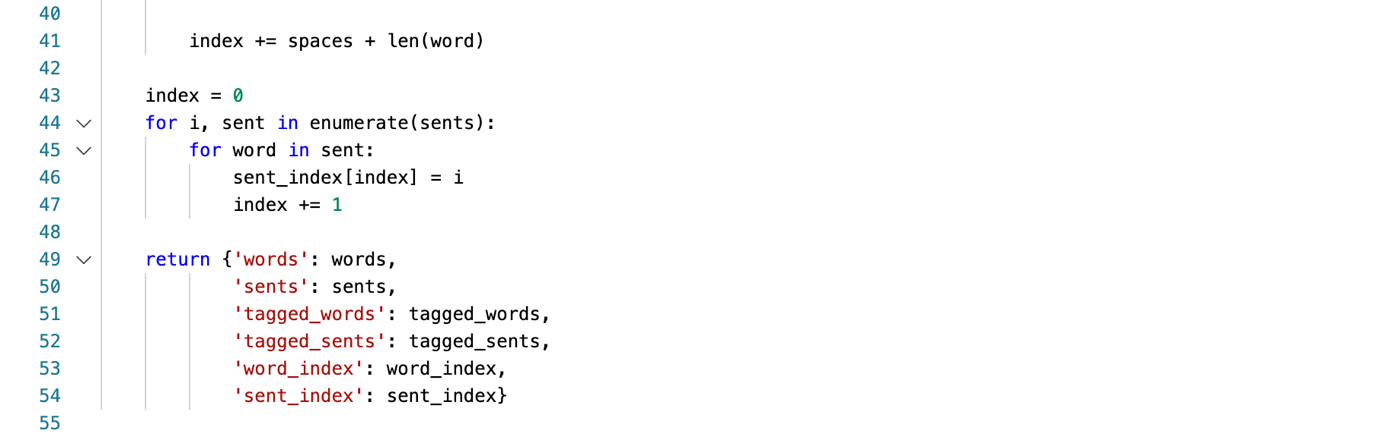
For this homework, I implemented a lot of different models : former\_word, latter\_word, closer\_word, farther\_word, both false, subject word, most appearance, always true, inside parenthesis, begin of sentence model… etc. And my final model looks a lot like a decision tree model, even though I didn’t intend it. If I could improve it later, I will think about more models, and apply them as a deeper decision tree model than now.

Especially, I will look carefully to ‘both false cases’, which played critical role in reducing accuracy. And some minor cases as well, such as objective pronoun(him, her) in Design 5-1.

**Appendix**

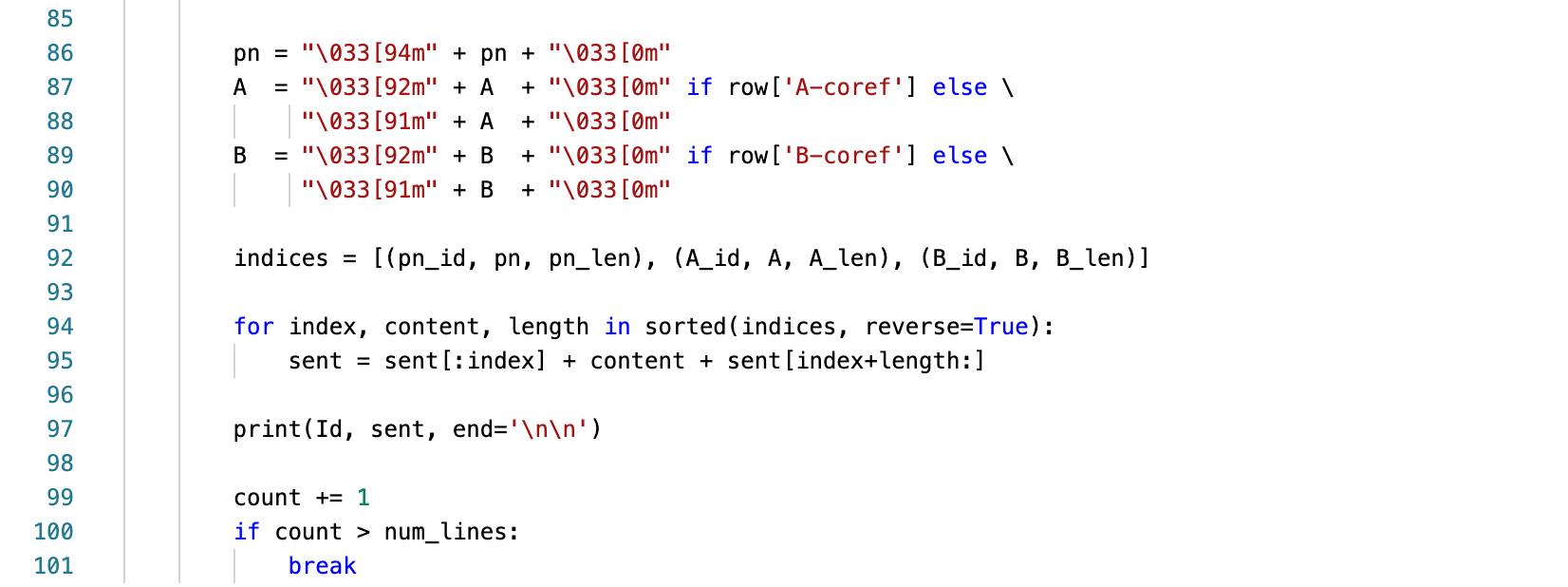
-tokenized\_index(row) : convert text into a lot of useful tags, dictionaries. The most important element is ‘word\_index’, which convert string offset into word index.

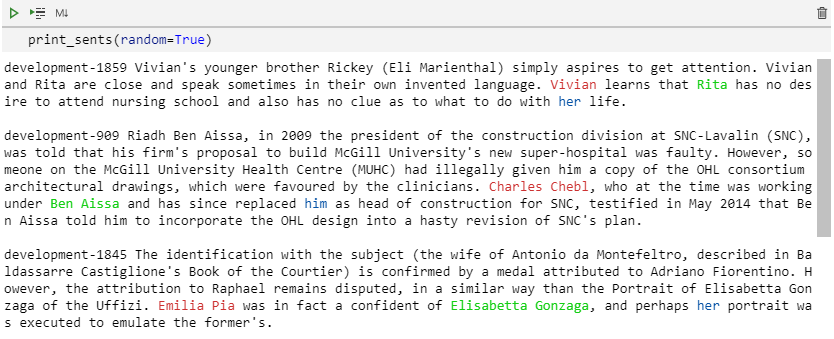




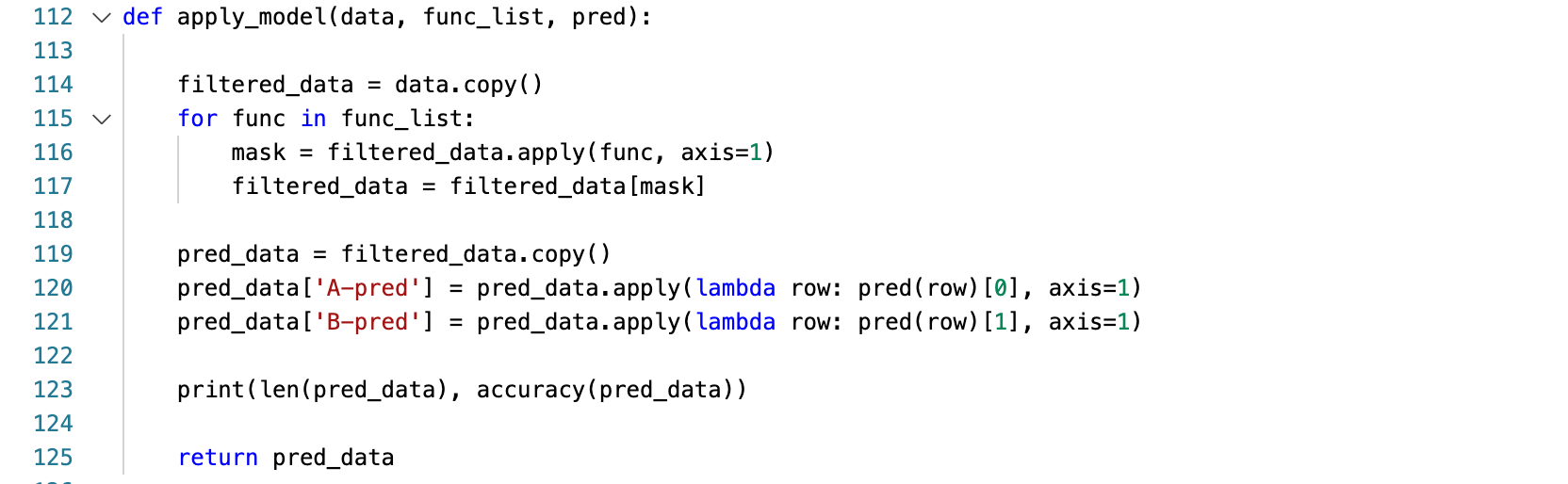
-print\_sents : print colored text with case ID. Pronoun is blue, true word is green, false word is red. We can choose which kind of data it will print.

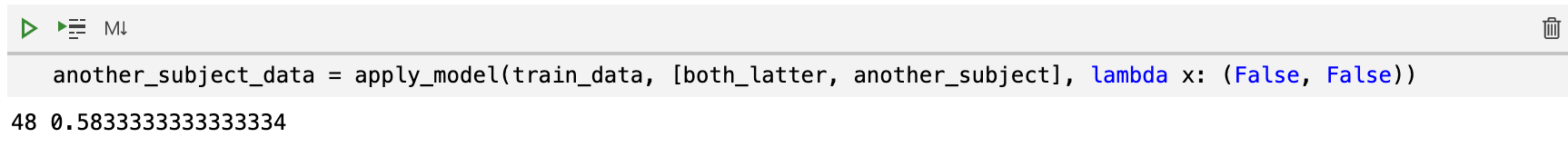




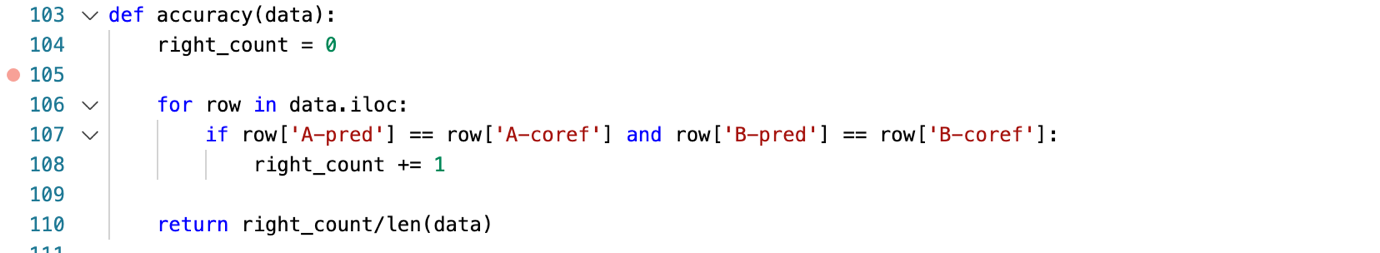


-apply\_model(data, list of condition functions, prediction function) : for given data, apply condition functions, which will filter the data with given conditions. And then apply a prediction model. Print the length of filtered data and accuracy of prediction. Return filtered data.





-accuracy(data) : input data should have following columns : ‘A-pred’, ‘B-pred’. Compare these predicted value with ‘A-coref’, ‘B-coref’, which are target columns. Return the accuracy of predicted columns.



-overall\_analysis(data, list of condition functions, list of prediction functions) : First, filter input data with condition functions. Then show the accuracy of each prediction function, by each pronoun.

