Name: Haodong Zhao. 10409845

Q1:

Following is the screenshot of part of my output

```
/usr/local/bin/python3.7 /Users/haodong/Desktop/BIA660/hw4.py
Test Q1
[('Grant Cornwell', 'College of Wooster', '2015', '911,651'), ('Marvin Krislov', 'Oberlin College', '2016', '829,913'), ('Mark Roosevelt', 'Antioch College,', '2015', '507,672'),
```

Q2:

Following is the screenshot of my output

```
Test Q2

lemmatized: No, no_stopword: No
0.6847826086956522

lemmatized: Yes, no_stopword: No
0.7445652173913043

lemmatized: No, no_stopword: Yes
0.6793478260869565

lemmatized: Yes, no_stopword: Yes
0.75
```

From the result, we can find when we lemmatize tokens and remove stop words, we get the highest percentage. Because after lemmatization, the same word with different word formation will become same word, it will enhance the accuracy. And after remove the stop words, the words left can represent the main meaning of the question either.

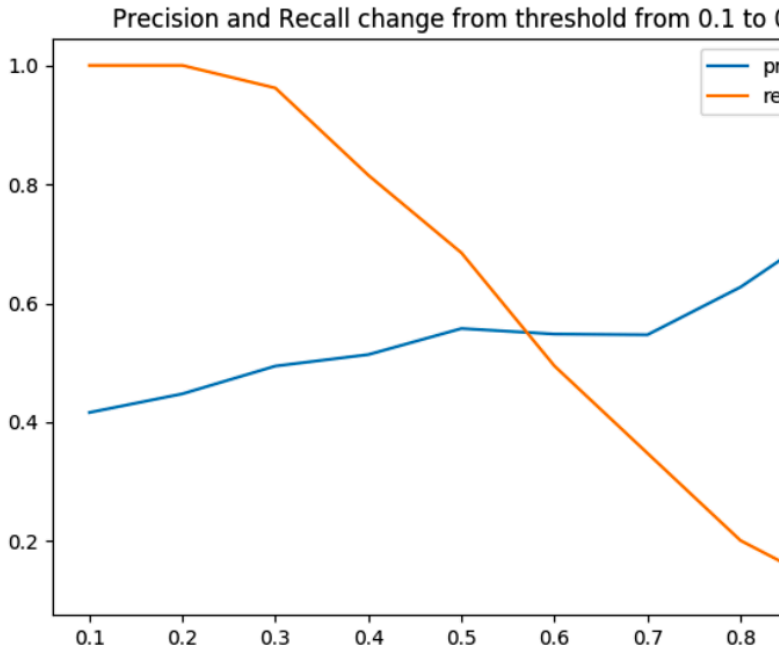
Q3.1:

Following is the screenshot of my Q3.1 output

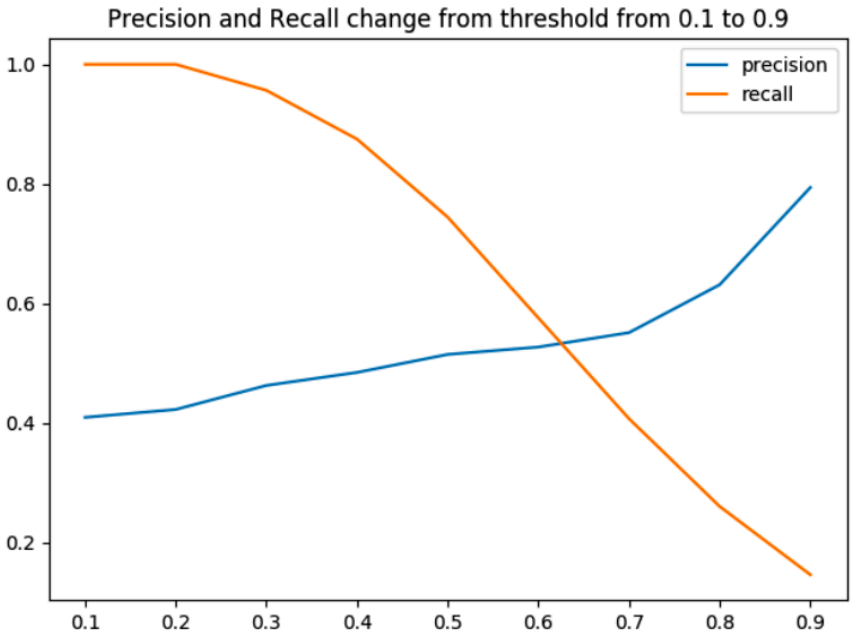
```
/usr/local/bin/python3.7 /Users/haodong/Desktop/BIA660/practice.py

Test Q3

lemmatized: No, no_stopword: No
recall is: 0.6847826086956522
precision is: 0.5575221238938053

lemmatized: Yes, no_stopword: No
recall is: 0.7445652173913043
precision is: 0.5150375939849624

lemmatized: No, no_stopword: Yes
recall is: 0.6793478260869565
precision is: 0.5364806866952789

lemmatized: Yes, no_stopword: Yes
recall is: 0.75
precision is: 0.5073529411764706

Process finished with exit code 0
```

Q3.2:

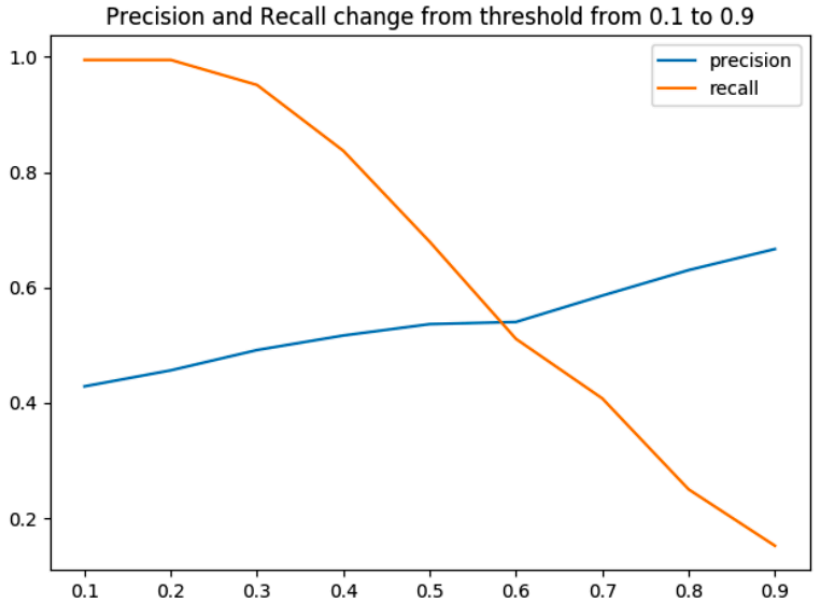
When lemmatized is False and no_stopword is False:



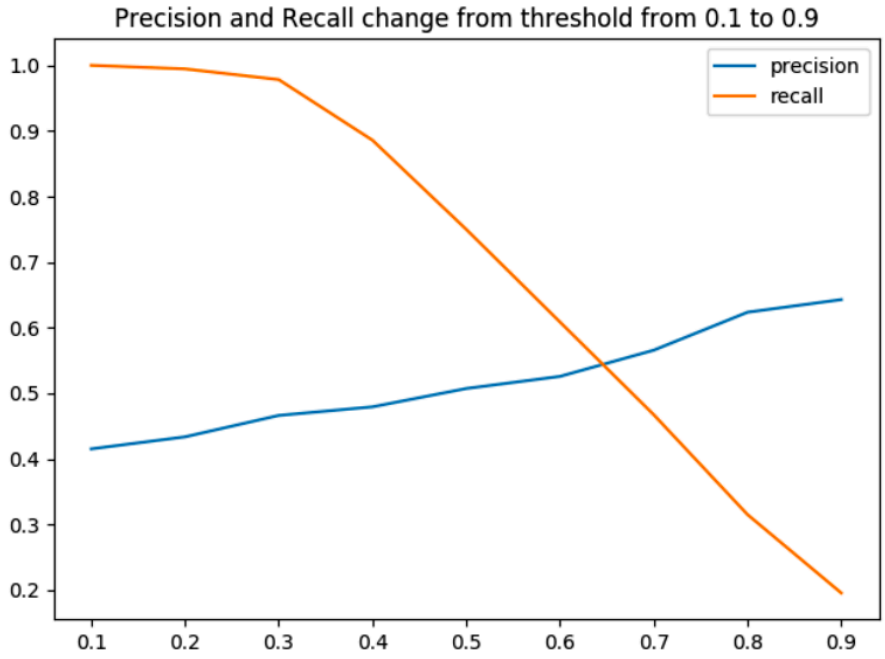
When lemmatized is True and no_stopword is False:



When lemmatized is False and no_stopword is True:



When lemmatized is True and no_stopword is True:



Answer:
When change threshold from 0.1 to 0.9 in any option situation, the recall will decrease and precision will increase.

From the plot above, I think when lemmatization and don't remove stop words will give the best performance. Because it can give the both highest value for recall and precision.

I think the easy found duplicates are: don't have many stop words and have the same words in different word formation.

I do think the TF-IDF approach is successful in finding duplicate questions, because this method ignore many meaningless words, only consider of the high-frequency word in each pair.