COMP20008 Assignment 1Task 6 (14 marks)

Hong Ye 977911

1. A description of the crawling method and a brief summary the output for Task 1. (2 marks)

'requests' library and BeautifulSoup from 'bs4' library are used to complete Task 1, which is to visit all pages accessible from the site and record their headlines and URLs.

To start the crawling process, the seed URL is visited first via requests.get(seed_url). Then, BeautifulSoup(page.text, 'html.parser') is used to extract the html script of the page. Then I use soup.findAll('a') to find outbound links on the current page and store them to a **to_visit** list. A dictionary **visited** is also used to specify that the seed URL is visited.

After visiting the seed URL, I visit all URLs in the **to_visit** list one at a time.
When scraping text from a particular URL, the URL from **to_visit** list is removed and appended to a **urls** list to keep track of URLs visited except for the seed URL. To retrieve html script on each page, the same process is applied when visiting the seed URL. soup.findAll(attrs = {"class": "headline"}) is used to extract headline stored to a list **headlines**.

I follow the same procedure to find all outbound links on the current page and store them to **to_visit** list to continue to find all pages accessible from the site.

To avoid appending repetitive URLs to **to_visit** list, I use **visited** dictionary and **to_visit** list to check. I will only append a new URL to **to_visit** if URLs are not in the **visited** dictionary and not in the **to_visit** list.

I also impose a page limit of 999 pages to avoid breaking the site. The final output is 'task1.csv' which contains two columns named **urls** and **headlines.** It contains 148 rows of results excluding the row for column names.

Limitation of this method is that two articles may have the same headline and content but have different URLs. Both of them are appended to the list despite their repetitive nature.

2. A description of how you scraped data from each page, including any regular expressions used for Task 2 and a brief summary of the output. (3 marks)

The first half of task 2 is to extract the first team mentioned in the article. rugby.json is read to create a dictionary of team names {'Zealand': 'New Zealand', 'Wales': 'Wales', 'England': 'England'…}. A list **body_text** is created to store text from the current page except the title.

```
body = soup.findAll('div', id = 'main_article')
body_text = [re.sub(r'<.+?>',r'',str(a)) for a in body]
```

That is to convert all text from html script into a list of strings while removing all html tags such as <a> <p>. This creates a list of sentences from the article.

While looping through each row in **body_text**, I tokenize each row and loop through words in each row until one team name is found. I then append the name to list **team**. If it is not found, I append None to **team**.

Limitation of this method is that I search for matching team name based on their last names. For instance, to find team 'New Zealand' I search word 'Zealand' in tokenized rows. This may create conflicts such as a matched 'Korea' may be either 'North Korea' or 'South Korea'.

The second half of the task is to find the largest match score in the article. I do this by looping through

body_text and find the pattern using re.findall to append to list **scores**.

My pattern is r'[\s\.\,\?\;\:\'\"\(\[\{*\!]\d{1,3}\-\d{1,3}[\s\.\,\?\;\:\'\"\)\]\}*\!]'

which allows a maximum of 999-999 score in a Rugby game as there is a maximum of 3 digit score in a Ruby game. The [ ] before and after the score pattern indicate that one of the characters in the brackets will be leading and trailing the score pattern. The white space character '\s' will prevent extracting invalid scores '1993 – 93 as '993-93', while other characters allow valid scores such as "50-50", (50-50), 59-10. to be identified.

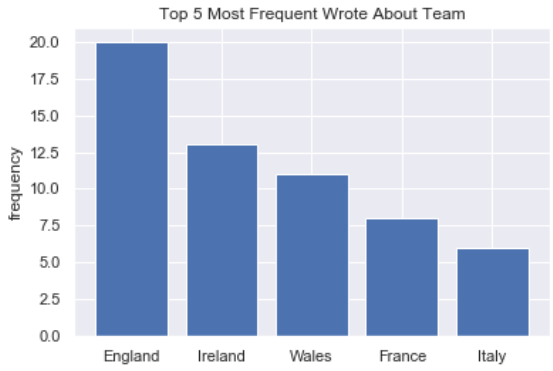
After I extract scores from the article along with the punctuations/whitespace character, I loop through **scores** to determine the largest score. I first separate the score for each team by applying .split('-'). I then remove punctuations/whitespace character to covert the score to integer. I compare each match score pair to determine the highest score of a team and append it to **team_score** list.

I create a panda dataframe which contains four lists **urls**, **headlines**, **team**, **team_score** and export it as 'task2.csv'.

'task2.csv' consists of four columns mentioned above with 147 rows of data excluding the row for column names.

Limitation for this method is that it is unable to identify some valid scores if the format is invalid. '50-50scores' is a valid score with formatting errors which this method will not identify as a valid score.

3. An analysis of the information shown in the two plots produced for Tasks 4 & 5, including a brief summary of the data used. The plots are to be shown (included) along with your analysis. (4 marks)



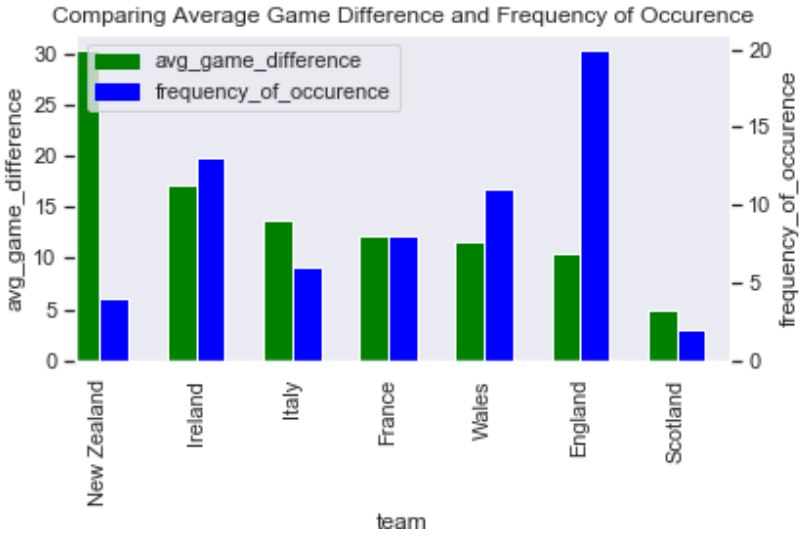
Task 4:

The top 5 teams that are most frequently mentioned in the articles are England, Ireland, Wales, France, and Italy in descending order shown in the bar plot above. The data used to produce the bar plot is the pandas dataframe created for task 2 which contains a list of articles with their teams and scores. I sort team names using value_counts and order them by frequency and then by alphabetical order.

The top 5 most frequently mentioned teams have a frequency range from 6 to 20 times. Team England is mentioned most frequently with 20 times, while Italy is mentioned least frequently with 6 times. The median frequency is 11 times which comes from Team Wales.

Limitation of the plot is that it if two teams ranked the same in terms of frequency, only the team name that has higher alphabetical order will be shown first.

Task 5:



For comparing each team's average game difference and its frequency of occurrence, a bar plot with two y axis is created with average game difference and frequency of occurrence. The x axis is the name of the team ordered by average game difference and then by alphabetical order.

The data used to create this plot is task 4 dataframe and task 3 dataframe. Task 4 dataframe contains team names and their frequency of occurrence, while task 3 dataframe contains team names and their average game difference. Inner joining the two dataframes creates a new dataframe that contains all information for each team used to create this plot.

From the bar chart it is observed that Team New Zealand has the highest average game difference of 30.5 points followed by Team Ireland with an average game difference of 17.23 points. It is not reasonable to conclude that New Zealand average game difference is the highest since the average came from a small sample size of 4 articles. However, it is relatively reasonable to conclude that Ireland average game difference is high compared to other teams as it comes from a relatively large sample size of 19 articles.

The team with second lowest average game difference is England with an average game difference of 10.5 followed by Scotland with an average game difference of 5. It is reasonable to conclude England's average game difference is low among other teams, while it is not reasonable to conclude that of Scotland due to the small sample size.

The median average game difference is 12.125 which comes from the Team France.

It may be concluded that team performance in terms of average game difference has a ranking of: Ireland, Italy, France, Wales, England in descending order. (excluding New Zealand and Scotland as sample size is below 5). However, it may not be reasonable to conclude that Ireland outperforms all other teams. Limitation of this assumption is that we are assuming teams always win which is highly unlikely. Another reason that some teams rank higher or lower in the table may be due to the fact that there are repetitive articles about the same match where the score is very high or low.

4. A discussion of the appropriateness of associating the first named team in the article with the first match score. (2 marks)

The appropriateness of associating the first named team in the article with the first match score is a discussion of the following subtopics:

Appropriateness of associating the article with regards to the first named team

Based on the assumption that one article is only about one team, associating the content with the first named team is a good choice since the subject of article is likely to be mentioned first. An alternative way to decide the subject of the article is to find the team name that is most frequently mentioned. However, the latter option is less practical since searching the entire article for each team name of interest is costly compared to searching the first matched team name. Therefore, the first option is

selected based on a tradeoff between search costs and accuracy.

However, the assumption that an article is only about one team is not practical since at least two teams are mentioned in each article of rugby games. Assuming so may lead to recorded frequency less than actual frequency.

Appropriateness of associating the first match score with the first named team

Associating the first match score with the team is not a very good choice as this may not be the final score. An alternative method is to search the article for the highest score. This method is more appropriate as it ensures the score is final. The search cost is not much as looping through the entire article once is not very costly.

In conclusion, associating the article with the first named team is feasible, while associating the team with the first matched score is not reasonable. A better way is to associate the first named team with highest match score which have relative low search costs and high accuracy. However, all methods mentioned above are based on the assumption that each article is only about one team, which may lead to inaccurate results if multiple teams are mentioned in the article or the score extracted is not relevant to the team.

5. At least two suggested methods for how you could figure out from the contents of the article whether the first named team won or lost the match being reported on and a comment on the advantages and disadvantages of each approach. (2 marks)

One method to determine if the team of interest won, lost or tied is to find the first matched word from the list that implies match status such as ‘win’, ‘won’, ‘defeat’, ‘draw’, ‘even’, ‘tie’ in the article.

Advantage: These words will be common and can be applied to those articles without numerical match scores.

Disadvantage: If a sentence structure is ‘other team defeat team’, it will be interpreted as the team of interest winning instead of losing.
It can be hard to create an exhaustive list that implies all match status without the use of nltk library functions to categories these words.

Another approach is to calculate the score difference in the highest match score to see whether it is positive, negative or zero. Positive scores indicate wining, while negative scores indicate losing based on the assumption that the score will be in the format ‘team - other team’. For instance, a match score of ‘18-25’ will have a difference of ‘-7’, which indicates that the team loses the game.

Advantage: Fewer ambiguous situations compared to the first method.
Disadvantage: If the score is written in the format ‘other team score – team score’, using score difference to determine if the team of interest is wining or losing will be misleading.
If no score is available, match result cannot be concluded.

6. A discussion of what other information could be extracted from the articles to better understand team performance and a brief suggestion for how this could be done. (1 mark)

We can extract all score patterns using re.findAll(pattern, row) from each article to understand how match scores change throughout the match to get an idea of team performance. For instance, if the score difference change drastically throughout the match, it may imply that one team is performing significantly better than the other.

Words that have positive or negative connotations can be used to determine team performance. For instance, ‘good’, ‘well’, ‘fantastic’, ‘stuttering’ indicate that the team has a better team performance, compared to those words such as ‘tough’, ‘hard’. These words can be identified via the SentimentIntensityAnalyzer from nltk.sentiment.vader to analyse sentiment intensity of words to determine if they are neutral, negative or positive.

Matching key words ‘penalty’, ‘replacement’ to understand if any special situation occur on each match that may affect team performance.