PROJECT ASSIGNMENT PRESENTATION

2024/02/21

HOW AND WHAT WE USE

- 使用 Python 撰寫 parser 來統計 title 和 keyword 數量
- 運用 set 和 dictionary(map)

ASSIGNMENT 2.1

- Code

```
# iterate through each file
for file_path in file_paths:
    with open(file_path, 'r', encoding="utf-8") as file:
        # initialize title and insideTI
        title = ""
        insideTI = False
        # iterate through each line
        for line in file:
            # starts of a title
            if line.startswith("TI "):
                insideTI = True
                title += line[3:].strip()
            # contents inside the TI tag
            elif line.startswith("   ") and insideTI:
                title += line[3:].strip()
            else:
                # print the current (title, file) and correspond (title, file) if the title has been seen before
                if title in titles:
                    print("---")
                    print(f'Title "{title}" appears in "{file_path}".')
                    print(f'Title "{title}" appears in "{title_file[title]}".')
                    print("---")
                # add title count and add it into the set and dictionary
                if title != "":
                    count += 1
                    titles.add(title)
                    title_file[title] = file_path
                # re-initialize title and insideTI
                title = ""
                insideTI = False

print(f"Number of titles: {count}")
print(f"Number of unique titles: {len(titles)}")
```

- Output

```
Number of titles: 26287
Number of unique titles: 26269
```

ASSIGNMENT 2.2

- Code

```
# iterate through each file
for file_path in file_paths:
    with open(file_path, 'r', encoding="utf-8") as file:
        # initialize keyword and insideDE
        keyword = ""
        insideDE = False
        # iterate through each line
        for line in file:
            # starts of keywords
            if line.startswith("DE "):
                insideDE = True
                keyword += line[3:].strip()
            # contents inside the DE tag
            elif line.startswith("   ") and insideDE:
                keyword += line[3:].strip()
            else:
                # add each keyword into the set and dictionary
                if keyword != "":
                    keyword = keyword.split(';')
                    for word in keyword:
                        word = word.strip() # .lower() to avoid same keywords but with different upper/lower characters
                        keywords.add(word)
                        # if word in keyword_count, increase keyword_count[word] by one
                        # else, set keyword_count[word] to one
                        if keyword_count.get(word, False):
                            keyword_count[word] += 1
                        else:
                            keyword_count[word] = 1
                # re-initialize keyword and insideDE
                keyword = ""
                insideDE = False

# sort the dictionary with it's value in descending order
sorted_keywords = sorted(keyword_count.items(), key=lambda x: x[1], reverse=True)

with open('output.txt', 'w') as file:
    for key, value in sorted_keywords:
        print(f"{key}({value})", file=file)
```

- Output

```
autonomous vehicles(2885)
autonomous vehicle(1102)
autonomous driving(982)
autonomous aerial vehicles(911)
path planning(737)
deep learning(723)
vehicle dynamics(721)
trajectory(715)
optimization(620)
task analysis(591)
safety(585)
sensors(571)
autonomous underwater vehicle(566)
navigation(529)
collision avoidance(520)
roads(498)
reinforcement learning(451)
uav(447)
object detection(435)
machine learning(415)
auv(414)
autonomous underwater vehicles(398)
feature extraction(359)
planning(356)
artificial intelligence(328)
unmanned aerial vehicles(327)
localization(322)
autonomous underwater vehicle (auv)(320)
computer vision(315)
cameras(305)
```