دانشکده مهندسی کامپیوتر

گزارش فاز اول

باوان دیوانی آذر ۹۸۵۲۲۲۱۱

نیم‌سال دوم

سال تحصیلی ۰۲-۱۴۰۱

اطلاعات جمع آوری شده از سایت زیر است:

https://store.steampowered.com/

با استفاده از کتابخانه BeautifulSoup اطلاعات مورد نیاز از سایت crawl شده است. به دلیل تعداد بالای رکوئست‌ها و پرشدن حافظه‌ی RAM هر ۵ هزار داده در یک فایل ذخیره شده‌اند. اما در نهایت، اطلاعات مربوط به بازی‌های یک گروه، در یک فایل قرار داده شده‌اند.

روش کار به این صورت است که در ابتدا، لیست app_id مربوط به بازی های یک گروه را استخراج کرده، به کمک تابع add_information به تک تک app_idها، رکوئست می‌فرستیم تا اطلاعات (نام و توضیحات) مربوط به بازی، به دست آید.

کد:

```
categories = ['action','adventure','rpg','strategy','simulation','sports_and_racing']
data = []
for category in categories:
    print(category)
    start = 5000
    while True:
        response = requests.get(f"https://store.steampowered.com/saleaction/ajaxgetsaledyna
        app_ids = response.json()["appids"]
        response.close()
        start += 100
        print(start)
        if len(app_ids) == 0:
            break
        add_information(app_ids, category, data)
        if start % 5000 == 0:
            df = pd.DataFrame(data, columns=['id', 'name', 'category', 'about'])
            filename = f'gdrive/MyDrive/NLP/Dataset/{category}{start//5000}.csv'
            df.to_csv(filename, index=False, quoting=csv.QUOTE_ALL)
            data = []
    if start % 5000 != 0:
        df = pd.DataFrame(data, columns=['id', 'name', 'category', 'about'])
        filename = f'gdrive/MyDrive/NLP/Dataset/{category}{(start//5000)+1}.csv'
        df.to_csv(filename, index=False, quoting=csv.QUOTE_ALL)
        data = []
```

```
def add_information(app_ids, category, data):
    for i, app_id in enumerate(app_ids):
        response = requests.get(f"https://store.steampowered.com/app/{app_id}")
        soup = BeautifulSoup(response.content, 'lxml')
        response.close()
        try:
            name = soup.title.string
            info = soup.find("div", {"id": "aboutThisGame"}).get_text("\n")
            info = re.sub('(\r\n|\r|\n)+', '\n', info).replace("\t","")
            data.append([app_id, name, category, info])
        except:
            print(f"An exception occurred app_id = {app_id}")
            print(f"https://store.steampowered.com/app/{app_id}")
```

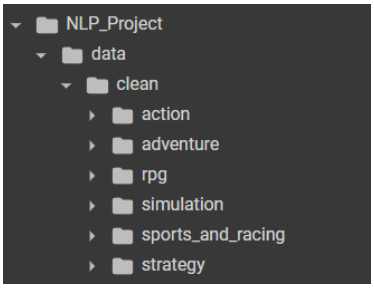
```
categories = ['action','adventure','rpg','strategy','simulation','sports_and_racing']
for category in categories:
  PATH = 'gdrive/MyDrive/NLP/Dataset/'
  df1 = pd.read_csv(f'{PATH}{category}1.csv')
  df2 = pd.read_csv(f'{PATH}{category}2.csv')
  df3 = pd.read_csv(f'{PATH}{category}3.csv')

  dict1 = df1.to_dict('dict')
  dict2 = df2.to_dict('dict')
  dict3 = df3.to_dict('dict')

  combined_dict = []
  for i in range(len(dict1['id'])):
      combined_dict.append([dict1['id'][i],dict1['name'][i],dict1['category'][i],dict1['about'][i]])
  for i in range(len(dict2['id'])):
      combined_dict.append([dict2['id'][i],dict2['name'][i],dict2['category'][i],dict2['about'][i]])
  for i in range(len(dict3['id'])):
      combined_dict.append([dict3['id'][i],dict3['name'][i],dict3['category'][i],dict3['about'][i]])

  directory = f'gdrive/MyDrive/NLP/github/data/raw/{category}/'
  df = pd.DataFrame(combined_dict, columns=['id', 'name', 'category', 'about'])
  if not os.path.exists(directory):
    os.makedirs(directory)
  df.to_csv(f'{directory}{category}.csv', index=False, quoting=csv.QUOTE_ALL)
```

لیست اطلاعات بازی های مربوط به ۶ گروه دراورده شده و اطلاعات هر گروه در فولدر مربوطه قرار گرفته شده است. فرمت داده ها به شکل فایل csv. هست.



برای تفکیک جملات و کلمات، از متدهای موجود در کتابخانه nltk استفاده کردم. این متدها شامل توابعی مانند sent_tokenize() برای تفکیک جملات و word_tokenize() برای تفکیک کلمات هستند.

```
for i in range(len(data['clean_description'])):
    id = data['id'][i]
    description = data['clean_description'][i]
    sentences = nltk.sent_tokenize(description)
    words = word_tokenize(description)
    sentences_data.append([id, sentences])
    words_data.append([id, words])
```

برای بخش clean کردن داده، ابتدا عبارت "About the game" را از ابتدای متن حذف کرده و سپس بررسی کرده‌ایم که آیا توضیحات بازی کمتر از ده حرف هستند؟ در صورتی که چنین باشد، آن‌ها را نیز حذف می‌کنیم.

```
for i in range(len(data['about'])):
    id = data['id'][i]
    description = data['about'][i]
    clean_description = description.replace('\nAbout This Game\n', '')
    data_count+=1
    if(len(clean_description)>10):
      clean_data_count +=1
      clean_data.append([id, clean_description])
```

تعداد داده قبل و بعد از clean کردن :

```
before cleaning data : 78719
after cleaning data: 78665
```

واحد برچسب گذاری برای هر متن بازی ، ژانر خود بازی هست. برای مثال اگر ژانر بازی اکشن باشند برچسب توضیحات آن ، اکشن می‌شود.

آمار داده به تفکیک برچسب در قالب جدول «و» نمودار

أ. تعداد «واحد» داده

$_{a}nd_{r}acing$"sports	"simulation"	"strategy"	"rpg"	"adventure"	"action"
"7206"	"14994"	"11513"	"14987"	"14982"	"14983"

ب. تعداد جملات

$_{a}nd_{r}acing$"sports	"simulation"	"strategy"	"rpg"	"adventure"	"action"
"76045"	"172196"	"155396"	"168316"	"170770"	"167376"

ج. تعداد کلمات

$_{a}nd_{r}acing$"sports	"simulation"	"strategy"	"rpg"	"adventure"	"action"
"1497667"	"3563133"	"3236588"	"3458951"	"3306827"	"3287666"

د. تعداد کلمات منحصر به فرد

"uncommon"	words" "common	words" "unique	"category"
"17348"	"85947"	"103295"	"action"
"16644"	"90025"	"106669"	"adventure"
"16927"	"95099"	"112026"	"rpg"
"20025"	"77818"	"97843"	"strategy"
"28647"	"86454"	"115101"	"simulation"
"9456"	"52599"	"62055"	$_{a}nd_{r}acing$"sports

ه. تعداد کلمات منحصر به فرد مشترک و غیر مشترک بین برچسب ها

"uncommon"	words" "common	words" "unique	"category"
"17348"	"85947"	"103295"	"action"
"16644"	"90025"	"106669"	"adventure"
"16927"	"95099"	"112026"	"rpg"
"20025"	"77818"	"97843"	"strategy"
"28647"	"86454"	"115101"	"simulation"
"9456"	"52599"	"62055"	$_{a}nd_{r}acing$"sports

و. ۱۰کلمه پرتکرار غیر مشترک هر برچسب

"word10"	"word9"	"word8"	"word7"	"word6"	"word5"	"word4"	"word3"
"GameGuru"	"Paintjob"	"Jutsu"	"honmaru"	"Warzone "	"Honmaru"	"Ranbu"	"CA2S"
"Pinecreek"	"Putt-Putt®"	"ScummVM"	"Mimpi"	"Log1"	"•Never"	"Deponia"	"HOPs"
" Special"	" Music"	"MML2"	"Sevin"	"Zelam"	"S.F.A"	"x2000"	"Lumena"
"Senet"	"9.9.2"	"ChessBase"	"Freecell"	"non-mythic"	"eXtra"	"ENnie"	"Catan"
"Bahn"	"Railfan"	"couplers"	"doubledecker"	"QJ"	"Quinnimont"	"TRS19"	"Subdivis
" "	"Gloryhammer"	"Ahlman"	"TrackMania"	"Betelgeuze"	"Ragnarock"	"Riskers"	"•Maximum

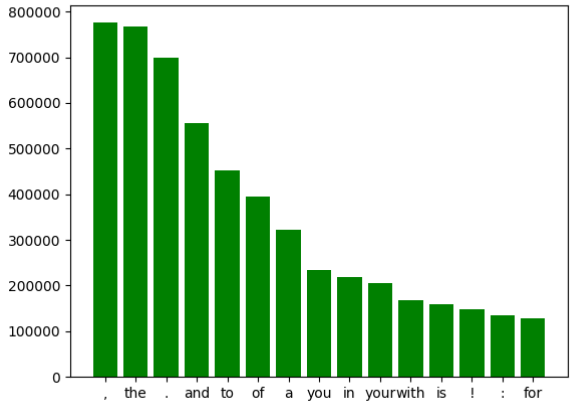
ز. ۱۰کلمه مشترک برتر هر برچسب نسبت به برچسبهای دیگر بر اساس معیار زیر

"word10"	"word9"	"word8"	"word7"	"word6"	"word5"	"word4"	"word3"
"Railroad"	"Adapted"	"Converted"	"Pathfinder"	"freight"	"Grounds"	"pre-placed"	"parcels"
"Arma"	"DB"	"BR"	"mm"	"parcels"	"km/h"	"handouts"	"pre-placed"
"BR"	"cab"	"Livery"	"GT"	"brake"	"RC"	"Arma"	"Airport"
"off-road"	"Duty®"	"Tyre"	"XR"	"PACK"	"Stella"	"brake"	"RC"
"Terms"	"Conversion"	"resized"	"Adapted"	"JRPG"	"ruleset"	"handouts"	"pre-placed"
"Male"	"erotic"	"Terms"	"subscription"	"Savage"	"freight"	"Adapted"	"Pathfinder"

ح. ده کلمه برتر بر اساس TF_IDF(W)

"word10"	"word9"	"word8"	"word7"	"word6"	"word5"	"word4"	"word3"
"Railroad"	"Adapted"	"Converted"	"Pathfinder"	"freight"	"Grounds"	"pre-placed"	"parcels"
"Arma"	"DB"	"BR"	"mm"	"parcels"	"km/h"	"handouts"	"pre-placed"
"BR"	"cab"	"Livery"	"GT"	"brake"	"RC"	"Arma"	"Airport"
"off-road"	"Duty®"	"Tyre"	"XR"	"PACK"	"Stella"	"brake"	"RC"
"Terms"	"Conversion"	"resized"	"Adapted"	"JRPG"	"ruleset"	"handouts"	"pre-placed"
"Male"	"erotic"	"Terms"	"subscription"	"Savage"	"freight"	"Adapted"	"Pathfinder"

ط. هیستوگرام تعداد تکرار هر کلمه منحصر به فرد به ترتیب از فرکانس بالا به پایین



لینک github :

https://github.com/bavanDA/NLP_Project

لینک huggingface:

https://huggingface.co/datasets/Bavanda/Steam_DG