به نام خدا

بازیابی پشرفته اطلاعات

دکتر عسگری

تمرین اول

سارا زاهدی

سینا الهی منش

امیررضا قاسمی ویسی

ترک انتخابی :

تشخیص جابجایی انواع وسیله نقلیه.

شرح تمرین :

در این ترک از ما خواسته شده است که مطابق جملات داده شده به نوت بوک بتوانیم مبدا و مقصد و همچنین وسیله ای که با آن جابجایی صورت گرفته را تشخیص دهیم.

راه حل کلی در نظر گرفته شده :

با برسی های به عمل امده بین اعضای تیم روش مورد نظر خود را برای رسید به هدف این تمرین مشخص کردیم :

در نظر گرفتیم که با توجه به این که حرف اضافه هایی مانند از، به، با از نشانه های برای تشخیص جابجایی هستند میتوان با کمک آنها و همچنین داده هایی مانند انواع وسایل نقلیه و انواع مکان ها به هدف تمرین رسید.

مراحل انجام کار به شرح زیر هستند.

مراحل انجام کار :

برای انجام دادن این تمرین ، این تمرین را به سه بخش تقسیم کردیم که در ادامه هریک از مراحل را توضیح میدهیم.

۱. جمع آوری داده های مورد نیاز
۲. انجام پیش پردازش ها
۳. آنالیز نهایی

جمع آوری داده های مورد نیاز :

در این تسک مطابق راه حلی که در نظر گرفته شده بود نیاز بود اطلاعاتی از قبیل اسامی شهر های ایران ، انواع و مدل های مختلف وسایل نقلیه ، انواع شناسه های مربوط به آدرس و مدل های بسیاری از خودرو ها را جمع آوری کنیم.

برای جمع آوری این داده ها ما از سایت های استفاده کردیم که این داده هارا در اختیار ما قرار میدادند.

پیش پردازش متن :

همانطور که در کلاس نیز گفته شد ما برای پردازش یک متن ابتدا نیاز داریم که آن را به مدل مناسبی تبدیل کنیم تا بتوانیم به مدلی پایدار برسیم و بتوانیم به راحتی با داده های تهیه شده عملیات پردازش را انجام دهیم.

برای این امر از همان ابزار هضم که در کلاس معرفی شده بود استفاده کردیم که این کار دارای چندین مرحله بود:

- با استفاده از نرمالایزر متن ورودی را نرمال سازی میکنیم.
- با استفاده از هضم جمله های داده ها را جدا میکنیم.
- همه ی توکن های داده هارا جدا میکنیم.
- حروف اضافه (stop words) را حذف میکنیم.

پردازش :

در بخش راه حل کلی به صورت جزئی در مورد روش پردازش صحبت شد ولی در این قسمت به طور کامل تری درمورد آن توضیح خواهیم داد.

۱. در هر جمله ابتدا به دنبال حروف اضافه کلیدی خود (از، به، با) میگردیم
۲. چک میکنیم که آیا پس از حرف اضافه "از" عضوی از لیست مکان ها جمع آوری شده آورده شده است یا خیر در صورت وجود آن را به عنوان مبدا در نظر میگیریم.
۳. چک میکنیم که آیا پس از حرف اضافه "به" عضوی از لیست مکان ها جمع آوری شده آورده شده است یا خیر در صورت وجود آن را به عنوان مقصد در نظر میگیریم.
۴. چک میکنیم که آیا پس از حرف اضافه "با" عضوی از لیست وسایل نقلیه جمع آوری شده آورده شده است یا خیر در صورت وجود آن را به عنوان مبدا در نظر میگیریم.

در واقع روش پیشنهادی ما استفاده همزمان از کلید واژه های وسایل نقلیه و مکان ها به همراه حروف اضافه مشخص شده است.

در ادامه توضیحات یک مثال نمونه را بررسی میکنیم و صحت عملکرد تمرین خود را میسنجیم :

به عنوان مثل جمله "ما از تهران با پیکان جوانان گوجه ای به مشهَدالرضا رفتیم."

انتظار میرود که الگوریتم ما داده های زیر را خروجی بدهد :

From: "تهران"

To: "مشهدالرضا"

Vehicle: "پیکان"

مرحله به مرحله الگوریتم خود را اجرا میکنیم :

جمله داده شده :

```
In [7]: print(data)
['ما از تهران با پیکان جوانان گوجه ای به مشهدالرضا رفتیم.']
```

بعد از نرمال سازی :

Normalization

```
In [8]: from hazm import *

        normalizer = Normalizer()

        # data_normalized = [[normalizer.normalize(y) for y in x] for x in tqdm.tqdm(datas)]
        data_normalized = [normalizer.normalize(y) for y in data]

In [9]: for x in data_normalized :
            print(''.join(x))
ما از تهران با پیکان جوانان گوجه‌ای به مشهدالرضا رفتیم.
```

الگوریتم پردازش و نتیجه نهایی آن :

Main algorithm

In [18]:
```
result = []
for j in range(len(data_tokens)):
    sentence = data_tokens[j]
    sentence_full = data[j]
    sentence = sentence[0]
    res_dict = {}
    ###
    res_dict["from"] = ""
    res_dict["from_span"] = "[-1, -1]"
    res_dict["to"] = ""
    res_dict["to_span"] = "[-1, -1]"
    res_dict["vehicle"] = ""
    res_dict["vehicle_span"] = "[-1, -1]"
    ###
    for i in range(len(sentence)-1):
        if sentence[i] == "از" or sentence[i] == "مبدا":
            if sentence[i+1] in places:
                res_dict["from"] = sentence[i+1]
                res_dict["from_span"] = spans[j][i]
        if sentence[i] == "به" or sentence[i] == "مقصد":
            if sentence[i+1] in places:
                res_dict["to"] = sentence[i+1]
                res_dict["to_span"] = spans[j][i]
        if sentence[i] == "با" or sentence[i] == "خودروی" or sentence[i] == "ماشین":
            if sentence[i+1] in vehicles:
                res_dict["vehicle"] = sentence[i+1]
                res_dict["vehicle_span"] = spans[j][i]
    result.append(res_dict)
```

In [19]:
```
result
```

Out[19]:
```
[{'from': 'تهران',
  'from_span': [4, 5],
  'to': 'مشهدالرضا',
  'to_span': [37, 38],
  'vehicle': 'پیکان',
  'vehicle_span': [13, 14]}]
```

همانطور که انتظار میرفت و بعد از اجرای الگوریتم و پردازش نهایی که با کمک حروف اضافه و اسامی خاص، ورودی خود را پردازش کردیم و در نهایت به این پاسخ رسیدیم که همان خروجی مدنظر بود.