*921531110*

**علی فاضلی**

**جناب آقای دکتر فیضی درخشی**

|  |
| --- |
|  |
| گزارش پروژه سوم درس مبانی پردازش زبان گفتار |
|  |
|  |

**استاد محترم:**

گزارش پروژه سوم درس مبانی پردازش زبان گفتار

علی فاضلی

**هدف پروژه :**

هدف این پروژه بطور کلی پیاده سازی الگوریتمی برای تشخیص اشتباه های تایپی می باشد. که این هدف بوسیله الگوریتم فاصله ویرایشی یا Edi Distance محقق میشود. زبان برنامه نویسی این پروژه نسخه 3 پایتون می باشد.

برای تحقق هدف این پروژه نیاز به طی کردن مراحل زیر می باشد:

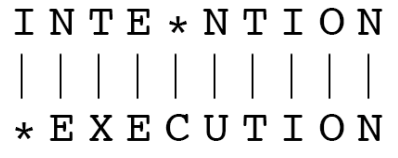
1. پیاده سازی الگوریتم و ساختار داده در زبان برنامه نویسی پایتون
2. تشکیل مجموع داده ای از کلمات صحیح زبان معیار زبان فارسی (و انگلیسی)
3. ادغام مراحل یک و دو و تهییه یک برنامه interactive که کاربر یک توکن را به عنوان ورودی وارد کرده و کلمه صحیح آن به همراه جزییات اشکالات نگارشی (در صورت وجود) به خروجی برده شود.

**فاز اول.پیاده سازی الگوریتم:**

به منظور پیاده سازی الگوریتم فاصله نگارشی، ابتدا لازم است خود الگوریتم بررسی شده، سپس پیاده سازی انجام گیرد.

قبل از هرچیز بایستی یاداور بود که هدف غایی این الگوریتم مقایسه دو کلمه (یا به عبارت کلی تر، توکن) می باشد.

که این دو کلمه عبارت ان از یک کلمه ، که کلمه ورودی ما می باشد و توسط کاربر وارد می شود. و دیگری کلمه صحیح ای است که ما در اختیار داریم. (بوسیله یک دیتاست یا مجوع داده ای). به عنوان مثال اگر کاربر کلمه "مدرشه" را وارد میکند الگوریتم بایستی میزان شباهت این کلمه را به کلمات موجود در مجوع داده ای خود بررسی کرده، کلمه "مدرسه" را به عنوان کلمه صحیح برگردانده، و میزان اختلاف کلمه صحیح با کلمه اشتباه وارد شده را به خروجی ببرد.الگوریتم فاصله ویرایشی از سه عملگر "اضافه کردن" ، "حذف کردن" و جایگزینی برای پیدا کردن میزان شباهت دو کلمه استفاده می کند.



به عنوان مثال در شکل بالا قابل مشاهده است که برای تبدیل کلمه بالایی به پایینی بایستی یک حذف و چهار جانشینی صورت گیرد. به عبارتی فاصله ویرایشی این دو کلمه 5 می باشد.

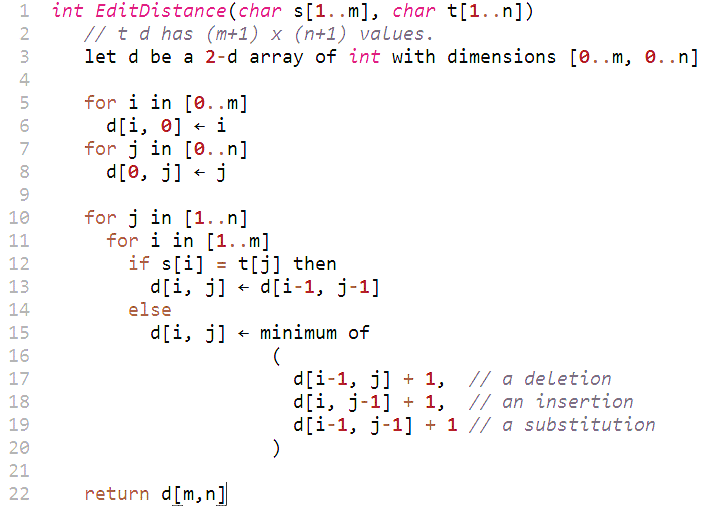
برای بررسی میزان تفاوت دو کلمه بایستی از راهکار و الگوریتم بسیار بهینه ای استفاده نمود چرا که حل اینگونه مسائل در اندازه ورودی بزرگ بسیار حائز اهمیت است. به همین علت از روش برنامه نویسی پویا (Dynamic Programming) استفاده می کنیم.

برای دو رشته فرضی X و Y که به ترتیب به طول n و m می باشند D(I,j) را بدین صورت تعریف می کنیم:

D(I,j) برابر است با فاصله نگارشی بین X[1…i] و Y[1…j]

بدین ترتیب فاصله نگارشی دو کلمه X و Y مقدار D(n,m) می باشد. همانطور که در کلاس اشاره شد برای تعیین فاصله نگارشی از یک جدول (ماتریس) برای محاسبه استفاده میکنیم. مقادیر D(1,J) و D(I,1) برابر تعداد درج های لازم برای ایجاد توکن ای با طول ان کلمه می باشد. بقیه سلول های ماتریس بوسیله الگوریتم برنامه نویسی پویای کمترین فاصله نگارشی پر می شوند.

شبه کد نحوه تکمیل مقادیر درون ماتریس به شکل زیر می باشد:



همانطور که مشخص است ماتریس d ماتریسی می باشد که مقادیر تفاوت نگارشی دو توکن را بررسی می کند.

خطوط 5 الی هشتم مقادیر سط اول و ستون اول جدول را پر میکند. (که واضحا همان مقادیر درج حروف برای ایجاد توکن است)

در خط دوازدهم ، اگر n ام ین حرف از حروف مورد بررسی در دو توکن برابر بود ، عملی انجام نمی گیرد(no operation)

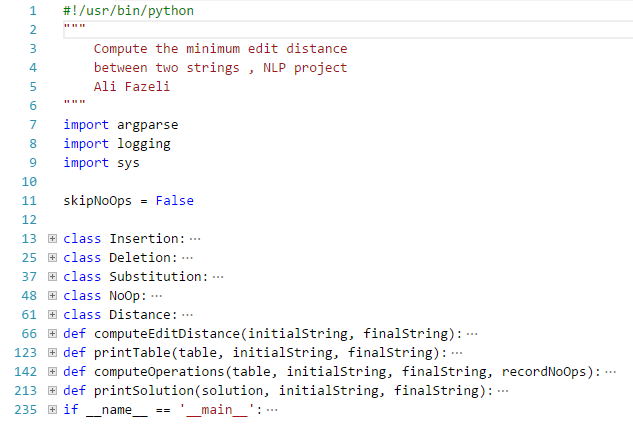
در خط 16 ام، در غیر اینصورت مقدار حداقل هزینه مابین انجام عمل درج، حذف و یا جایگزینی در سلول I,j ام از ماتریس d قرار می گیرد.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | S | a | t | u | r | d | a | y |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **S** | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| **u** | 2 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 6 |
| **n** | 3 | 2 | 2 | 2 | 3 | 3 | 4 | 5 | 6 |
| **d** | 4 | 3 | 3 | 3 | 3 | 4 | 3 | 4 | 5 |
| **a** | 5 | 4 | 3 | 4 | 4 | 4 | 4 | 3 | 4 |
| **y** | 6 | 5 | 4 | 4 | 5 | 5 | 5 | 4 | 3 |

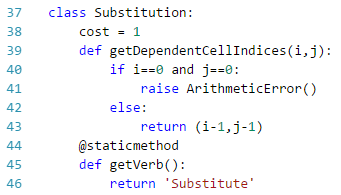
به عنوان مثال در جدول بالا مقایسه دو توکن Sunday و Saturday را داریم. جدول با استفاده از الگوریتم مفروض تکمیل شده. D(m,n) میزان فاصله نگراشی دو کلمه می باشد. اما مسیر حداقل هم در این جدول حائز اهمیت است. بدین منظور که با استفاده از کدام مسیر میتوان کلمه اول را به کلمه دوم تبدیل کرد و از چه ترتیب عملی برای اینکار بایستی استفاده کرد. در جدول بالا مسیر کمترین فاصله نگارشی با کشیدن خط تیره در زیر مقدار فاصله در سلول ها مشخص شده اند.

در پیاده سازی پروژه در وهله اول، ماتریس فاصله نگارشی پیاده سازی شد. سپس مقادیر فاصله نگارشی هر مرحله از بررسی دو توکن در جدول جاسازی خواهد شد ( هر مرحله به n امین شماره Indexer دو توکن در n امین حرف مورد بررسی اشاره می کند.) سپس نوبت به مهم ترین بخش الگوریتم که به نوعی "برنامه نویسی پویا" یا Dynamic Programming می باشد میرسیم. که در یک جدول دیگر (با همان ابعاد جدول فاصله نگارشی و شاخص های آن) محل بهینه ترین عمل ممکن برای تولید آن قسمت از توکن در آن مرحله را در خود نگه می دارد. از این مقادیر در آخر برای بررسی و به خروجی بردن تغییرات لازمه برای یکی کردن دو توکن مورد بررسی استفاده میشود.

با توجه به توضیحات بالا الگوریتم در زبان پایتون پیاده سازی شد. که interface کد و کلاس های بکار رفته در آن به شرح زیر می باشد:



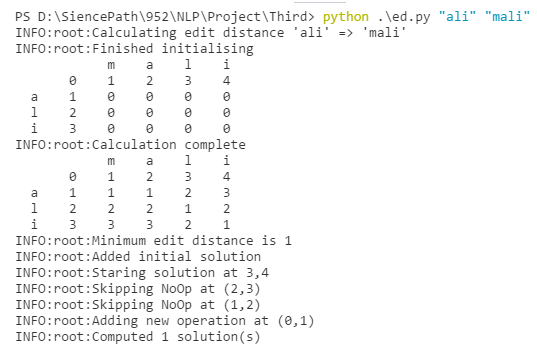
چهار کلاس اول برای پیاده سازی عملگر های اصلی برای تکمیل ماتریس برنامه نویسی پویا می باشد. اعمال درج، حذف، جانشینی و No Operation پیاده سازی شده اند. به عنوان مثال برای عمل جانشینی کلاس زیر پیاده سازی شده است:



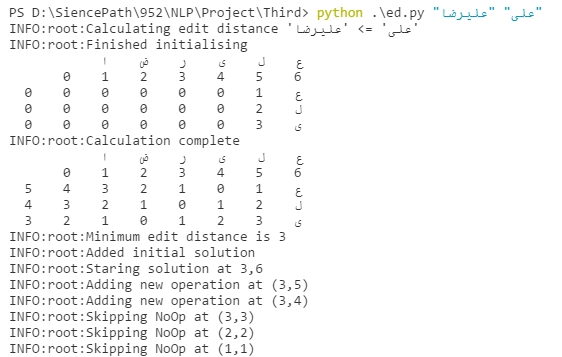
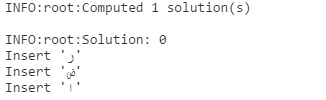
که به عبارتی همان جایگزنی اُریب در جدول فاصله نگارشی (درصورت نابرابری حروف مورد بررسی) می باشد.

تابع printTable() وظیفه چاپ جدول اولیه فاصله نگارشی را بر عهده دارد. تابع computeOperrations() وظیفه محاسبه تعداد اعمال لازم برای رسیدن به مقدار فاصله نگارشی D(m,n) را بر عهده دارد. و در نهایت تابع printSolution() راه حل بهینه کم ترین فاصله نگاری را ببین دو رشته اولیه و رشته مورد بررسی در خروجی چاپ می کند.

به عنوان مثال کُد را برای مقایسه کمترین فاصله نگارشی دو کلمه “ali” و "mali" اجرا کرده و نتیجه زیر را می بینیم:



برای ورودی هایی با زبان فارسی هم مقایسه کلمات "علی"و "علیرضا" را به عنوان آزمایش به عنوان ورودی به کد می دهیم:



همانطور که مشاهده میشود، درکد مشکل راست به چپ (RTL) وجود داشته، اما علیرغم این مشکل اعمال مناسب برای تبدیل دو رشته به یکدیگر بدرستی محاسبه شده است. بدین صورت که برای تبدیل توکن "علی" به "علیرضا" بایستی سه حرف "ر"، "ض" و "ا" به توکن "علی" اضافه بشود.

**فاز دوم.تشکیل مجموع داده ای از کلمات معیار زبان فارسی:**

همانطور که در جلسه شش ام از کلاس مبانی پردازش زبان گفتار اشاره شد، لیست های مجموع داده ای کلمات به دو دسته لیست های باز و بسته تقسیم بندی می شوند. به نحوی که لیست های بسته، لیست هایی از کلمات هستند که در طول زمان تغییری نکرده و ثابت هستند. و لیست های باز به نوعی از لیست های کلمات اطلاق میشود که در طول زمان ممکن است دچار تغییراتی چون تغییر در حروف و یا نحوه ادا کلمه، جایگزینی و یا حذف بشوند.

بدین به تولید دو دسته از لیست های زبان فارسی (اعم از کلمات کهن و معیار) در دو نوع بسته و باز اقدام کرده و دو لیست Closed\_List\_FA.txt و Open\_List\_FA.txt تولید شد که در پوشه فاز دوم موجود است.

در ضمن یک پایگاه داده ای کامل از کلمات مورد استفاده در زبان انگلیسی تولید شد که با نام Word\_database\_EN و با پسوند.list در همان پوشه قابل مشاهده می باشد.

**فاز سوم.پیاده سازی برنامه تعاملی برای تعیین اشتباهات نگارشی با استفاده از الگوریتم فاصله نگارشی:**

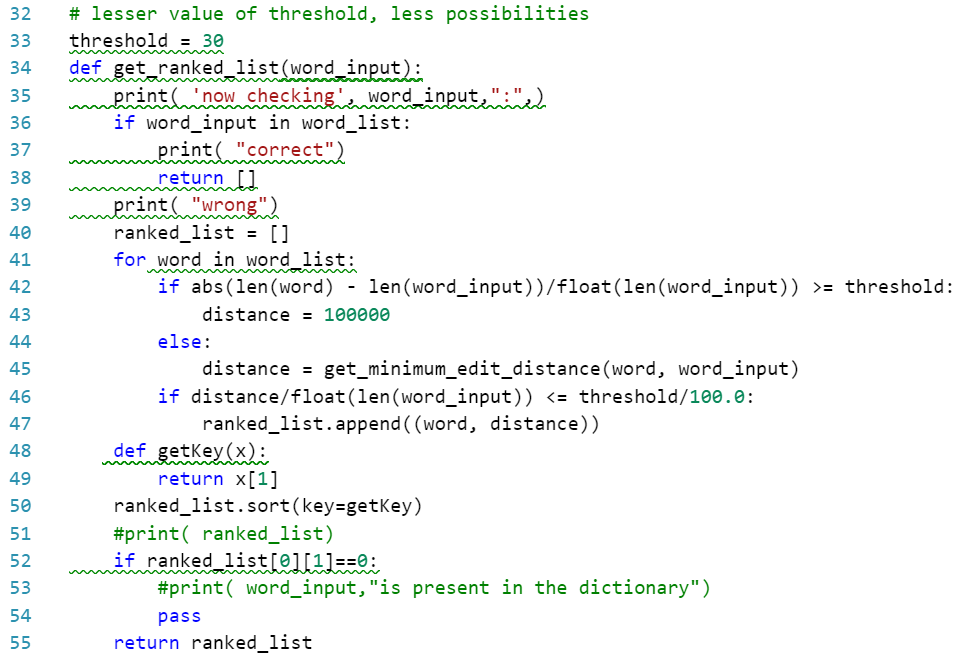
برنامه ای پیاده سازی شد که با استفاده از یک مجوع داده ای از کلمات صحیح به عنوان پایگاه دانش خود، اقدام به شناسایی و تشخیص غلط های نگارشی در متن ورودی، و تصحیح آن بوسیله الگوریتم پیاده سازی شده در مرحله اول می کند.

نام این برنامه interactive\_minimum\_edit\_distance.py می باشد که از الگوریتم فاصله نگارشی برای انجام وظیفه خود استفاده میکند. برنامه همانند پیاده سازی فاز اول پروژه با پایتون نسخه سوم توسعه داده شده است. این برنامه از یک لیست مجموعه کلمات انگلیسی رایج برای تشخیص شیوه صحیح نگارش کلمات (spelling checking) استفاده می کند.

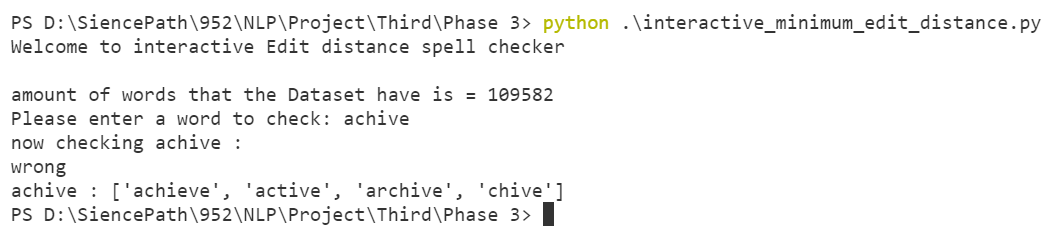
نحوه کار کد به صورتی است که بصورت تعاملی کلمه ای را از کاربر می گیرد. در ابتدا بررسی میکند که آن کلمه در لیست مجموع داده ای خود قرار دارد یا نه. اگر کلمه در لیست داده ای موجود بود، آن را به عنوان کلمه صحیح (منظور از صحیح در این متن به معنای صحت نگارشی می باشد) بر میگرداند. در صورت وجود نداشتن کلمه در مجموع داده ای، الگویتم فاصله نگارشی برای آن کلمه و کلمات مشابه آن در مجموع داده ای محاسبه می شود. و چندی از کلمات موجود در مجموع داده ای که کمترین فاصله نگارشی را با کلمه وارد شده توسط کاربر دارند به عنوان پیشنهاد برنامه به خروجی برده می شوند.

توجه داشته باشید که میزان انتخاب کلمات مشابه مجموع داده ای را به حد آستانه ای بنام threshold وابسته میکنیم. بدین معنی که با کاهش میزان این عبارت میزان کلمات مشابه حدس زده شده برای کلمه ورودی کاهش می یابد. (و برعکس) . حالت پیشفرض برای threshold مقدار 30 مناسب بنظر می رسد.

تابع get\_ranked\_list کلمه وارد شده توسط کاربر را به عنوان ورودی گرفته، چک می کند که کلمه در Dataset موجود هست یا خیر. در غیر اینصورت مقدار حد آستانه را با فاصله مطلق طول کلمه ورودی و کلمات مشابه موجود در لیست (که با استفاده از جست و جوی ترتیبی انتخاب شده اند) مقایسه میکند. اگر از حد آستانه بیشتر باشد کلمه حدس زده شده غیرمفید طلقی شده و از دور خارج می شود. در غیر اینصورت با استفاده از الگوریتم فاصله نگارشی فاصله نگارشی کلمات حدس زده شده محاسبه شده و با کلمه وارد شده مقایسه میشود. در اخر این کلمات حدس زده شده بصورت صعودی در لیست ranked\_list ذخیره شده و به خروجی میروند. در ادامه تصویری از نحوه اجرای کد آورده شده است.



در اخر به عنوان کار های آتی میتوان به پیاده سازی همین الگوریتم برای مجموع داده ای فارسی اشاره کرد.



منظور از کلمه وارد شده کلمه achieve بوده که به اشتباه وارد شده. همانطور که مشاهده می کنید نحوه نگارش صحیح آن به خروجی برده شده است.

یک مورد هم از نحوه اجرای یک کلمه صحیح به شکل زیر است:

