فایل ضمیمه : hw3.ipynb

مجموعه داده های آموزشی را به صورت دوتایی های (word, tag) از هر خط فایل Train.txt خواندم و در لیست train_tagged قرار دادم. روی این لیست پارامترهای مدل آموزشی را محاسبه خواهم کرد.

گام اول این است که تگ ها را جدا کنم و بدانم در داده ها ی آموزشی از چه برچسب‌هایی استفاده شده. مجموعه ی کل برچسب ها ۲۳ عنصر دارد و شامل موارد زیر است :

```
tags = {V, AR, DET, PRO, MS, MQUA, P, OH, INT, DEFAULT,
PS, CON, IF, DELM, ADJ, OHH, SPEC, QUA, NP, ADV, MORP,
N, PP }
```

برای الگوریتم ویتربی دو محاسبه از روی داده های آموزشی نیاز هست. یکی احتمال دو تک داده شده پشت سرهم و دیگری احتمال یک کلمه به ازای تگ داده شده.

برای محاسبه ی اولی یعنی p(tag1|tag2) برای هر tag1, tag2 که عضو tags باشد. این احتمالات در هرگام از الگوریتم ویتربی به ازای اینکه برچسب فعلی و قبلی آن چه باشند مورد استفاده قرار می‌گیرد ولی مقدار آن ها بر اساس داده های آموزشی تعین می‌شود و در طول کار تغیری نمی‌کند. این مقادیر را در یک ماتریس ۲۳*۲۳ میتوان از طریق شمارش تمام برچسب‌های پشت سرهم در دوتایی های آموزشی ذخیره کرد. محاسبات درایه های این ماتریس در تابع p_t2_given_t1 انجام شده. ماتریس ایجاد شده را به صورت یک pandas.DataFrame به نام tags_df ایجاد می‌کنم.

برای محاسبه ی دوم باید p(word|tag) را بر اساس کلمات مجموعه‌ی تست و تگ‌هایی که در هر گام از ویتربی روی آن قرار دارد محاسبه کنم. این مقادیر قبل از اجرای الگوریتم به ازای همان هر زوج کلمه و برچسب که در مجموعه ی تست وجود دارد از روی مجوعه آموزشی با تابع word_given_tag محاسبه می‌شود. این تابع دو مقدار را روی داده های آموزشی شمارش می‌کند، مواردی که زوج کلمه-برچسب مورد نظر رخ داده باشد و مواردی که برچسب به تنهایی رخ داده باشد. این اطلاعات روی یک pandas.DataFrame به نام test_words_df ذخیره می‌شود.

تولید دو ماتریس بالا در واقع بخش آموزش مدل هستند. برای یک مدل مخفی مارکوف دو ماتریس بالا در واقع ماتریس A و بخشی از ماتریس B هستند. تولید این دو یا همان شمارش زوج ها روی مجموعه بیشترین زمان از اجرای این کد را به خود اختصاص می‌دهد.

الگوریتم ویتربی پیاده شده در تابع Viterbi برای هر کلمه از مجموعه آموزشی یک گام مسیر را محاسبه می‌کند. احتمالات بیشتر را انتخاب و نگهداری می‌کند تا به پایان مسیر (پایان کلمات) برسد. سپس مسیری را

که بیشترین احتمال را به دست دهد انتخاب و خروجی می‌دهد. این دنباله از state ها در واقع حاصل برچسب گذاری کلمات ورودی هست.

داده‌های تست را در از فایل Test.txt خط به خط در قالب دو لیست می‌خوانم . یکی test_untagged هست که شامل دنباله ی کلمات تست بدون برچسب هست. دومی test_tagged هست که مشابه لیست داده های آموزشی از دوتایی‌های کلمه-برچسب تشکیل شده.

test_untagged را به عنوان ورودی به Viterbi می‌دهم. مدت زمانی طول می‌کشد تا دنباله‌ی برچسب‌های خروجی ساخته شوند و نهایتا در tagged_sequence ذخیره می‌شوند. برای مقایسه ی تگ های واقعی با تگ های تولید شده دوتا لیست tagged_sequence و test_tagged را مقایسه و نتایج درست را شمارش می‌کنم. معیار accuracy به دست آمده روی کل داده های تست 82 درصد است.