Farshad Borjalizade

HW4 - Classification for Person Name Detection

Natural Language Processing

University of Tehran

Mordad 1400

گزارش کار انجام شده

هدف اصلی در این تمرین پیاده سازی یک طبقه بند کننده (classifier) ساده برای شناسایی اسامی اشخاص و افراد است. برای لیبل گذاری اسامی افراد از تابع predict() که تک تک توکن ها را به همراه index آنها میگیرد استفاده می کنیم حال به تکنیک ها و ویژگی های در نظر گرفته شده می پردازیم :

۱. **استفاده از خلاصه سازی به وسیله pattern گذاری بر روی توکن ها :** از آنجایی که اسامی افراد با حروف بزرگ شروع می شود با استفاده از تعریف الگوهایی سعی در شناسایی این اسامی داریم به این صورت که حروف بزرگ به A ، حروف کوچک به a و اعداد به 0 و دیگر کاراکترها به خودشان نگاشت (maping) کردیم برای مثال Farshad_Borjalizade به طور خلاصه نگاشت می شود به Aa_Aa و 1400 نگاشت می شود به 0.

۲. **استفاده از وجود Quotes :** اگر نشانه "" در جمله ای باشد در اکثر اوقات بدین معناست که شخصی در حال صحبت کردن است که اگر این گونه باشد، احتمالا در آخر جمله اسم آن شخص ظاهر خواهد شد.

۳. **موقعیت اولین و آخرین کلمه :** در اکثر متون حرف اول کلمه اول به صورت بزرگ نوشته می شود، از این دو ویژگی هم استفاده کردیم.

۴. **شمارش تعداد کلمات ظاهر شده :** تعداد دفعاتی که کلمه در جمله تکرار شده است را می توان به عنوان یک معیار برای اینکه آن کلمه چقدر احتمال دارد برچسب اسم را بخورد باشد و تعداد تکرار را به یک، دو و دو بار بیشتر تقسیم می کنیم.

۵. **کلمه فعلی :** از خود کلمات هم می توان ویژگی هایی را استخراج کرد.

۶. **طول کلمات** : کلمات را به چندین دسته بر اساس طولشان به بازه های کوچک تر از ۵ بزرگ تر از ۲۰ و ما بین بازه های ۱۶ تا ۲۰ و ۶ تا ۱۰ و ۱۱ تا ۱۵ تقسیم کردیم، بر اساس تجربه ثابت شد که با انجام دادن این کار چند درصدی افزایش در دقت مدل خواهیم داشت.

۷. **لغات قبلی و بعدی** : استفاده از لغات بعدی و قبلی هم می تواند برای پیش بینی اسامی بسیار مفید باشد و این کار را با استخراج pattern ها و با توجه به index کلمه فعلی می توان به سه قسمت متفاوت تقسیم کرد:

۱. اگر اولین کلمه باشیم فقط از کلمه بعدی استفاده می کنیم.
۲. اگر آخرین کلمه باشیم فقط از کلمه آخر استفاده می کنیم.
۳. اگر نه کلمه اول و نه کلمه آخر باشیم در این حالت می توانیم از هر دو سمت کلمه استفاده کنیم.

۸. **کلمات جدید** : یک حالت منحصر به فردی را هم در نظر می گیریم برای زمان هایی که طبقه بند کننده با کلمه ای رو به رو شود که در زمان آموزش با آن مواجه نشده است در این صورت از ویژگی کلمه جدید (new word) هم برای آن کلمه استفاده می کنیم.

با وزندهی تصادفی به ویژگی ها و استفاده از تابع بهینه ساز SGD با ۶۰ بار تکرار اجازه می دهیم که مدل آموزش ببیند.

```
weights = np.random.randn(num_features) * 0.001

grad_ascent = SGDOptimizer(weights,alpha)
num_epochs = 60
```

در جدول زیر خلاصه ای از نتایج حاصل شده را می توان دید.

	Train	Test
Accuracy	204289/204567 = 0.9986	51050/51578 = 0.9897
Precision	10594/10761 = 0.9844	2747/2954 = 0.9299
Recall	10594/10705 = 0.9896	2747/3068 = 0.8954
F1	0.9870	0.9123
Data reading and training took 312.365001 seconds		