«به نام پروردگار»

گزارش پروژه سوم مبانی هوش مصنوعی

«کلاس بندی مصرع های شعرا: nlp »

استاد: دکتر بهنام روشنفکر

دانشجو: محسن محمدیان

شماره دانشجویی: 9831502

بررسی اجمالی:

در این پروژه هدف آن هست که یک دسته بندی یا classification داشته باشیم. به اینصورت که سه پیکره آموزشی از سه شاعر بزرگ ایرانی مانند مولانا و حافظ و فردوسی داده خواهد شد که بوسیله آنها مدل یاد خواهد گرفت چگونه مصرع ها را تشخیص دهد و در فاز تست تشخیص دهد که هر مصرع مختص کدام شاعر است.

شیوه پیاده سازی:

برای پیاده سازی این پروژه از مدل های Unigram و Bigram تنها استفاده شده و از مدل trigram استفاده نشده است. برای پیاده سازی اینگونه عمل شده که برای هر شاعر مدل unigram و bigram آن ساخته شده و برای جلوگیری از وقوع احتمال صفر، در فاز تست از backoff مدل بهره برده شده است.

توابع پیاده سازی شده:

main():

تابع اصلی ما که در آن تابع train_poets_model() فراخوانی می شود و مقدار برگردانده شده ی آن، که یک دیکشنری ای از دیکشنری شعرا می باشد، در متغیر poet_probs ریخته می شود.

سپس تابع read_test_data() فراخوانی می شود تا داده ی تست را بخوانیم.

و در نهایت به ازای پارامترهای مختلف، مدل ما چک می شود(یا به عبارتی مدل های مختلفی با دقت های مختلف خواهیم داشت، که آن مدلی که دقت بیشتری دارد، مسلما بهتر خواهد بود)

read_data(address):

data را به صورت text از آدرسی که به آن داده شده است، می خواند و return می کند.

read_test_data(addr):

برای خواندن فایل تست ما می باشد و دوتا لیست را بر می گرداند. Hemistich و poets_names که لیست اول شامل مصراع های موجود در فایل تست می باشد و لیست دوم نام شاعر متناظر با آن مصراع را در index متناظر ذخیره می کند.

Cal_word_vocab(lines):

این تابع خطوط موجود در یک فایل آموزشی را می گیرد(فایل یک شاعر) و می شمارد که هر کلمه چندبار تکرار شده است و در نهایت یک دیکشنری بر می گرداند که کلید آن خود کلمه و مقدار آن تعداد تکرار کلمه می باشد.

Non_repititive_words_removal(vocab, min_count=2):

این تابع می آید و در vocabulary ما آن کلماتی که مقدار کمتر از دو را دارند، (کمتر از دو بار تکرار شده اند) را حذف می کند.

:Bigram_counter(lines)

این تابع تعداد تکرار جفت کلمه ها را در پیکره ی آموزشی شمارش می کند (برای مثال می شمارد چندتا a|b در پیکره ما وجود دارد) و یک دیکشنری به نام _2units_count بر می گرداند که کلید آن ("کلمه|کلمه") می باشد و مقدار آن تعداد تکرار آن جفت کلمه.

calculate_bigram_probs(_2units_count, numberOfRepitionOfWords_dict, numberOfLines):

این تابع دیکشنری تعداد تکرار جفت کلمه ها و دیکشنری تعداد تکرار هر کلمه و تعداد خطوط موجود در فایل آموزشی را گرفته و به کمک آنها احتمالات Bigram را محاسبه می کند و یک دیکشنری بر می گرداند که کلید آن ("کلمه|کلمه") و مقدار آن احتمال این جفت کلمه می باشد.

:Make_model(data)

این تابع اگر بخواهیم بگوییم چه کاری می کند، خلاصه ی آن این است که مدل Unigram و Bigram پیکره آموزشی را بر می گرداند.

این تابع برای هر شاعر یک دیکشنری می سازد به نام poet_model، که کلیدهای Unigram و Bigram را دارد و مقادیر این دیکشنری خود دیکشنری می باشند. دو دیکشنری هستند که یکی احتمالات Unigram آن پیکره را در بر می گرد و دیگری احتمالات Bigram که در تابع calculate_bigram_probs محاسبه شده است.

:Train_poets_model()

این تابع خطوط پیکره آموزشی را می خواند و به عنوان data به تابع make_model می دهد. پس از آنکه تابع make_model تعداد کلمات موجود در آن فایل یا پیکره و مدل شاعر را برگرداند، این تابع یک دیکشنری می سازد به نام model که مدل کلی ما می باشد و شامل نام هر شاعر با احتمالات Unigram و Bigram آن می باشد. در نهایت یک فیلد دیگر به این دیکشنری اضافه می شود که کلید آن "poetProb" هست و مقدار آن یک دیکشنری که سه کلید hafez و molavi و fedowsi را دارد و مقادیر آنها احتمال شهودی آنها می باشد که از تقسیم تعداد کلمات هر فایل به تعداد کلمات کل سه فایل به دست می آید.

:Unigram_probs_calculator(hemistich, unigram_model)

این تابع یک مصرع را می گیرد و احتمال آن را به وسیله مدل Unigram مربوطه حساب می کند و بر می گرداند.

:backOff_prob_calculator(hemistich, model, l1, l2, l3)

این تابع یک مصرع را می گیرد و احتمال آن را با BackOff مدل حساب می کند.

:Test_model(model, hemistich, poets_name, l1, l2, l3, eps)

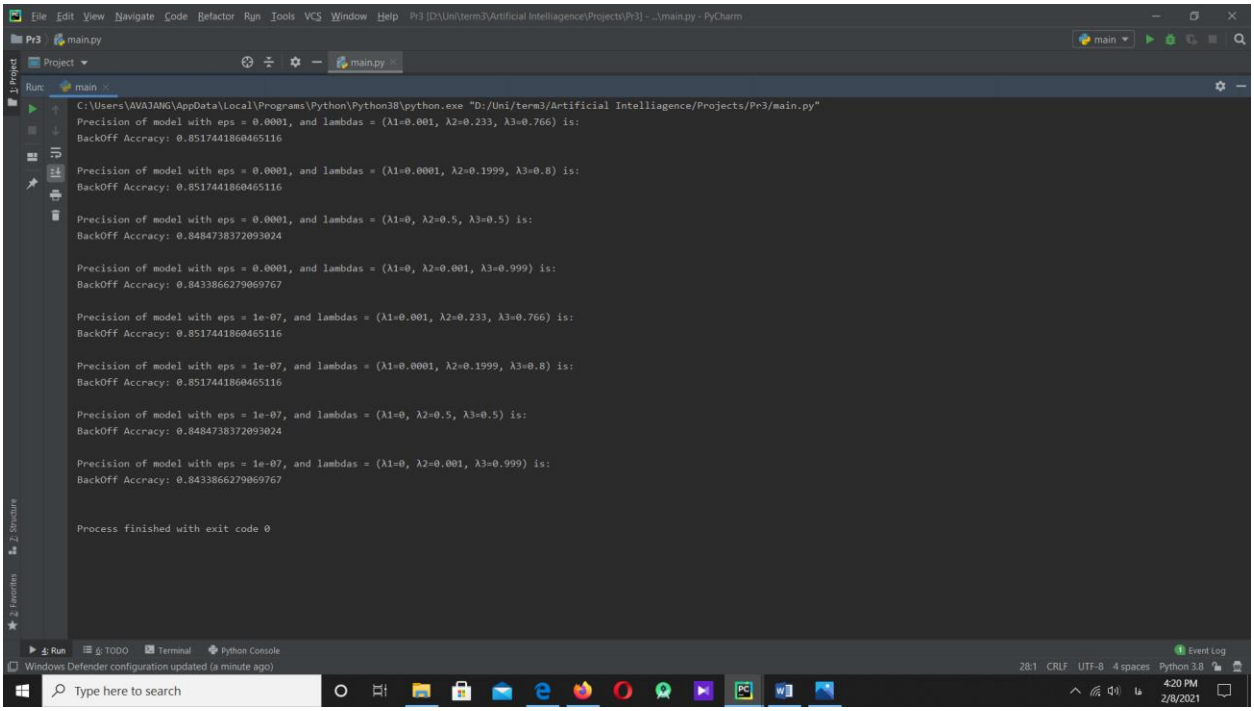
این تابع در main فراخوانی می شود و متغیر های ورودی را از Main می گیرد و اینگونه کار می کند که به ازای هر مصرع می آید در دیکشنری model یک مدل یک شاعر را بر می گزیند و با احتمالات Unigram و

Bigram تعریف شده برای آن شاعر و به کمک تابع backOff_prob_calculator احتمال آن مصرع را حساب می کند و اگر مقدار احتمال حساب شده بیشتر ازmax_prob_backoff بود، آن مقدار را در متغیر max_prob_backoff می ریزد و کلید دیکشنری مدل که نام شاعر هست را در متغیر predicted می ریزد و سپس برای دو شاعر دیگر این کار را تکرار می کند.

بنابراین آن شاعری که مقدار احتمال بیشتری را برای یک مصرع بر می گرداند، شاعر آن مصرع برگزیده خواهد شد. در نهایت شاعری که مدل ما برگردانده، با شاعر اصلی مصرع مقایسه می شود و اگر این دو یکسان بودند، یک مقدار به متغیر b_true_predicted اضافه خواهد شد و اینگونه می توان دقت کلی مدل را نیز تشخیص داد.

دقت تشخیص برنامه به ازای پارامترهای مختلف:

نتیجه ی برخی از پارامترهای تست شده در تصویر زیر قابل مشاهده است.



همانطور که میدانیم بسیار بعید است که احتمالات Unigram یک مدل صفر باشند. اما برای Bigram احتمالات صفر بیشتر خواهد بود و برای اپسیلون هم که هیچگاه صفر نخواهیم داشت چراکه بین 0 و 1 می باشد.

روش BackOff به ما می گوید که در مواقعی که احتمال Bigram صفر شد به مقدار Unigram آن احتمال BackOff کنیم و اگر Unigram آن نیز صفر شد، به مقدار اپسیلون بسنده کنیم.

پارامترهای ما $\lambda_3, \lambda_2, \lambda_1, \varepsilon$ می باشند. که لاندا ها به ترتیب ، ضرایب Bigram و Unigram و Epsilon در روش هموارسازی BackOffهستند.

$$\hat{P}(c_i : c_{i-1}) = \lambda_3 P(c_i | c_{i-1}) + \lambda_2 P(c_i) + \lambda_1 \varepsilon$$

می دانیم که مجموع سه لاندا باید برابر با یک باشد؛ هر کدام از لاندا های ما می توانند یک مقدار fix باشند یا توسط الگوریتم maximization بدست آیند.

همچنین این وزن دهی ما باید به گونه ای باشد که اگر برای مثال تعداد زیادی از احتمالات ما Bigram بود، ضریب Bigram را به نسبت سایر ضریب ها مقدار بیشتری می دهیم، اگر تعداد Bigramهای ما کم بود، ضریب Unigram و اپسیلون را به نسبت مقدار بزرگتری می دهیم.

درنهایت می دانیم هرچقدر dependency بیشتری درنظر بگیریم، مدل منطقی تری خواهیم داشت بنابراین چون Bigram دقیق تر از Unigram عمل می کند، پس اگر ضریب آن بزرگتر باشد، مدل ما دقیق تر خواهد بود. و در نهایت باتوجه به نسبت وزن Bigram و Unigram، بهترین پارامترهای ما برابر با

$$\lambda_3 = 0.8, \qquad \lambda_2 = 1999, \qquad \lambda_1 = 0.0001, \;\; \varepsilon = 0.0001$$

خواهند بود.

باسپاس از توجه شما