

توضیحات:

ابتدا از فایل 'Train_data.txt' جملات داده شده را خوانده، سپس با استفاده از تابع `word_tokenize`، کلمات هر جمله را استخراج می کنیم. تمام کلمات را در یک `set` نگهداری می کنیم. تعداد هر کلمه به طور مجزا، زوج کلمه و ۳ تایی بررسی شده و ذخیره می شوند.

برای هر چندتایی از کلمات یک تابع محاسبه احتمال با فرمول های زیر در نظر گرفته شده است:

• unigram: M تعداد کل واحدها در پیکره

$$P(c_i) = \frac{\text{count}(c_i)}{M}$$

• bigram

$$P(c_i|c_{i-1}) = \frac{\text{count}(c_{i-1}c_i)}{\text{count}(c_{i-1})}$$

• trigram

$$P(c_i|c_{i-2:i-1}) = \frac{\text{count}(c_{i-2}c_{i-1}c_i)}{\text{count}(c_{i-2}c_{i-1})}$$

تابع `guess_next_word` کلمه ی جا افتاده را بر اساس کلمات ذخیره شده حدس می زند؛ با استفاده از این تابع، به جای \$ یک کلمه را پیشنهاد می دهد.

$$\hat{P}(c_i|c_{i-2:i-1}) = \lambda_3 P(c_i|c_{i-2:i-1}) + \lambda_2 P(c_i|c_{i-1}) + \lambda_1 P(c_i)$$

$$\lambda_3 + \lambda_2 + \lambda_1 = 1$$

$$\lambda_1 = 0.000013, \quad \lambda_2 = 0.00987, \quad \lambda_3 = 0.990117$$

اعداد تصادفی انتخاب شده اند و برای واقعی تر بودن نتایج ضریب Trigram از بقیه بیشتر است.

سپس نتایج در فایل Test_data_result ذخیره شده و با فایل labels مقایسه می شوند.

دقت مدل امتحان شده: 0