توضیحات:

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

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

- unigram: M تعداد کل واحدها در پیکره

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

- bigram

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

- trigram

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

تابع guess_next_word کلمه‌ی جا افتاده را بر اساس کلمات ذخیره‌شده حدس می‌زند.

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

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

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

اعداد تصادفی انتخاب شده‌اند و برای واقعی‌تر بودن نتایج ضریب Trigram از بقیه بیشتر است.

جملات ورودی از فایل Train_data خوانده‌شده و تابع guess_next_word روی $ اعمال می‌شود؛ سپس نتایج نمایش داده شده، در فایل Test_data_result ذخیره‌شده و با فایل labels مقایسه می‌شوند.

دقت مدل امتحان‌شده: 0