برای تمام حالات خواسته شده در صورت سؤال توکن ایجاد شده و این توکن‌ها در پوشه‌ی tokens موجود است.

همچنین برای تمامی حالات جملات از مدل‌های ۱-گرم تا ۵-گرم در پوشه‌ی Generated Sentences ذخیره شده است.

در فایل main.py کلاسی نوشته‌شده است با گرفتن متن train و شماره‌ی بالاترین n برای ساختن مدل، اقدام به ایجاد مدل‌های ۱-گرم تا n-گرم با شرایط خواسته‌ شده ( شمارش علایم نگارشی، استفاده از لم یا توکن و استفاده از توکنایزر ساده یا هوشمند) می‌‌کند. برای مثال مدل‌هایی که در خط آخر main.py ایجاد می‌شوند با شرایط استفاده لم به جای توکن، عدم شمارش علایم نگارشی و توکنایزر نوشته شده توسط ما مدل‌های ۱ تا ۵ گرم را ایجاد و احتمال‌های هر کدام را محاسبه‌ می‌کند.

در خط آخر این کد perplexity مدل ترایگرم روی فایل تست به عنوان مثال چاپ می‌شود.

اجرای این کد با شرایط مختلف روی داده‌ی تست به نتایج زیر منجر شد:

1. استفاده از لم به جای توکن perplexity مدل را کاهش می‌دهد و در نتیجه‌ی احتمال‌های بهتری به دست ‌می‌آید.
2. توکنایزر ما نتایج بهتری نسبت به توکن کردن فقط از روی فاصله می‌دهد.
3. نشمردن علایم نگارشی نیز perplexity را کاهش داده احتمال‌های ما را بهبود می‌بخشد.
4. با توجه به تفاوت زیاد متن تست و ترین، استفاده از n-گرم‌های بالاتر نتیجه‌ی بدی روی perplexity می‌گذارد. علت این امر یافت نشدن n-گرم‌ها در مدل و تأثیر منفی smooth کردن مدل‌های اسپارس می‌باشد. برای دیدن این تفاوت باید به این امر توجه کرد که در صورت محاسبه‌ی perplexity روی داده‌ی train این عدد با افزایش n شدیداً به ۱ نزدیک می‌شود ولی در مورد داده‌ی test به خاطر smoothing اینگونه نیست. Smoothing استفاده شده به صورت unknown word می‌باشد چون به عدم دسترسی به کل n-گرم‌هایی که ممکن است در داده‌های تست باشند امکان استفاده از add-1 smoothing وجود ندارد.