Sindokht

Compiler Project Report - Phase 1 - Winter 2017

Benyamin Delshad - Mohammad Mahzoun

03.16.2017

COMPUTER SCIENCE DEPARTMENT

UNIVERSITY OF TEHRAN

TEHRAN, IRAN

مقدمه

پروژه‌ی درس کامپایلر در ترم جاری(پاییز ۹۵ و بهار ۹۶) عبارت است از ساختن یک قصه‌گو که بتواند داستان‌های نسبتا معقولی را تولید کند و بخواند. ما نام این پروژه خود را "سیندخت" نهادیم و دلیل این نام‌گذاری آن است که در فرهنگ پارسی زبان‌ها، عموما این مادربزرگ‌ها هستند قصه برایمان می‌گویند و سیندخت مادربزرگ رستم است و رستم بزرگ‌ترین پهلوان اساطیری ایرانیان.

در فاز نخست سیندخت، او باید بتواند بر حسب داستان هایی که در ورودی می‌گیرد، در خروجی داستانی را تولید نماید که در ادامه الگوریتم این کار، شرح داده می‌شود.

زبان مورد استفاده‌ی ما در سیندخت، پایتون ۲,۷ است.

ابزار ها و کتابخانه‌های به کار رفته

1. HAZM 0.5 : ازین کتابخانه برای نرمالایز کردن و توکنایز کردن داستان های ورودی استفاده می‌شود.
2. re : ازین کتابخانه برای ویرایش های ساده ای در انتها که متن نهایی ساخته می‌شود، استفاده می‌گردد.
3. git : ازین ابزار، در کار تیمی خود، برای مدیریت بهتر ازین ابزار بهره بردیم.

الگوریتم

روش کار ما، بسیار مشابه با آن چه در کلاس حل تمرین یا در فایل توضیح فاز نخست پروژه گفته شد، می‌باشد. در ابتدا، متن ورودی را از آدرس داده شده، استخراج می‌کنیم و در ابتدا آن را نرمالایز می‌کنیم. کار هایی از قبیل: حذف فاصله های تکراری اضافی، یکی کردن انواع مختلف حروفی از فارسی که کد های مختلفی در Unicode دارند مثل انواع 'ک' ها و 'ی' ها، بعضی فاصله ها که میان یک کلمه هستند را تبدیل کردن به نیم‌فاصله تا در جداسازی توکن‌ها از هم جدا نگردند مثل 'می باشد' که یک کلمه است ولی اگر با فاصله توکنایز کنیم به دو کلمه تبدیل می‌شود پس در عملیات نرمالایز به 'می‌باشد' تبدیل می‌گردد. و ... . یکی از دلایل استفاده از کتابخانه HAZM نیز هندل‌کردن راحت این موارد بود. سپس به توکنایز کردن آن می‌پردازیم. این کار دیگر به راحتی با دلیمیتر فاصله قابل انجام است.

در مرحله بعدی، n-گرام ها را می‌سازیم و به جای احتمال‌ها از فرکانس یا همان تعداد تکرار استفاده می کنیم. این اعداد صحیح مثبت، بعد تر وزن های ما می‌شوند در رندوم وزن‌داری که می‌زنیم. در واقع به ازای هر n-1 توکن متوالی وزن هر توکن برای بعد این n-1 تا بودن را داریم و با رندوم وزن‌دار کلمه بعدی را می‌سازیم و به پیش می‌رویم. برای شروع هم یک n-1

تایی رندوم اول کار می‌گذاریم که قبل‌تر به همین صورت متوالی در رشته ورودی ظاهر شده است.

در نهایت نیز یک متن حداکثر N جمله ای می‌سازیم که مقدار دیفالت N صد است و معیار هر جمله، نقطه، علامت سوال یا علامت درنگ است.

در انتها نیز متن خروجی را کمی می‌پردازیم. فاصله های زاید آن را می‌زداییم و در فایل خروجی می‌ریزیم. فایل Prob.txt نیز مقادیر فرکانس های صحیح ما را نگه می‌دارد.

نحوه کار کردن با برنامه

همان‌طور که در فایل StoryGenerator.py دیده می‌شود، این برنامه پس از اجرا از شما یک آدرس برای داستان ورودی درخواست می‌کند. در این قسمت می‌توانید از داستان‌های نمونه‌ای که در پوشه‌ی Docs قرار دادیم نیز به عنوان فایل داستان ورودی استفاده نمایید.

مقدار دیفالت N و حداکثر تعداد جملات (نه کلمات!) را می‌توانید در تابع مربوطه نیز تغییر دهید ولی از پیش ۴ و ۱۰۰ تعیین شده اند.

در انتها پس از وارد نمودن آدرس فایل داستان ورودی، داستان تولید شده در فایل Story.txt در کنار فایل‌های پایتون پروژه ساخته می‌شود.

درباره‌ی ما

ما یک تیم دو نفره برای انجام این پروژه هستیم و متشکل از: بنیامین دلشاد ممقانی و محمد محزون با شماره های دانشجویی ۶۱۰۳۹۳۰۹۳ و ۶۱۰۳۹۳۱۳۵. هر دو از ورودی های ۹۳ علوم کامپیوتر دانشگاه تهران هستیم.