به نام خدا

دانشکده علوم و فنون نوین

آشنایی با زبان‌شناسی رایانشی (۱۲۱-۰۵-۸۳)
نیم‌سال دوم ۱۴۰۱-۱۴۰۰

تاریخ تحویل: ۱۴۰۱/۰۳/۱۳

تمرین شماره ۳

---

۱. (۲۰٪) **[آشنایی با روش‌های نوین مدل زبانی]** برای استخراج مدل‌های زبانی، علاوه بر روش آماری N-gram، روش‌های جدیدی در سال‌های اخیر مبتنی بر شبکه‌های عصبی ارائه شده است که عملکرد خوبی را داشته‌اند (مانند روش‌های مبتنی بر RNNها، BERT و ...). در این سوال مروری بر برخی روش‌های این حوزه صورت بگیرد و به صورت خلاصه ایده روش‌ها و مقایسه کارایی آنها را بیان کنید. توجه شود که در این سوال انتظار نمی‌رود بر مفاهیم و جزئیات خود شبکه‌های عصبی مسلط باشید.

---

۲. (۸۰٪) **[پیاده‌سازی: مدل‌سازی زبانی]** هدف این تمرین پردازش متون فارسی به منظور استخراج مدل زبانی N-gram می‌باشد. برای این کار از دادگان **زبرا** که در تمرین شماره ۱ ارائه شد، استفاده کنید و برای بخش‌های زیر یک برنامه بنویسید. ابتدا با استفاده از توابع نوشته شده در تمرین‌های قبلی (CorrectPuncs، CorrectCoding و CorrectVariations)، دادگان را نرمال کنید و فایل ZebraAllNormalized را تولید کنید. سپس مشابه تمرین‌های قبل، در هر بخش از تمرین، خروجی برنامه را به همراه کد برنامه نوشته شده و گزارش آن، ارسال کنید.

در این تمرین فرض شده است از تمرین‌های قبل، توابع زیر را اختیار دارید.

توابع تمرین‌های شماره ۱ و ۲
NoAllWords , NoAllUniqWords, NoEachUniqWord = **CountWordsFile**(TextFileName) تابعی که فایل متنی (TextFileName) را گرفته و تعداد کل کلمات (NoAllWords)، تعداد کل کلمات یکتا (NoAllUniqWords) و تکرار هر کدام از کلمات یکتا (NoEachUniqWord) را به عنوان خروجی برگرداند.
NoAllWordsPerFolder , NoAllUniqWordsPerFolder, NoEachUniqWordPerFolder, NoEachUniqWord = **CountWordsFolder**(Path) تابعی که مسیر (Path) حاوی یک یا چند پوشه را گرفته و تعداد کل کلمات هر پوشه (NoAllWordsPerFolder)، تعداد کل کلمات یکتا در هر پوشه (NoAllUniqWordsPerFolder)، تکرار هر کدام از کلمات یکتا در هر پوشه (NoEachUniqWordPerFolder) و تکرار هر کدام از کلمات یکتا در کل پوشه‌های موجود در مسیر داده شده (NoEachUniqWord) را به عنوان خروجی برگرداند.
OutFile = **CombineFiles** (Path) تابعی که مسیر (Path) را گرفته و تمام فایل‌های موجود در آن را با یکدیگر ادغام کند به گونه‌ای که هر سطر فقط حاوی مطالب یکی از فایل‌ها باشد. خروجی این تابع در متغیر OutFile قرار داده می‌شود.

h.veisi@ut.ac.ir

به نام خدا

دانشکده علوم و فنون نوین

آشنایی با زبان‌شناسی رایانشی (۱۲۱-۰۵-۸۳)
نیم‌سال دوم ۱۴۰۱-۱۴۰۰

تاریخ تحویل: ۱۴۰۱/۰۳/۱۳

تمرین شماره ۳

NoChar = **CountChar** (InTextFileName, CharName) تابعی که فایل متنی (InTextFileName) و یک کاراکتر (CharName) را گرفته و تعداد کل آن کارکتر (NoChar) را در آن فایل به عنوان خروجی برگرداند.
OutTextFileName = **CorrectPuncs** (InTextFileName) تابعی که فایل متنی (InTextFileName) را گرفته و پس از اصلاح اشکال‌های علائم سجاوندی در متن، فایل خروجی (OutTextFileName) را به عنوان خروجی برگرداند.
OutTextFileName = **CorrectCoding** (InTextFileName, TableCodings) تابعی که فایل متنی (InTextFileName) و فایل حاوی اصلاحات (TableCodings) را گرفته و پس از اصلاح اشکال‌های کدگذاری در متن، فایل خروجی (OutTextFileName) را به عنوان خروجی برگرداند.
OutTextFileName = **CorrectVariations** (InTextFileName, TableVariations) تابعی که فایل متنی (InTextFileName) و فایل حاوی اصلاحات (TableVariations) را گرفته و پس از اصلاح اشکال‌های تنوع املایی در متن، فایل خروجی (OutTextFileName) را به عنوان خروجی برگرداند.

الف) (۵٪) تابعی با نام SelectLexicon بنویسید که فایل ZebraAllNormalized و یک عدد به اسم ThrCount را به عنوان ورودی بگیرد و یک فایل متنی حاوی تعدادی کلمات یکتا به عنوان Lexicon برگرداند. برای این کار از تابع CountWordsFile کل کلمات یکتا و تکرار هر کدام را محاسبه کنید و کلماتی که تکرار آنها کمتر از ThrCount بود را حذف کنید و مابقی را به عنوان کلمات واژگان در نظر بگیرید. در این تمرین ThrCount را برابر با ۲ قرار دهید. در واژگان، کلمه "خاو" (خارج از واژگان)، کلمه <s> و </s> را نیز به عنوان کلمات واژگان اضافه کنید.

ب) (۱۰٪) تابعی با نام ReplaceOOV بنویسید که فایل ZebraAllNormalized، فایل Lexicon و رشته Phrase را به عنوان ورودی گرفته و همه کلماتی را که در ZebraAllNormalized آمده‌اند ولی در Lexicon نیستند با عبارت موجود در Phrase جایگزین کند. در این تمرین برای Phrase را برابر با "خاو" (خارج از واژگان) در نظر بگیرید. همچنین، در این تابع به ابتدا و انتهای هر جمله (پایان جمله \n است)، به ترتیب <s> و </s> را اضافه کنید. خروجی این تابع یک فایل حاوی کل پیکره به اسم

h.veisi@ut.ac.ir

ZebraAllNormalizedNoOOV است.

ج) (۱۵٪) تابعی با نام CalculateMonoGram بنویسید که فایل ZebraAllNormalizedNoOOV و فایل Lexicon را به عنوان ورودی گرفته و احتمال یک‌تایی (Uni-gram) کلمات واژگان را از روی تکرارهای ZebraAllNormalizedNoOOV محاسبه کند و در فایل LexiconMonogram ذخیره کند. برای محاسبه احتمال‌ها از تابع CountWordsFile استفاده کنید تا هم هم آمار تکرار هر کلمه و هم جمع کل کلمات را داشته باشید. از تقسیم آمار تکرار هر کلمه بر جمع کل کلمات، احتمال‌ها حاصل می‌شود.

د) (۱۵٪) تابعی با نام CalculateBiGram بنویسید که فایل ZebraAllNormalizedNoOOV و فایل Lexicon را به عنوان ورودی گرفته و احتمال دوتایی (Bi-gram) کلمات واژگان را با توجه به تکرارهای آنها در ZebraAllNormalizedNoOOV محاسبه کند. خروجی این تابع را فایل LexiconBigram ذخیره کند.

ه) (۲۰٪) تابعی با نام CalculateBiGramKN بنویسید که فایل ZebraAllNormalizedNoOOV و فایل Lexicon را به عنوان ورودی گرفته و احتمال دوتایی (Bi-gram) کلمات واژگان را با روش هموارسازی Kneser-Ney محاسبه کند. خروجی این تابع را فایل LexiconBigramKN ذخیره کند.

و) (۱۵٪) تابعی با نام CalculatePerplexity بنویسید که روی یک مجموعه فایل تست و یک مدل زبانی سرگشتی (Perplexity) را محاسبه کند. برای استفاده از این تابع، مقدار سرگشتی جملات زیر را برای دو مدل دوتایی (بخش د) و دوتایی هموار شده (بخش ه) محاسبه کرده و میانگین آنها را گزارش کنید.

h.veisi@ut.ac.ir

به نام خدا

دانشکده علوم و فنون نوین

آشنایی با زبان‌شناسی رایانشی (۱۲۱-۰۵-۸۳)
نیم‌سال دوم ۱۴۰۱-۱۴۰۰

تاریخ تحویل: ۱۴۰۱/۰۳/۱۳

تمرین شماره ۳

دین از جمله مواردی است که از زندگی انسان تفکیک‌پذیر نیست

بر خلاف ایران، در فرهنگ‌هایی مثل هندی به بعد معنوی تکیه شده است

آمارهای بین المللی بیانگر صعود ۸ پله‌ای ایران از نظر مبارزه با فقر و فساد است

استقبال بسیار خوبی از خریدن تولیدات فناوری ما شده است

در صورت عدم نتیجه‌گیری این تیم مقابل تیم رنجرز، به لیگ دسته یک سقوط خواهد کرد

بعد از این تمرین، توابع زیر به مجموعه برنامه شما افزوده می‌شود.

توابع تمرین شماره ۳
OutTextFileName = **SelectLexicon** (InTextFileName, ThrCount) تابعی که فایل متنی (InTextFileName) و یک عدد (ThrCount) را گرفته و کلمات یکتایی که تعداد آنها از ThrCount کمتر است را به عنوان خروجی در فایل OutTextFileName ذخیره کند.
OutTextFileName = **ReplaceOOV** (InTextFileName, Lexicon, Phrase) تابعی که فایل متنی (InTextFileName) و فایل واژگان (Lexicon) را گرفته و پس از جایگزینی کلمات خارج از واژگان با عبارت Phrase، فایل خروجی (OutTextFileName) را به عنوان خروجی برگرداند.
OutTextFileName = **CalculateMonoGram** (InTextFileName, Lexicon) تابعی که فایل متنی (InTextFileName) و فایل واژگان (Lexicon) را گرفته و پس از محاسبه احتمال یک‌تایی (Uni-gram) کلمات واژگان، فایل خروجی (OutTextFileName) را به عنوان خروجی برگرداند.
OutTextFileName = **CalculateBiGram** (InTextFileName, Lexicon) تابعی که فایل متنی (InTextFileName) و فایل واژگان (Lexicon) را گرفته و پس از محاسبه احتمال دوتایی (Bi-gram) کلمات واژگان، فایل خروجی (OutTextFileName) را به عنوان خروجی برگرداند.
OutTextFileName = **CalculateBiGramKN** (InTextFileName, Lexicon) تابعی که فایل متنی (InTextFileName) و فایل واژگان (Lexicon) را گرفته و پس از محاسبه احتمال دوتایی (Bi-gram) کلمات واژگان با هموارسازی Kneser-Ney، فایل خروجی (OutTextFileName) را به عنوان خروجی برگرداند.
PP = **CalculatePerplexity** (InTextSentence, LM) تابعی که جمله متنی (InTextSentence) و فایل مدل زبانی (LM) را گرفته و پس از محاسبه سرگشتی، مقدار آن را به عنوان خروجی برمی‌گرداند.

h.veisi@ut.ac.ir