به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

# درس پردازش زبان طبیعی

## تمرین ۱

**استاد درس:** دکتر هشام فیلی

**سرپرست دستیاران آموزشی:** سمانه پیمانی‌راد

**طراح تمرین:** محمدامین غنی زاده

**مهر ماه ۱۴۰۳**

# فهرست

## مقدمه

**موضوع تمرین: آشنایی با regular expression، Tokenization و n-gram language modeling**

**اهداف تمرین:**

- **آشنایی با مفاهیم گفته شده regular expression، edit distance، Tokenization و n-gram**
- **پیاده سازی و درک عمیق آن ها**

**نکات قابل توجه در هنگام پاسخ به سؤالات:**

برای هر سوال، دقت شود که استفاده از کتابخانه های آماده مجاز است یا خیر. در صورتی که در صورت سوال گفته نشده، میتوانید از کتابخانه های آماده استفاده کنید.

# سوال اول (۳۰ نمره)

در این سوال قصد داریم تا با مفاهیم Regular expression و levenshtein distance یا edit distance آشنا شویم.

برای بخش سوم این سوال نمی توانید از کتابخانه های آماده استفاده کنید.

## مجموعه داده

مجموعه داده مورد استفاده برای این سوال، یک فایل ساده به اسم ner.txt است که در اختیار شما قرار گرفته است و برای آشنایی با Regular expression و Tokenization از آن استفاده خواهید کرد.

## بخش اول: استفاده از REGEX برای NER (۱۰نمره)

أ در مورد Named Entity Recognition تحقیق کرده و هدف آن را توضیح دهید.

ب فایل ner.txt را باز کنید و از regular expression استفاده کنید و آدرس های ایمیل، شماره تلفن ها و Url ها را استخراج کنید و نمایش دهید.

ج مزایا و معایب استفاده از regex در تسک NER را توضیح دهید.

## بخش دوم: اشنایی اولیه با TOKENIZER (۱۰نمره)

أ توکنایزر های rule-based و مبتنی بر یادگیری ماشین را مقایسه کنید. توضیح دهید بهتر است از هر کدام در چه شرایطی استفاده شود.

ب یک توکنایزر whitespace طراحی کرده و آن را روی داده این قسمت اعمال کرده و نتایج را نمایش دهید. سپس معایب این توکنایزر را شرح دهید.

## بخش سوم: EDIT DISTANCE(۱۰نمره)

أ در مورد الگوریتم های Levenshtein distance و Damerau-Levenshtein distance تحقیق کنید و طرز کار آن ها را شرح دهید.

ب سپس آن ها را پیاده سازی کرده و فاصله جفت کلمه های زیر را در هر کدام اندازه بگیرید:

[ ("kitten", "sitting"), ("saturday", "sunday"), ("book", "back"), ("algorithm", "logarithm"), ("", "test"), ("abc", "acb") ]

## سوال دوم (۳۵ نمره)

در این سوال قصد داریم تا با n-gram language models آشنا شده و به تولید متن با استفاده از آن بپردازیم. همچنین با perplexity آشنا خواهیم شد و از یک کاربرد آن استفاده خواهیم کرد.

برای بخش سوم این سوال و همچنین محاسبه perplexity نمی توانید از کتابخانه آماده استفاده کنید.

### مجموعه داده

سه فایل برای این سوال در اختیار شما قرار داده خواهد شد. فایل Ferdowsi.txt شامل اشعار فردوسی، hafez.txt شامل اشعار حافظ و modern_poet.txt شامل مجموعه ای از اشعار نو است.

### بخش اول: آشنایی با توکنایزر WORDPIECE (۸ نمره)

درباره Wordpiece Tokenizer تحقیق کنید و به سوالات زیر پاسخ دهید:

أ نحوه آموزش این توکنایزر را به طور کامل شرح دهید.

ب نمونه ای از مدل های معروفی که با این توکنایزر آموزش دیده را نام ببرید.

ج آن را با BPE مقایسه کنید.

### بخش دوم: آموزش WORDPIECE (۸ نمره)

أ یک wordpiece توکنایزر ساخته و روی Ferdowsi.txt آموزش دهید. (می توانید از کتابخانه های آماده برای آموزش این توکنایزر استفاده کنید).

ب پس از آموزش، یک متن را به انتخاب خودتان با استفاده از این توکنایزر، توکنایز کنید و نتایج را نشان دهید.

### بخش سوم: آموزش N-GRAM (۸ نمره)

أ یک کلاس برای ساختن و آموزش n-gram بسازید و با استفاده از این کلاس و توکنایزری که در بخش قبلی آموزش دادید، ۲-gram، ۴-gram و ۸-gram بسازید و روی داده Ferdowsi آموزش دهید.

ب با استفاده از مدل های آموزش داده شده، متن ۲۰۰ توکنی تولید کنید و متن های تولید شده توسط این مدل ها را مقایسه کنید.

ج n های بزرگ در n-gram ها، با چه مشکلاتی روبه رو می شوند؟ توضیح دهید.

## بخش چهارم: معیار PERPLEXITY (۱۱ نمره)

أ معیار perplexity را توضیح دهید و بگویید برای زمانی که perplexity یک متن، روی یک language model کم است، نشان دهنده چه چیزی است؟

ب با استفاده از یکی از مدل هایی که در بخش قبلی آموزش دادید، (۴-gram یا ۸-gram) این کار را انجام دهید: با استفاده از مدل آموزش داده شده روی Ferdowsi، perplexity هر یک از سطر های فایل hafez را اندازه بگیرید، سپس این مقدار را تقسیم بر تعداد کل سطر ها بکنید. به زبان ساده تر، میانگین perplexity مدل آموزش دیده را روی سطر های فایل hafez به دست اورید. سپس همین کار را با فایل اشعار نو انجام دهید. دو عدد به دست آمده را با هم مقایسه کرده و تحلیل خود را از این نتیجه بگویید..

# سوال سوم (۳۵ نمره)

در این سوال قصد داریم از n-gram ها به عنوان یک طبقه بند استفاده کنیم. به همین منظور می خواهیم با استفاده از n-gram ها، نظر مردم در سایت دیجیکالا را بررسی کنیم.

برای بخش سوم این سوال نمی توانید از کتابخانه آماده استفاده کنید.

## مجموعه داده

مجموعه داده برای این سوال در فایل digikala.csv قابل دسترسی هستند. همچنین می توانید داده را از اینجا دانلود کنید. دو ستون Text و Suggestion از این دیتاست مورد استفاده خواهند بود. می توانید توضیحات مربوط به هر کلاس از Suggestion را در لینک بالا مطالعه کنید.

## بخش اول: N-GRAM به عنوان طبقه بند(۸ نمره)

أ توضیح دهید چگونه می توان از n-gram به عنوان طبقه بند استفاده کرد.

ب در صورت کم بودن داده، افزایش داده n در n-gram باعث چه مشکلاتی می شود؟

## بخش دوم: TEXT TO TOKEN(۸ نمره)

أ یک توکنایزر BPE روی داده آموزش دهید.

ب با استفاده از این توکنایزر، دیتاست را توکنایز کنید.

## بخش سوم: آموزش طبقه بند(۱۹ نمره)

أ مانند سوال قبلی، یک کلاس برای n-gram بسازید با این تفاوت که این بار وظیفه طبقه بندی را بر عهده خواهد داشت.

ب دیتاست را به دو بخش آموزش و ارزیابی تقسیم کنید. با استفاده از دیتاست آموزش، یک ۲-gram و یک ۳-gram آموزش دهید.

ج معیار های accuracy، precision و recall را برای هر دو مدل آموزش داده شده گزارش کنید.

د تحلیل خود را از نتایج به دست آمده بیان کنید.

ه بیان کنید در چه صورتی استفاده از طبقه بند n-gram به جای طبقه بند های با پارامتر های زیاد می تواند مفید باشد.

## سوال چهارم (امتیازی)

در این سوال قصد داریم تا با ایده توکنایز کردن تصاویر آشنا شویم. سپس با استفاده از تصاویر توکنایز شده، یک طبقه بند را آموزش دهیم.

در این سوال برای آموزش KMeans می توانید از کتابخانه آماده استفاده کنید ولی برای آموزش n-gram نمی توانید استفاده کنید.

### مجموعه داده

مجموعه داده این سوال، دیتاست mnist است که به سادگی می توانید از اینترنت دانلود کنید.

### بخش اول: توکنایز کردن تصاویر

أ اکنون میخواهیم به جای متن، عکس را توکنایز کنیم. برای این کار، نیاز داریم تا عکس ها را patch کنیم. تحقیق کنید و توضیح دهید این کار در Vision Transformer[1] چگونه انجام می شود.

ب دیتاست mnist را لود کرده، هر نوع پیش پردازشی که لازم است انجام دهید. سپس همه ی پچ های عکس ها را از دیتاست استخراج کرده و یک مدل خوشه بندی KMeans روی این داده آموزش دهید. اکنون این مدل خوشه بندی، مانند یک توکنایزر عمل میکند و باید ورودی گرفتن یک patch، آن را به یک خوشه خاص encode می کند که شبیه به توکنایز کردن است. پس از آموزش این خوشه بند، میتوانید دیتاست را به این صورت توکنایز کنید: ابتدا هر عکس را به patch ها تقسیم کنید، سپس با ورودی دادن این patch ها به مدل KMeans آموزش داده شده، خوشه را به عنوان توکن این patch در نظر بگیرید.

به عنوان مثال اندازه هر patch را ۷x۷ در نظر بگیرید. از آن جایی که عکس های mnist، دارای ابعاد ۲۸x۲۸ هستند، ۱۶ تا patch از این عکس استخراج خواهد شد.(با در نظر گرفتن stride=۷) هر patch به مدل

[1] https://arxiv.org/abs/۲۰۱۰.۱۱۹۲۹

Kmean آموزش داده شده داده می شود، فرض کنید مدل آموزش داده شده دارای تعداد خوشه ۲۵۶ است. در نتیجه هر عکس پس از توکنایز شدن، تبدیل به یک دنباله ۱۶ تایی از اعداد بین ۰ تا ۲۵۵ می شود. (در انتخاب hyperparameter ها مانند اندازه patch آزاد هستید)

برای مثال این می تواند یک نمونه توکنایز شده از یک تصویر باشد:

[۱۷۰, ۲٤, ۱٤۷, ۸۳, ۱۲, ۱۵۰, ۱۱۰, ۶۴, ۳۲, ۷۰, ۱۲۸, ۳٤, ۵٤, ٤۵, ۲۱۵, ۱۷۰]

## بخش دوم: آموزش طبقه بند

أ حال که دیتاست تصویر را توکنایز کردیم، یک gram-۴ روی آن آموزش دهید تا با قبول کردن یک عکس، بتواند کلاس آن را با استفاده از این gram-۴ پیشبینی کند.

ب در نهایت دقت مدل را اندازه بگیرید و چند نمونه از پیشبینی های مدل همراه با لیبل اصلی را نمایش دهید. ( اندازه دقت مورد نظر نیست اما طبقه بند باید حداقل بهتر از رندوم عمل کند)

(پارامتر های پیشنهادی برای این سوال:patch_size=۹،stride=۹،تعداد کلاستر برای KMeans=۲۵۶)

---

## ملاحظات (حتما مطالعه شود)

تمامی نتایج شما باید در یک فایل فشرده با عنوان NLP-CA۱-StudentID تحویل داده شود.

- خوانایی و دقت بررسی‌ها در گزارش نهایی از اهمیت ویژه‌ای برخوردار است. به تمرین‌هایی که به صورت کاغذی تحویل داده شوند یا به صورت عکس در سایت بارگذاری شوند، ترتیب اثری داده نخواهد شد. دقت کنید که حتما گزارشات خود را در قالب ارائه شده برای تحویل تکالیف که در سامانه برای شما بارگذاری شده است ارسال بفرمایید.
- کدهای نوشته شده برای هر بخش را با نام مناسب مشخص کرده و به همراه گزارش تکلیف ارسال کنید. همه‌ی کدهای پیوست گزارش بایستی قابلیت اجرای مجدد داشته باشند. در صورتی که برای اجرا مجدد آن‌ها نیاز به تنظیمات خاصی می‌باشد بایستی تنظیمات مورد نیاز را نیز در گزارش خود ذکر کنید. **دقت کنید که تمامی کدها باید توسط شما اجرا شده باشند و نتایج اجرا در فایل کدهای ارسالی مشخص باشد. به کدهایی که نتایج اجرای آن‌ها در فایل ارسالی مشخص نباشد نمره‌ای تعلق نمی‌گیرد.**
- تمرین تا یک هفته بعد از مهلت تعیین شده با تاخیر تحویل گرفته می شود. دقت کنید که شما جمعا برای تمام تکالیف، ۱۴ روز زمان تحویل بدون جریمه دارید که تنها از ۷ روز آن برای هر تمرین می‌توانید استفاده کنید، در صورتی که این ۱۴ روز به اتمام رسیده باشد، به ازای هر روز تأخیر در ارسال تمرین، ده درصد جریمه می‌شوید.
- **توجه کنید این تمرین باید به صورت تک نفره انجام شود و پاسخ‌های ارئه شده باید نتیجه فعالیت فرد نویسنده باشد (همفکری و به اتفاق هم نوشتن تمرین نیز ممنوع است). در صورت مشاهده تشابه به همه افراد مشارکت کننده، نمره صفر تعلق می‌گیرد و به استاد نیز گزارش می‌گردد.**
- در صورت بروز هرگونه مشکل با ایمیل زیر در ارتباط باشید:

ghanizadeh.amin@ut.ac.ir

| | |
|---|---|
| تاریخ آپلود تمرین | ۱۴۰۳/۷/۱۴ |
| مهلت تحویل بدون جریمه | ۱۴۰۳/۷/۲۷ |
| مهلت تحویل با تأخیر، با جریمه ۱۰ درصد | ۱۴۰۳/۸/۴ |