به نام خدا

تمرین اول درس پردازش زبان طبیعی

«آشنایی با ابهام زدایی از مفهوم کلمات و نحوه‌ی ساخت تجزیه‌کننده وابستگی»

استاد درس: دکتر ممتازی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

بهار ۱۴۰۱

برای ارسال تمرین به نکات زیر توجه کنید.

۱- در جدول زیر نحوه اعمال نمره منفی برای تاخیر در ارسال تمرین‌ها ذکر شده‌است.

میزان تاخیر (روز)	میزان جریمه
۱ الی ۲ روز	هر روز ۵٪
۲ الی ۶ روز	هر روز ۱۰٪

در صورتی که برای ارسال تمرین‌ها بین ۷ تا ۱۴ روز تاخیر داشته باشید، نمره شما از ۵۰٪ محاسبه می‌شود و پس از این بازه به تمرین ارسالی نمره‌ای تعلق نمی‌گیرد.

۲- هرگونه کپی‌برداری در انجام تمرین‌ها موجب کسر نمره خواهد شد.

۳- آخرین مهلت ارسال تمرین، **ساعت ۲۳:۵۵ روز سه‌شنبه ۱۷ خرداد** می‌باشد.

۴- فایل های ارسالی خود شامل فایل‌های پیاده‌سازی و گزارش را فشرده کنید و با عنوان «**شماره دانشجویی_HW3**» مانند `HW3_97131022` ارسال کنید.

۵- زبان برنامه‌نویسی برای انجام تمرین‌ها، پایتون یا جاوا در نظر گرفته شده‌است.

۶- کدهای ارسالی خود را برای افزایش خوانایی و درک بهتر به صورت مناسب کامنت‌گذاری کنید.

۷- نحوه انجام پیش‌پردازش بر روی داده‌ها شامل کتابخانه مورد استفاده و مراحل انجام‌شده را در گزارش خود مکتوب کنید.

۸- برای انجام این تمرین می‌توانید از کتابخانه‌های آماده استفاده کنید.

۹- در صورت هرگونه سوال یا مشکل می‌توانید با تدریس‌یاران درس از طریق ایمیل زیر در ارتباط باشید.

محمدجواد ظهرابی – رضا زادکمالی nlp.aut.1401@gmail.com

بخش اول: ابهام‌زدایی معنایی کلمات

در این بخش قصد داریم با کار ابهام‌زدایی معنایی کلمات آشنا شویم. مجموعه دادگان و کلیه فایل‌های مورد نیاز برای این تمرین را می‌توانید از طریق لینک زیر دانلود کنید.

```
!gdown 1oin_Sw1Gk_WLS9zpDrap5FdlRcUrTP_D
!gdown 1EvtGQ8-sYXXQ3VA9ByjD4OmFH13WxvAs
!gdown 154f-z0PsPAp0yvOLdNXP8NXYjPljpgCZ
!gdown 1EDG_j6F5ohIjpkihRQd0-i9daHWV7StN
```

داده‌های قرار داده شده برای این تمرین شامل چهار فایل است. هر سطر از فایل‌های sentences_train.txt و senses_train.txt به ترتیب نشان‌دهنده‌ی جملات آموزشی دارای کلمه‌ی مبهم و شناسه‌ی مفهوم مرتبط با آن است. به همین شکل، فایل‌های sentences_test.txt و senses_test.txt نشان‌دهنده‌ی داده‌های آزمون هستند. در جدول زیر کلمات به همراه تعداد مفهوم‌های آن‌ها آمده است.

کلمه	تعداد شناسه‌های مفهوم
Hard	3
Interest	6
Line	6
Serve	2

گام اول: استفاده از بازنمایی BERT

در این قسمت با استفاده از مجموعه‌داده‌های معرفی شده، برای کلمات دارای ابهام معنایی در مجموعه داده، بازنمایی BERT را به دست آورید. برای استخراج بازنمایی‌هایBERT می‌توانید از کتابخانه bert-embeddings استفاده کنید (استفاده از سایر کتابخانه‌ها نیز مانعی ندارد). با توجه به اینکه بردار خروجی مدلBERT برای هر کلمه ۷۶۸ بعد است و برای این که در هنگام آموزش در قسمت بعد با کمبود منابع رو به رو نشوید، بردارهای خروجی برای کلمات مبهم را با استفاده از الگوریتم PCA به ۳۰۰ بعد کاهش دهید. با استفاده از مدل از پیش آموزش داده شده BERT و استخراج بازنمایی تنها برای کلمه مبهم موجود در متن از مدل BERT بدون استخراج بازنمایی کلمات دیگر متن، بازنمایی هریک از ورودی‌های مبهم را به دست آورید. دقت داشته باشید که مدل BERT مبتنی بر بافت است یعنی کلمات اطراف کلمه هدف تاثیر خود را در بازنمایی کلمه هدف به جا می‌گذارند. سپس با استفاده از بازنمایی و توضیحات بخش بعد مفهوم مرتبط با آن ورودی را به دست آورید.

گام دوم: دسته‌بندی

در این بخش قصد داریم با استفاده از بازنمایی ایجاد شده در گام قبل، با استفاده از الگوریتم SVM بهترین مفهوم را برای کلمه‌ی مبهم پیدا کنیم. دقت داشته باشید که تنظیم پارامترهای مدل به عهده‌ی شماست.

- معیارهای Accuracy و F-measure را روی داده‌ی آزمون گزارش کنید.

بخش دوم: ایجاد تجزیه‌کننده روابط وابستگی

در این قسمت می‌خواهیم یک تجزیه‌گر روابط وابستگی را ایجاد کنیم. مجموعه دادگان و کلیه فایل‌های مورد نیاز برای این تمرین را می‌توانید از طریق لینک زیر دانلود کنید.

```
!gdown 11WeeMttH6I6MJ0t1h7FVSEtw0lKwpRA6
!gdown 1gLGNxjQzy6C8y4Oivr8etU1MMGfuKuEE
!gdown 127-sOeW6KMf6XNSAVM3bGjfwnmW0NciU
```

داده‌های قرار داده شده برای این تمرین شامل سه فایل train.conll، dev.conll و test.conll است. ستون اول تا سوم هر فایل به ترتیب نشان‌دهنده‌ی این موارد هستند: ۱) شماره مربوط به جایگاه کلمه در جمله، ۲) خود کلمه و ۳) شماره مربوط به جایگاه کلمه‌ای که به آن وابسته است.

گام اول: پیش‌پردازش

در مرحله‌ی پیش‌پردازش لازم است که از طریق دو ستون اول و سوم، یک ستون جدید ایجاد کنید که نشان دهنده‌ی این است که هر کلمه چه تعداد جابجایی به سمت چپ یا راست برای رسیدن به کلمه‌ای که به آن وابسته است، نیاز دارد. دقت داشته باشید که برای ریشه مقدار ستون سوم برابر با صفر است. در جدول زیر مثالی از نحوه‌ی پیش‌پردازش روی یک سطر آمده است، که در آن R و L به ترتیب مخفف راست و چپ هستند.

جدول ۱: مثال از نحوه‌ی انجام پیش‌پردازش روی مجموعه‌ی داده

ستون‌های مجموعه‌داده‌ی اولیه			ستون‌های جدید	
1	We	2	We	1R
2	had	0	had	Root
3	to	4	to	1R
4	think	2	think	2L
5	about	6	about	1R
6	it	4	it	2L

گام دوم: ایجاد شبکه‌ی عصبی حافظه کوتاه-مدت بلند دوطرفه[1]

الف) مشابه با تمرین سری دوم، در این گام قصد داریم که از شبکه‌ی LSTM دو طرفه (BiLSTM) به عنوان دسته‌بندی کننده استفاده کنیم. برای پیاده‌سازی این شبکه می‌توانید از کتابخانه‌های Tensorflow (Keras) و یا PyTorch استفاده کنید.

- برای ورودی شبکه، از بردارهای تعبیه پیش آموزش دیده‌ی word2vec با ۱۰۰ بعد استفاده کنید. برای این کار می‌توانید از کتابخانه genism استفاده کنید.

ب) بخش امتیازی: با تغییر ساختار شبکه (استفاده از شبکه‌های عصبی دیگر نیز مجاز است) و یا جایگزینی تعبیه‌های پیش‌آموزش‌دیده‌شده با حالت‌های دیگری مانند: مقداردهی اولیه تصادفی، glove، BERT و موارد مشابه دیگر سعی کنید به مدل بهتری برسید. مدل پیشنهادی و نتایج آن را با مدل بخش الف مقایسه کنید.

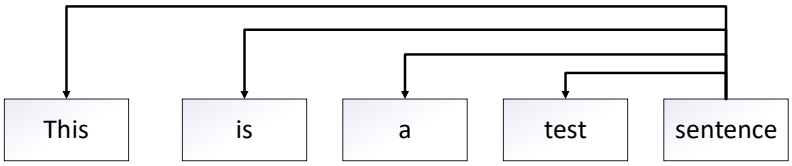
گام سوم: تحلیل و ارزیابی

الف) برای ارزیابی مدل گام قبل، معیارهای Accuracy، Precision، Recall و F-measure را برای الگوریتم‌های انتخاب شده روی داده‌ی آزمون گزارش کنید.

ب) نمودار خطای مدل انتخاب شده را روی مجموعه‌داده‌های train و validation در هر ایپاک[2] رسم کنید.

ج) برای سه جمله‌ی زیر نتایج مدل را به دست آورده و آن‌ها را به شکل زیر نشان دهید (دقت کنید که خروجی مدل خود را به همراه حالت نموداری آن را که در شکل ۱ آمده است، گزارش کنید).

There are no mistakes, only opportunities.
Simplicity is the ultimate sophistication.
Whatever you do, do it well.



شکل ۱: حالت نموداری جمله‌ی "This is a test sentence" که پیش‌بینی مدل برای آن "4R, 3R, 2R, 1R, Root" بوده است.

موفق باشید

[1] BiLSTM
[2] Epoch