به نام خدا

تمرین دوم درس پردازش زبان طبیعی

عنوان: دسته‌بندی متن

استاد درس:

دکتر اکبری

موعد تحویل: ۱۳۹۹/۰۸/۱۲

فهرست مطالب

شرح تمرین

هدف از این تمرین طراحی مدلی است که بتواند خبرهای موجود در یک مجموعه داده خبری را به ۱۰ مقوله‌ی جداگانه دسته‌بندی کند. مانند تمرین قبل از مجموعه داده‌ی اخبار باشگاه خبرنگاران جوان و فارس‌نیوز که در سال ۱۳۹۷ تهیه و جمع‌آوری شده است بهره می‌گیریم. در ادامه مجموعه داده، نحوه طراحی مدل و همچنین ورودی و خروجی مدل را تشریح می‌کنیم.

بخش ۱: شناخت داده‌ها

ابتدا فایل `News.rar` را از طریق این لینک دانلود کنید. این فایل متشکل از دو فایل `train.csv` و `test.csv` می‌باشد. این دو فایل حاوی شش ستون می‌باشد که سه ستون `text`، `title` و `category` که به ترتیب نمایش‌دهنده‌ی متن خبر، عنوان خبر و دسته خبر می‌باشند، دارای اهمیت هستند. در واقع `text` و `title` به عنوان ورودی و `category` به عنوان خروجی مدلی که قرار است طراحی شود در نظر گرفته شده است. فایل `train.csv` حاوی ۱۱۷۱۹۰ نمونه آموزشی و فایل `test.csv` حاوی ۲۱۱۰۴ نمونه آزمایشی می‌باشند. در ضمن، `encoding` این دو فایل `utf-8` می‌باشد.

توجه کنید که داده آموزشی `train.csv` را با یک نسبت معقول مانند ۷۰ به ۳۰ به دو بخش آموزشی و اعتبارسنجی تفکیک کنید. از داده اعتبارسنجی برای یافتن پارامترهای غیر قابل تنظیم[1] و همچنین یافتن بهترین مدل استفاده کنید.

بخش ۲: نوع دسته‌بند مورد استفاده

برای این تمرین از ۵ دسته‌بند زیر می‌توانید استفاده کنید:

الف- `Naïve Bayes`

ب- `Support Vector Machine (SVM)`

پ- `Hidden Markov Model (HMM)`

ت- `Conditional Random Field (CRF)`

ث- `Long Short Term Memory (LSTM)`

دسته‌بندهای (ب)، (پ) و (ث) امتیازی هستند.

توجه شود که در این تمرین `HMM` و `CRF` به صورت مرتبه اول مورد قبول است. همچنین در صورت استفاده از دسته‌ند (ث) تنها از یک لایه `LSTM` و حداکثر سه لایه‌ی تماما متصل[2] استفاده شود.

بخش ۳: پیش‌پردازش داده

برای آموزش دسته‌بند و اعمال داده به آن، ابتدا باید داده‌ی مورد نظر پیش‌پردازش شود. با دنبال کردن مراحل زیر می‌توانید داده را پیش‌پردازش کنید.

[1] `Non-tunable`
[2] `Fully connected`

۱. متن اخبار را پاکسازی کنید. بدین منظور، تمامی کلمات انگلیسی را حذف نمایید. همچنین کاراکترهای خاص (مانند *) و علائم نگارشی (به جز نقطه، علامت سوال) را نیز حذف کنید. در پایان این مرحله، متن هر خبر باید فقط حاوی حروف فارسی، اعداد و دو علامت نگارشی خاص (نقطه و علامت سوال) باشد.

۲. در متن کلیه اخبار، به جای اعداد، کاراکتر N را قرار دهید.

۳. متن اخبار را به **کاراکترها یا کلمات** تجزیه کرده و تعداد کلمات و تعداد کاراکترها را گزارش کنید. هدف ما نمایش سطح-کلمه[۳] و سطح-کاراکتر[۴] از یک جمله می‌باشد. برای این تمرین می‌توانید از نمایش سطح کلمه، سطح کاراکتر و یا هر دو به صورت همزمان استفاده کنید.

۴. پس از محاسبه ۱۰۰۰۰ کلمه پرتکرار، میزان پوشش کل توکن‌ها توسط این کلمات را بدست آورید؛ یعنی محاسبه کنید که این کلمات چند درصد توکن‌ها را تشکیل می‌دهند.

۵. در کلیه جملات، به جای هر کلمه که جزء ۱۰۰۰۰ کلمه پرتکرار نیست، نماد UNK را قرار دهید. (توجه کنید که اعداد تا پایان این مرحله به شکل N نمایش داده شده و حذف نمی‌شوند).

۶. این بخش توکن کردن[۵] متن نام دارد. برای این کار ابتدا کلمات و کاراکترها را در یک لیست براساس حروف الفا مرتب کنید. سپس دو دیکشنری word2index و index2word و همچنین دو دیکشنری char2index و index2char را به گونه‌ای بسازید که هر کلمه و کاراکتر به یک عدد منحصر به فرد نگاشت شود و بالعکس. اکنون نمایش سطح کاراکتر و سطح کلمه را با استفاده از indexها نمایش دهید.

۷. در داده آموزشی میانگین طول اخبار را به دسته آورید و اخباری که دارای طولی بیش از طول میانگین می‌باشند را کنار بگذارید. توجه کنید که میانگین‌گیری را بر اساس تعداد کلمات انجام دهید. سپس بر اساس نوع نمایش (سطح کلمه و یا سطح کاراکتر)، طول جملات را با افزودن یک توکن به نام PAD به مجموعه واژگان و مجموعه کاراکترها یکسان کنید.

۸. این بخش بردار کردن[۶] متن نام دارد. ساده‌ترین روش نمایش متن استفاده از روش نمایش One-hot می‌باشد. در این روش برای نمایش هر توکن بدین صورت عمل می‌شود که یک بردار ۰ و ۱ به طول کل توکن‌ها ساخته می‌شود که تمامی درایه‌های آن به جز داریه‌ای که index آن با index آن توکن برابر است، صفر است. برای نمایش یک متن می‌توانید تمامی توکن‌ها (کاراکترها یا کلمات) را به صورت یک بردار درآورید و سپس با کنار هم قرار دادن این بردارها یک ماتریس بسازید که نمایش دهنده کل متن یا جمله می‌باشد. با توجه به اتلاف شدید حافظه در این روش می‌توانید از روش‌های دیگری نظیر Bag of Words یا ماتریس TF-IDF استفاده کنید. اکنون از شیوه‌های نمایش بیان شده یکی را به دلخواه استفاده کنید.

۹. در صورتی که حافظه شما محدود است و نمی‌توانید کلیه اخبار را در حافظه قرار دهید، داده‌ها را حین فرایند آموزش و به صورت دسته‌ای بردار کنید. برای این کار می‌توانید از data generatorها استفاده کنید.

---

[3] Word-level
[4] Character level
[5] Tokenizing
[6] Vectorizing

۱۰. برای مدل‌های Naïve bayes و HMM نیازی به انجام مراحل ۸ و ۹ نیست.

بخش ۴: طراحی دسته‌بند

در این بخش به تشریح توابع‌ای که باید پیاده‌سازی شود پرداخته می‌شود. با توجه به داده‌های بیان شده در بخش گذشته، توابع زیر را پیاده‌سازی کنید.

- **تابع Clean**

در این تابع فرایند پاکسازی متون خبر انجام می‌شود. بدین ترتیب که کلیه متون اخبار به صورت یک لیست به این تابع داده می‌شود و یک لیست از اخبار پاکسازی شده از آن خروجی گرفته می‌شود.

- **تابع Tokenize**

ورودی‌های این تابع چهار دیکشنری word2index، index2word، char2index و index2char به همراه لیستی از داده‌های پاکسازی شده و یک متغیر عدد صحیح به نام level می‌باشد. اگر level برابر ۰ باشد خروجی تابع یک لیست از اخبار با نمایش سطح-کلمه می‌باشد. اگر این متغیر برابر ۱ باشد خروجی به صورت یک لیست از اخبار با نمایش سطح-کاراکتر و اگر ۲ باشد خروجی به صورت دو لیست از اخبار با نمایش سطح-کلمه و سطح-کاراکتر می‌باشد.

- **تابع Vectorize**

ورودی این تابع خروجی تابع Tokenize به همراه یک متغیر عدد صحیح به نام level می‌باشد. که این متغیر مانند متغیر level در تابع Tokenize عمل می‌کند. خروجی تابع یک آرایه از اخباری است که به صورت بردار درآمده‌اند.

امتیازی ۱: برای نمایش سطح-کلمه از Embeddingهای Fasttext، Word2vec، Glove، Elmo و یا embedding مدل زبانی BERT استفاده کنید.

- **تابع Defining_model**

در این تابع مدل مورد استفاده را تعریف کرده و پارامترهای آن را به عنوان ورودی تابع معین کنید.

امتیازی ۲: مدل را با استفاده از شبکه‌ی عصبی LSTM، HMM و یا CRF پیاده‌سازی کنید. برای تعریف مدل می‌توانید از کتابخانه‌های tensorflow، keras و یا pytorch استفاده کنید.

- **تابع Train**

در این تابع با دریافت ورودی و خروجی (X_train و y_train) مدل مورد استفاده را آموزش می‌دهد.

امتیازی ۳: برای آموزش مدل از نمایش سطح-کاراکتر و سطح-کلمه به صورت همزمان استفاده کنید. ورودی مدل را به صورت (X_word_train و X_char_train) در نظر بگیرید.

متناسب با نیاز خود و سبک برنامه‌نویسی که دارید، می‌توانید توابع دیگری را نیز اضافه کنید. اما به خاطر داشته باشید توابع فوق حتما باید موجود باشند و به درستی کار کنند. پاسخ شما به تمرین بر اساس این توابع و هر گونه عملی که از طریق آن‌ها انجام شود، ارزیابی می‌گردد.

بخش ۵: ارزیابی مدل روی داده‌های آموزش، اعتبارسنجی و آزمایش

برای ارزیابی مدل از معیارها و آنالیزهای زیر استفاده کنید.

- معیار دقت
- معیار `Precision`
- معیار `Recall`
- معیار `F1-score`
- آنالیز ماتریس `Confusion`
- آنالیز نمودار `Receiver Operating Characteristic`

در این بخش، باید **یک تابع با نام `evaluate`** را پیاده سازی کنید. که تمامی موارد فوق را انجام دهد. این تابع یک جمله را به عنوان ورودی دریافت می‌کند و پس از پیش‌پردازش، داده موردنظر را به مدل اعمال کرده و خروجی را ارزیابی می‌کند.

انتظار می‌رود با روش‌هایی نظیر `cross validation`، `grid search` و ... بهترین مدل انتخاب شود. توجه کنید که لازم است خروجی به دست آمده از بخش ارزیابی را تحلیل کنید.

شرح مستندسازی

مستندسازی یک تکه کد، به دیگر توسعه‌دهندگان در فهم آن کمک می‌کند. در این تمرین از شما تقاضا داریم یک فایل کوتاه در قالب `pdf` در شرح کدهای خود بنویسید. یک تا دو صفحه کافی است. لطفا مختصر توضیح دهید. برای هر تابعی که نوشته‌اید، به طور مختصر نحوه کارکرد آن را گزارش دهید. همچنین ورودی و خروجی (در صورتی که عینا مطابق تمرین نیست و یا پارامتر اضافه‌ای دارد) را ذکر نمایید. **در مستندسازی حتما نام و نام خانوادگی خود را به همراه شماره دانشجویی‌تان ذکر نمایید.**

تقسیم‌بندی نمرات برای ارزیابی

خواسته تمرین	نمره	نوع
تابع clean	۱۰	اصلی
تابع Tokenize	۲۵	اصلی
تابع Vectorize	۲۰	اصلی
تابع Vectorize (Embedding)	۵	امتیازی
تابع Defining_model	۱۰	اصلی
تابع Defining_model (LSTM، HMM، CRF)	۵	امتیازی
تابع Train	۱۰	اصلی
تابع Train (سطح- کلمه و سطح کاراکتر)	۱۰	امتیازی
ارزیابی مدل و تحلیل خروجی	۲۰	اصلی
مستندسازی	۵	اصلی
مجموع اصلی	۱۰۰	
مجموع امتیازی	۲۰	

نحوه ارسال پاسخ

پاسخ شما به این تمرین باید در قالب یک فایل فشرده (zip) باشد که در سامانه courses بارگذاری می‌گردد. این فایل شامل موارد زیر است:

- فایل‌هایی با پسوند py. و یا ipynb. که شامل کد مربوط به پیاده‌سازی توابع هستند. لازم است به وضوح مشخص شود که هر بخش از کد شما مربوط به پاسخ کدام بخش از تمرین است. برای این کار، لطفا به یکی از روش‌های زیر عمل نمایید:

  - درج comment در فایل py.
  - درج کدهای markdown در notebook
  - استفاده از فایل‌های جدا برای هر بخش از تمرین

توجه: کد برنامه شما باید به زبان پایتون ۳ نوشته شود.

- یک فایل متنی با نام words.txt که حاوی تمامی کلمات می‌باشد.

- یک فایل متنی با نام chars.txt که حاوی تمامی کاراکترها می‌باشد.

چهار فایل pickle. که که دیکشنری‌های بیان شده در این تمرین در آن‌ها ذخیره شده است.

- یک فایل متنی با نام most_frequent.txt که حاوی ۱۰۰۰۰ کلمه با بیشترین میزان رخداد است.

- یک فایل با نام `docs.pdf` که در آن مستندسازی توابع قرار دارد.

- هر گونه فایل دیگری که برای بارگذاری مدل شما موردنیاز است. (مجموعه داده را دوباره بارگذاری نکنید)

لطفا در صورت وجود هر گونه سوال از طریق ایمیل زیر آن را مطرح بفرمایید:

`sadeghi.hamidreza1400@gmail.com`

توجه: مهلت ارسال تمرین تا ساعت ۲۴ روز دوشنبه ۱۳۹۹/۰۸/۱۲ می‌باشد و پاسخ به تمرین پس از این زمان پذیرفته نیست.

با آرزوی موفقیت

حمیدرضا صادقی