به نام خدا

عنوان: تشخیص زبان متن

تهیه کنندگان: زهرا مختاری و هدیه نوحی نژاد

استاد: جناب آقای مهندس مجتبی‌فر

لینک github:

https://github.com/Eddie-NN/LanguageD.git

- **مقدمه**

این پروژه یک نرم‌افزار ساده و کاربردی برای تشخیص زبان متن ورودی کاربر است که با استفاده از کتابخانه‌ی قدرتمند fastText و یک مدل آموزش‌دیده از پیش، زبان متن را شناسایی کرده و نام زبان را به کاربر نمایش می‌دهد. رابط گرافیکی پروژه با استفاده از Tkinter پیاده‌سازی شده و استفاده از آن برای کاربران نهایی بسیار ساده و سریع است.

- **پیش‌نیازهای سخت‌افزاری و نرم‌افزاری**

پیش‌نیازهای نرم‌افزاری:

۱. Python 3.8 یا بالاتر

نسخه استفاده شده پایتون در این پروژه Python 3.11.9 است.

نسخه پیشنهادی را از سایت python.org دریافت و نصب کنید.

۲. نرم افزاری برای اجرای کدهای پایتون نیاز داریم. برنامه استفاده شده و محیط توسعه این پروژه، نرم افزار pycharm است که میتوان به عنوان یک محیط توسعه پایتون قوی و کامل از آن استفاده کرد.

برای دریافت این نرم افزار، از سایت jetbrains.com، این نرم افزار را دانلود و نصب کنید.

۳. PIP ابزار مدیریت بسته‌های پایتون

برای نصب کردن کتابخانه‌های این پروژه، نیاز به فعال کردن pip دارید. وارد محیط terminal پای‌چارم شده و pip –version را وارد کنید.

ورژن pip نصب شده برای اجرای این پروژه، 25.1.1 میباشد.

- **ایجاد محیط مجازی**

۱. نرم‌افزار PyCharm را اجرا کنید.

۲. در صفحه‌ی شروع، روی گزینه‌ی New Project کلیک کنید.

۳. در قسمت سمت چپ، گزینه‌ی Pure Python را انتخاب کنید.

۴. در قسمت Location، مسیر دلخواه برای ذخیره‌ی پروژه را مشخص کنید (مثلاً روی دسکتاپ، در پوشه‌ای به نام LanguageDetector.)

۵. در قسمت " "Python Interpreter این گزینه را فعال کنید:
New environment using: Virtualenv

۶. مطمئن شوید که Location همان مسیر env داخل پوشه‌ی پروژه باشد. (به‌صورت خودکار تنظیم می‌شود)

۷. Base interpreter روی ورژن 3.11.9 از پایتون که نصب کردید قرار دهید.

۸. اگر چیزی نمایش داده نشد، روی دکمه‌ی ... کلیک کنید و مسیر فایل python.exe را به‌صورت دستی بدهید.

۹. روی دکمه‌ی Create کلیک کنید.

بعد از چند ثانیه، پروژه با محیط مجازی اختصاصی خودش باز خواهد شد.

برای ایجاد فایل پروژه:

۱. در پنل سمت چپ PyCharm (Project Explorer)، روی نام پروژه راست‌کلیک کنید.

۲. گزینه‌ی New > Python File را انتخاب کنید.

۳. نام فایل را مثلاً detection.py بگذارید.

۴. سپس کدهای مورد نظر را وارد کنید.

- نصب کتابخانه‌های موردنیاز با pip

پس از ساخت محیط مجازی و باز کردن پروژه در PyCharm، حالا باید کتابخانه‌هایی که برنامه به آن‌ها وابسته است را نصب کنیم. این کار به‌سادگی و از طریق ترمینال داخلی PyCharm قابل انجام است.

Fasttext: برای تشخیص زبان متن با استفاده از مدل آموزش‌دیده‌ی lid.176.bin که توسط Facebook AI ساخته شده است.

Pycountry: برای تبدیل کد زبان‌ها (مانند en یا fr به نام کامل آن‌ها مثل English یا French) این کتابخانه اطلاعات استاندارد ISO درباره‌ی کشورها و زبان‌ها را فراهم می‌کند.

Numpy: برای مدیریت و پردازش آرایه‌های عددی (در این پروژه برای تبدیل خروجی مدل fastText به آرایه استفاده می‌شود).

Unicodedata: برای بررسی کاراکترهای متن و اطمینان از اینکه ورودی شامل حروف واقعی زبان باشد (برای فیلتر کردن متن‌های بی‌معنی یا فقط عدد و علامت).

Tkinter: برای ساخت رابط گرافیکی ساده که کاربر بتواند متن وارد کند و نتیجه را به‌صورت پیام دریافت نماید. این کتابخانه به‌صورت پیش‌فرض همراه پایتون نصب می‌شود.

۱. فعال کردن پنجره‌ی Terminal

در پایین صفحه‌ی PyCharm، روی تب Terminal کلیک کنید.

دستورات زیر را به ترتیب وارد کنید:

```
pip install fasttext-wheel
pip install pycountry
pip install numpy
```

نکته مهم درباره‌ی tkinter:

tkinter معمولاً همراه با نصب پایتون در ویندوز به‌صورت پیش‌فرض نصب است و نیازی به نصب با pip ندارد.

بعد از نصب، می‌توانید با اجرای این دستور مطمئن شوید کتابخانه‌ها نصب شده‌اند:

```
pip list
```

- معرفی و دانلود دیتاست و مدل fastText (lid.176.bin)

در این پروژه، برای تشخیص زبان از مدل آموزش‌دیده‌ی fastText استفاده می‌شود که به صورت فایل lid.176.bin ارائه شده است. این مدل، شامل اطلاعات آموزش‌دیده شده روی بیش از ۱۷۶ زبان مختلف است و دقت بالایی در تشخیص زبان متن‌های کوتاه و بلند دارد.

۱. آشنایی با مدل fastText

مدل lid.176.bin توسط تیم تحقیقاتی Facebook AI آموزش داده شده است.

از مقالات موجود در سایت Wikipedia جمع آوری شده است.

این مدل به صورت فایل باینری است که می‌توان آن را در برنامه بارگذاری کرد.

۲. وارد لینک رسمی دانلود مدل شوید:

https://fasttext.cc/docs/en/language-identification.html

فایل lid.176.bin را دانلود کنید.

این لینک در فایل readme در گیت‌هاب پروژه نیز وجود دارد.

فایل دانلود شده را در پوشه‌ی اصلی پروژه (جایی که فایل detectipn.py قرار دارد) ذخیره کنید.

- معرفی توابع اصلی پروژه

در این پروژه چند تابع کلیدی وجود دارد که هر کدام وظیفه مشخصی را بر عهده دارند. در این بخش، عملکرد کلی هر تابع و همچنین روند کلی پروژه شرح داده می‌شود.

۱. get_language_name()

وظیفه: دریافت کد زبان (مثلاً en) و تبدیل آن به نام کامل زبان (مثلاً English).

روش کار: با استفاده از کتابخانه‌ی pycountry، کد زبان را به نام رسمی آن تبدیل می‌کند.

۲. contains_valid_letters()

وظیفه: بررسی اینکه متن ورودی حداقل شامل یک حرف معتبر از یک زبان باشد.

روش کار: کاراکترهای متن را بررسی می‌کند و اگر حداقل یک کاراکتر از دسته‌بندی حروف یونیکد باشد (مثل حروف الفبا)، مقدار True برمی‌گرداند، در غیر این صورت False.

۳. safe_predict()

وظیفه: پیش‌بینی زبان متن با استفاده از مدل fastText به‌صورت امن.

روش کار: ابتدا متن را از کاراکترهای خاص مثل \n و \r پاکسازی می‌کند، سپس مدل fastText را برای پیش‌بینی زبان فراخوانی می‌کند. در صورت بروز خطا، مقدار پیش‌بینی ناشناخته (unknown) با احتمال صفر برمی‌گرداند.

۴. detect_language()

وظیفه: دریافت متن از رابط گرافیکی، اعتبارسنجی متن، پیش‌بینی زبان و نمایش نتیجه به کاربر.

روش کار:ابتدا متن ورودی را می‌گیرد.

بررسی می‌کند که متن خالی یا فاقد حروف معتبر نباشد.

با استفاده از تابع safe_predict زبان متن را پیش‌بینی می‌کند.

نام زبان و درصد اطمینان را با استفاده از messagebox به کاربر نمایش می‌دهد.

در نهایت ورودی متن را پاک می‌کند تا آماده متن جدید باشد.

- **روند کلی و قابلیت‌های پروژه**

رابط کاربری ساده و گرافیکی: کاربر می‌تواند جمله یا کلمه‌ای را در کادر متنی وارد کند و با زدن دکمه یا کلید Enter، زبان متن تشخیص داده شود.

اعتبارسنجی ورودی: قبل از تشخیص زبان، برنامه بررسی می‌کند که ورودی خالی نباشد و حتماً شامل حروف واقعی باشد، تا از پیش‌بینی اشتباه جلوگیری کند.

استفاده از مدل آموزش‌دیده fastText: پروژه از مدل lid.176.bin استفاده می‌کند که قادر به تشخیص ۱۷۶ زبان مختلف است.

نمایش نام زبان و درصد اطمینان: پس از پیش‌بینی، نتیجه به صورت پنجره پیام به کاربر نمایش داده می‌شود.

مدیریت خطاها: در صورتی که مدل هنگام پیش‌بینی دچار خطا شود، برنامه به جای کرش کردن، پیغام مناسب نشان داده و برنامه را پایدار نگه می‌دارد.

- **چطور زبان تشخیص داده می‌شود؟**

مدل fastText روی میلیون‌ها نمونه متن از زبان‌های مختلف آموزش دیده است.

وقتی شما یک متن را به مدل می‌دهید، مدل آن را به قطعات کوچک‌تری به نام n-gram تقسیم می‌کند (مثلاً بخش‌های چند حرفی پشت سر هم).

سپس مدل این n-gramها را بررسی و با داده‌های آموزش دیده مقایسه می‌کند تا ببیند کدام زبان بیشترین تطابق را با آن‌ها دارد.

در نهایت مدل یک یا چند کد زبان را به همراه میزان احتمال (اعتماد یا confidence) ارائه می‌دهد که نشان می‌دهد چقدر مطمئن است متن به آن زبان است.

- **درصد اعتماد چگونه محاسبه می‌شود؟**

معیار اعتماد یا Confidence Score یک عدد بین ۰ و ۱ است که نشان می‌دهد مدل چقدر مطمئن است زبان پیش‌بینی شده برای متن ورودی صحیح است.

این عدد، بر اساس احتمال آماری است که مدل به هر برچسب (کد زبان) نسبت می‌دهد.

مدل fastText پس از پردازش متن و استخراج ویژگی‌ها (مثل n-gramها)، یک تابع احتمال برای هر زبان محاسبه می‌کند.

این احتمال‌ها به صورت یک (softmax) بین تمام زبان‌های ممکن ارائه می‌شوند.

هر زبان یک عدد احتمال (مثلاً ۰,۸۵ برای انگلیسی، ۰,۱۰ برای فرانسوی و...) دریافت می‌کند که مجموع همه احتمال‌ها برابر ۱ است.

زبان با بیشترین احتمال به عنوان زبان تشخیص داده شده انتخاب می‌شود.

نکات:

مقدار اعتماد به کاربر کمک می‌کند که تصمیم بگیرد آیا نتیجه قابل قبول است یا نیاز به بررسی بیشتر دارد.

در پروژه شما، این عدد به صورت درصد نمایش داده می‌شود تا خواندنش ساده‌تر باشد.

اگر مقدار اعتماد خیلی پایین باشد، ممکن است نشان‌دهنده‌ی متن کوتاه، نامفهوم یا دوپهلو بودن زبان باشد.

- ## اجرای پروژه

1. **از سایت گیت‌هاب پروژه که در صفحه اول این فایل قرار دارد، تمامی پوشه‌ها و فایل‌های قرار داده شده را دانلود کنید**
2. **در فایل readme موجود، لینک پوشه Lib و فایل مدل Lid.176.bin را دانلود کنید.**

۳. آنها را در پوشه اصلی پروژه قرار دهید. (فایل مدل حتما باید درکنار فایل اصلی پروژه باشد)

علت وجود نداشتن این دو در فایل‌های اصلی پروژه، پشتیبانی نکردن سایت github از فایل‌ها و پوشه‌های بالای ۱۰۰ مگابایت است.

۴. پس از بارگزاری کردن تمامی فایل ها، فایل اصلی، یعنی detection.py را اجرا کنید.

۵. Run را کلیک کنید.

۶. پنجره رابط گرافیکی باز شده و در کادر متن جمله یا کلمه را در زبان دلخواه خود وارد کنید، در آخر دکمه تشخیص را بزنید.

پاسخ و حدس برنامه به صورت یک messagebox به همراه درصد اعتماد برنامه به تشخیص خود نمایش داده می‌شود.

با تشکر از توجه شما.