دانشگاه اصفهان
دانشکده مهندسی کامپیوتر

تمرین اول: فاز سوم پروژه سخنرانی‌های TED | **استاد:** دکتر احمد براآنی
درس: بازیابی پیشرفته اطلاعات | **دستیار:** الهام اسماعیلی
نام و نام خانوادگی: سید عمید اسدالهی مجد | **شماره دانشجویی:** ۴۰۰۳۶۱۴۰۰۴

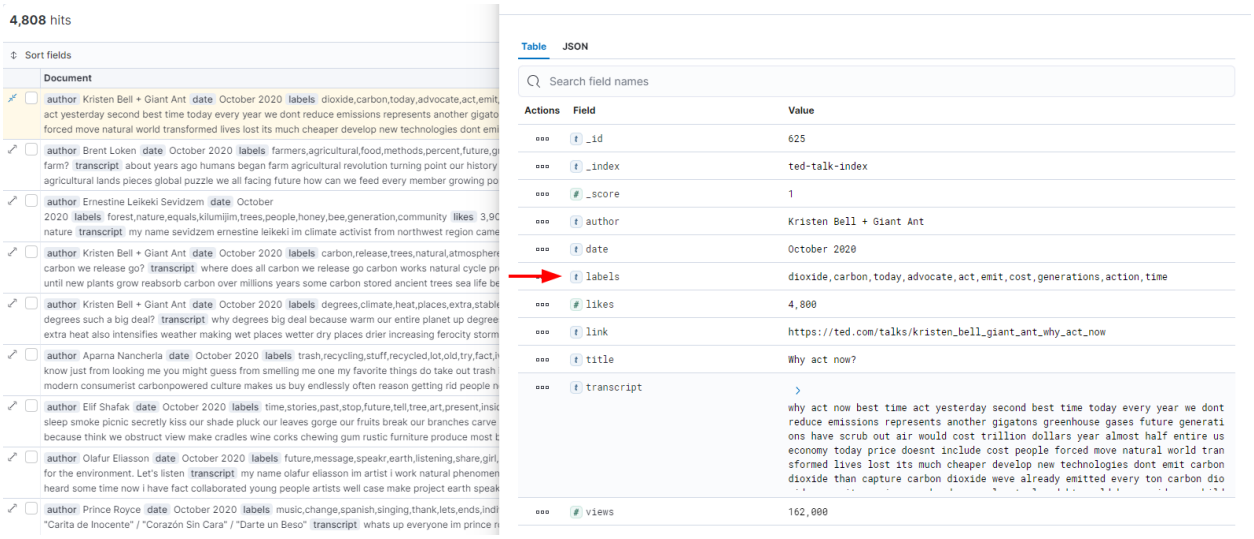
آدرس گیت: https://github.com/amidmajd/ted-talk-classification

یافتن برچسب‌های سخنرانی‌ها با استفاده از روش skip-gram

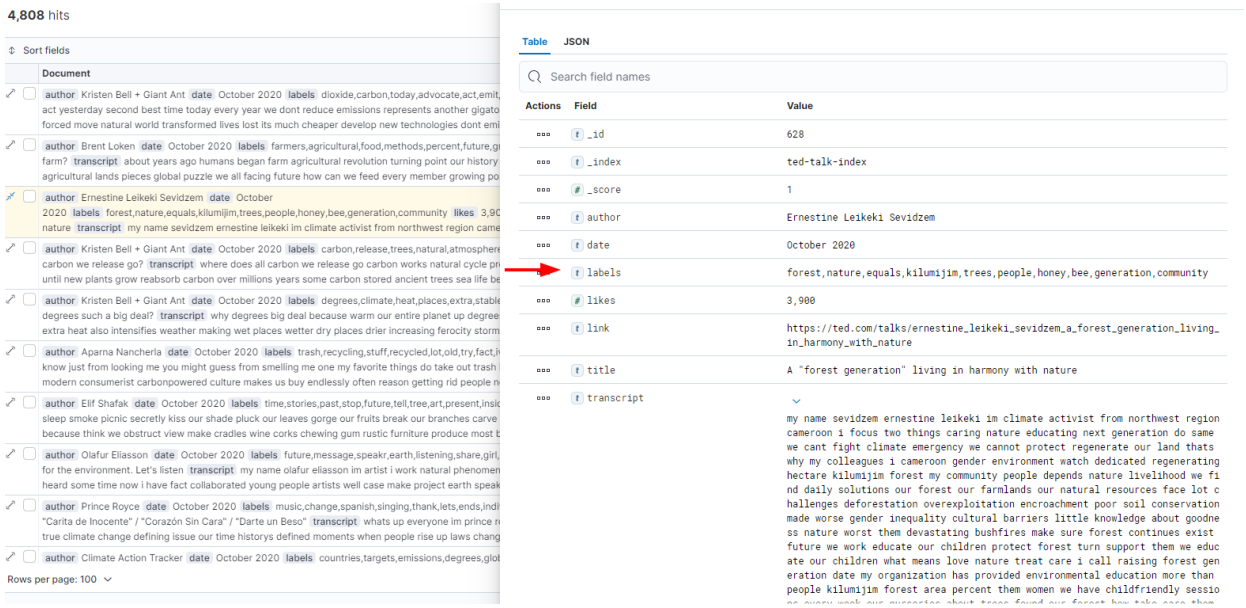
در فاز دوم پروژه با استفاده از روش skip-gram بردار مربوط به transcript هر سنخرانی ساخته‌شد. حال برای استفاده از این بردارها تغییری در کد skip-gram.py داده‌میشود تا با استفاده از این بردادها ده کلمه با بیشترین فرکانس به‌عنوان برچسب‌های هر سخنرانی در الستیک‌سرچ ذخیره شوند. با فرض وجود داده‌های مربوط به سخنرانی‌ها در الستیک‌سرچ (حاصل فاز اول پروژه)، تغییرات زیر برای اجرای فاز سوم پروژه انجام شدند.

در کد skip-gram.py در خط ۲۴ متغیر جدیدی با نام file_path_per_doc تعریف می‌شود که شامل محل ذخیره متن سخنرانی به‌ازای هر id سخنرانی است (برای ساخت بردار با استفاده از کتابخانه FastText نیاز است تا فایل متنی داشته باشیم). در ادامه در خط ۳۶، به ازای هر متن سخنرانی بردار مربوط به transcript ساخته می‌شود. سپس با فراخوانی model.words آرایه‌ای شامل کلمات سخنرانی به ترتیب فراوانی هرکلمه از مدل مربوط به بردارها دریافت می‌شود. پس از حذف کلمات stop-words، ده کلمه با بیشترین فرکانس انتخاب می‌شوند. سپس هریک از داده‌های ایندکس شامل متن سخنرانی‌ها با افزودن فیلد جدیدی با نام labels که شامل برچسب‌های بدست‌آمده است، آپدیت می‌شوند.

در شکل‌های زیر نمونه‌ای از خروجی حاصل از اجرای فایل skip-gram.py را مشاهده می‌نماییم.



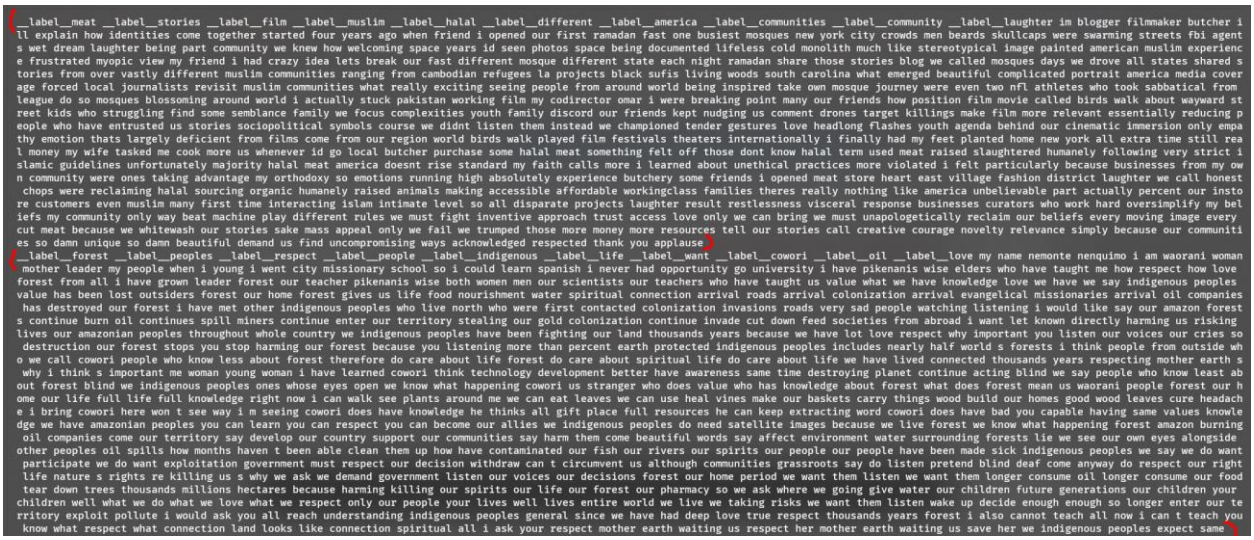
یک سخنرانی به‌همراه برچسب‌های جدید اضافه‌شده به آن



یک سخنرانی به‌همراه برچسب‌های جدید اضافه‌شده به آن

تابع save_transcript_with_labels

در این تابع یک آرایه شامل اطلاعات خروجی الستیک و به فرمت مخصوص آن و همچنین نام فایل خروجی دریافت می‌شود. برای استفاده از classifier کتابخانه FastText باید یک فایل شامل خطوط متن داشته باشیم. هر خط از این فایل، مربوط به متن یک سخنرانی بوده و قبل از متن سخنرانی در هر خط برچسب[1]های مربوط به آن سخنرانی ذخیره می‌شوند. به ابتدای برچسب‌ها باید عبارت __label__ افزوده شود و باهم یک فاصله داشته باشند. در این تابع هر سخنرانی و برچسب‌های مربوط به آن که در مرحله پیشین استخراج شدند از الستیک‌سرچ استخراج می‌شوند و در یک خط با فرمت ذکرشده ذخیره می‌شوند. دو متن سخنرانی به‌همراه برچسب‌های مربوطه را می‌توان در شکل زیر مشاهده نمود. خروجی این تابع به‌شکل زیر خواهد بود. از این تابع برای ذخیره داده‌های test و train به‌صورت جداگانه استفاده می‌شود.



[1] Label

توضیح کد مربوط به دسته‌بندی سخنرانی‌ها

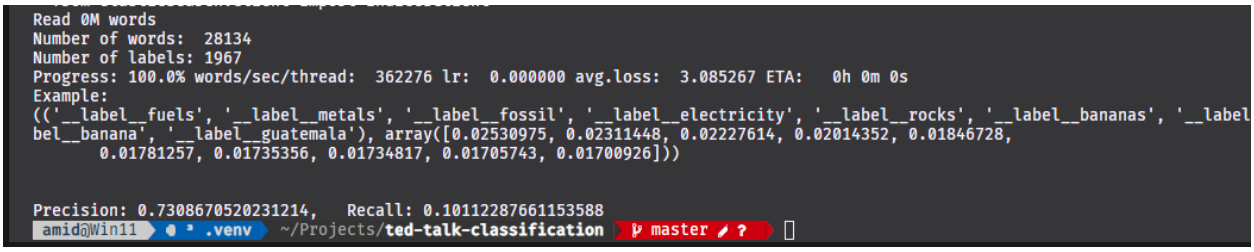
در بدنه‌ی اصلی کد با استفاده از تابع train_test_split از کتابخانه sklearn با نسبت ۹۰ به ۱۰ درصد داده‌ها به train و test تقسیم شدند. سپس با استفاده از تابع save_transcript_with_labels این داده‌ها با فرمت مربوطه با تحت عناوین test_data.txt و train_data.txt ذخیره شدند.

سپس با استفاده از تابع train_supervised از کتابخانه FastText و با ورودی train_data.txt به‌عنوان فایل آموزش، نرخ یادگیری[2] برابر با ۱.۲۵، ngramهای ۳تایی و تعداد اجرای[3] برابر با ۵۰۰۰ مدل آموزش دیده و ساخته می‌شود. سپس مدل را با نام classifier_model.bin ذخیره می‌نماییم تا در آینده بتوان آن را بارگیری نمود و استفاده کرد.

در ادامه با فراخوانی تابع test با ورودی test_data.txt مقادیر precision و recall بدست آمدند و نمایش داده می‌شوند که در شکل زیر قابل مشاهده است. برای نمونه متن زیر به مدل داده‌شده است و مدل برچسب‌هایی مثل fuels، metals، fossil، electricity، bannana و ... را با دقت‌های نمایش داده‌شده در شکل زیر پیشبینی کرده‌است.

متن ورودی:

We currently have enough fossil fuels to progressively transition off of them, says climate campaigner Tzeporah Berman, but the industry continues to expand oil, gas and coal production and exploration. With searing passion and unflinching nerve, Berman reveals the delusions keeping true progress from being made -- and offers a realistic path forward: the Fossil Fuel Non-Proliferation Treaty.



خروجی کار به همراه precision و recall مدل ساخته‌شده (precision برابر با ۷۳ درصد و recall برابر با ۱۰ درصد)

[2] Learning Rate
[3] Epoch