دانشکده مهندسی کامپیوتر

استاد درس: سید صالح اعتمادی

بهار ۱۴۰۲

تحلیل احساسات روی ویدئوهای نقد فیلم‌های سینمایی

درس پردازش زبان‌های طبیعی

حوریه سبزواری

شماره دانشجویی: ۹۸۴۱۲۰۰۴

گزارش فاز دوم

موضوع پروژه تسک analysis sentiment روی ویدیوهای نقد و بررسی فیلم‌های سینمایی است. داده‌ها از منتقدهای مشغول به فعالیت در پلتفرم یوتیوب جمع‌آوری می‌شود. در انتها برای هر داده‌ی ورودی مشخص می‌گردد که نظر کلی ویدیو راجع به آن فیلم سینمایی خوب، متوسط یا بد بوده است.

لینک Repository: https://github.com/lhoorie/SentimentAnalysisOnReviewVideos

۱ Word2Vec

- ابتدا مدل Word2Vec از کتابخانهٔ gensim را برای هر ۳ برچسب آموزش می‌دهیم.
- سپس مدل‌های نهایی را با نام <label>.word2vec.bin در پوشهٔ models ذخیره می‌کنیم.
- حال کلمات مشترک بین ۳ برچسب را پیدا کرده و بردارهای آن‌ها را با استفاده از مقایسهٔ شباهت کسینوسی میزان شباهت و تفاوت کلمات را می‌یابیم.
- در انتها نیز یک مدل word2vec روی تمام داده آموزش می‌دهیم.

کلمات با بردارهای مشابه و متفاوت در خروجی چاپ شده در کد موجود در notebook قسمت Word2Vec قابل مشاهده است. تفاوت احتمالاً به این دلیل است که Word2Vecجاسازی های کلمه را بر اساس زمینه‌ای که کلمه در آن ظاهر می شود، یاد می‌گیرد. بردارهای کلمه برای به تصویر کشیدن معنا و روابط بین کلمات در زمینهٔ خاص داده های آموزشی آموزش داده شده‌اند. هنگام آموزش Word2Vecالگوریتم از یک رویکرد پنجرهٔ کشویی برای گرفتن متن هر کلمه استفاده می‌کند. احتمال وقوع یک کلمه را بر اساس کلمات همسایه آن پیش‌بینی می کند. در نتیجه، بردارهای کلمه آموخته شده بسته به کلمات اطرافی که کلمه در آن ظاهر می‌شود، می‌تواند متفاوت باشد.

۲ Tokenizer

در این بخش ابتدا از tokenizer گفته شده در صورت سوال استفاده کردم و خروجی درستی نمی‌داد. سپس از tokenizer BERT استفاده کردم و تنظیمات مربوطه و خود مدل را در پوشهٔ models ذخیره کردم. BERT متن را با در نظر گرفتن قوانین خاص زبان به کلمات جداگانه تقسیم می کند. به عنوان مثال، ”running” را به ”run” و ”ning” تقسیم می کند تا پردازش زیر کلمه را فعال کند.

BERT از تکنیکی به نام WordPiece استفاده می کند. این بیشتر کلمات را به زیرکلمه ها یا دنباله های کاراکتر تقسیم می کند تا کلمات خارج از واژگان (OOV) را مدیریت کند و تغییرات مشخص شود.

BERT نشانه‌های خاصی را برای علامت‌گذاری شروع و پایان یک دنباله، جملات جداسازی و نشان دادن padding اضافه می‌کند. این نشانه ها عبارتند از [CLS] (طبقه بندی)، [SEP] (جداکننده) و [PAD]

یک ماسک attention نیز ایجاد می‌شود تا نشان دهد کدام نشانه‌ها بخشی از دنباله ورودی هستند و کدام نشانه‌های padding هستند. به مدیریت توالی های ورودی با طول متغیر کمک می کند.

۳ Language Model

در این بخش از مدل از پیش آموختهٔ OpenAI بنام GP2 استفاده کردم. مدل را بر روی داده‌های هر دسته finetune کرده و در پوشهٔ models ذخیره کردم. کیفیت جملات تولیدشده پس از یکی دو خط بسیار افت می‌کرد و یا جملات تکراری تولید می‌کرد. این مورد می‌تواند ناشی از این باشد که GPT-۲ متن را به صورت متوالی پردازش می کند و بر زمینه ارائه شده در توکن های قبلی متکی است. با این حال، ممکن است همیشه وابستگی های دور را شامل نشود یا تفاوت های ظریف بافتی پیچیده را درک نکند. در نتیجه، گاهی اوقات می تواند متنی تولید کند که فاقد انسجام باشد یا ناسازگاری های معنایی را نشان دهد.

۴ Feature Engineering

ویژگی‌های گفته‌شده در در صورت سوال در کد موجود پیاده‌سازی شده اما ران کردن این بخش باعث پر شدن رم Google Collab شد و متاسفانه نتیجه‌ای حاصل نگردید.

۵ Model Architecture

در این قسمت از معماری BERT استفاده شده است، دقت و loss و خود مدل در پوشهٔ مربوطه ذخیره شده‌اند. دقت حدودا ۵۰ درصد بدست آمد که به نظر ناشی از طول جملات بالا و جمله‌بندی نبودن متن است.

۶ Data Augmentation

با استفاده از API ChatGPT و propmt های داده‌شده اقدام به داده‌افزایی کردم. داده‌های تولیدشده از نظر طول متن بسیار کوتاه‌تر از داده‌های اصلی بودند. اما از نظر محتوا نسبت به برچسب داده‌شده دارای کیفیت مطلوبی داشتند.

Prompt for label 0 : prompt = ”Generate a transcript review of a bad movie:”
Prompt for label 1 : prompt = ”Generate a transcript review of a so-so and neutral movie:”
Prompt for label 2 : prompt = ”Generate a transcript review of a good movie:”

نمونه داده‌های تولیدشده در کد قابل مشاهده می‌باشد.