به نام خدا

کارگاه هضم NLP بهاره كاوسي نژاد – 99431217

سوالات تئوری

1. روند شروع به کار یک محصول ASR (Automatic Speech Recognition) و توسعه آن را با توجه به نکات گفته شده سر کارگاه، توضیح دهید.

محصولی مانند حرف، ابتدا با کمک مدل هایی مثل google و mozilla کار خود را آغاز کرد. پس از تبدیل شدن به یک مدل، با اضافه کردن دیتاهایی که متن آنها نیز در دسترس بود مانند پادکست ها و کتاب های صوتی، این مدل بهبود یافت و به دلیل تمرکز روی زبان فارسی حتی از مدل های google و mozilla نیز بهتر شد و اکنون دارای درصد دقت بالاتری است. با این وجود همچنان در برخی از موارد از این مدل ها استفاده می شود.

به طور کلی این روند را می توان اینگونه شرح داد:

1. فرض کنید چند مدل (برای مثال harf – whisper – mozilla – google – nemo - ...) برای تبدیل صوت به متن فارسی در دسترس داریم و هرکدام 16 خروجی با ترتیب اولویت احتمال درستی میدهد، برای اینکه از بین این تعداد زیاد خروجی، بتوانیم تعداد 16 خروجی نهایی انتخاب کنیم و خروجی دهیم حداقل دو روش (که در عمل استفاده می شود یا اگر روش جدیدی به ذهنتان میرسد که منطقی است توضیح دهید).

پس از خروجی گرفتن از مدل ها به آنها ضریبی را نسبت می دهیم. به عنوان مثال تعداد تکرار یک کلمه در هر 16 خروجی هر مدل را بدست می آوریم. اگر تعداد تکرار کلمه ای زیاد بود، احتمال وجود آن در متن خروجی بیشتر است، در نتیجه به آن وزن بیشتری اختصاص می دهیم. همچنین اگر کلمه ای جزء خروجی های اول مدل ها بود، وزن آن خیلی بیشتر خواهد بود.

بعد از بدست آمدن وزن کلمات، ماتریسی ساخته می شود که بیانگر تعداد تکرار هر کلمه و اهمیت آن است.

روش دیگر استفاده از Deep Learning برای ترکیب کردن مدل ها می باشد.

همچنین برای ارزیابی مدل های ASR می توان سنجش را بر اساس کلمه و یا بر اساس کاراکتر انجام داد.

همچنین یک Language Model داریم که احتمالات را پیش بینی می کند.

1. مفهوم توابع normalizer، formalizer، lemmatizer، stemmer، chunker، tagger، postagger، embedder، wordembedder و parser را در دنیای پردازش متن توضیح دهید.

* normalizer: normalizer در NLP به تابع یا فرآیندی اشاره دارد که متن را به فرم استاندارد یا نرمال شده تبدیل می کند. معمولاً شامل تبدیل متن با حذف علائم نگارشی، تبدیل به حروف کوچک، حذف لهجه‌ها، گسترش contractionها و مدیریت سایر قراردادهای خاص زبان است. هدف از نرمال سازی ایجاد یک نمایش ثابت از متن است که تجزیه و تحلیل و پردازش را آسان تر می کند.
* formalizer: formalizer تابعی است که در NLP برای تبدیل متن غیررسمی یا محاوره‌ای به فرمی رسمی یا استاندارد استفاده می‌شود. ممکن است شامل جایگزینی کلمات عامیانه با معادل های رسمی آنها، تصحیح اشتباهات املایی و بهبود ساختارهای دستوری برای رعایت قوانین زبان رسمی باشد. هدف formalizer افزایش خوانایی، اطمینان از وضوح و حفظ ثبات در متن است.
* lemmatizer: lemmatizer تابعی است که کلمات را به شکل پایه یا متعارف خود کاهش می دهد که به آن لم می گویند. برای استخراج لم یک کلمه، بافت و بخشی از گفتار آن را در نظر می گیرد. برای مثال، لم «running»، «run» و لم «better»، «good» خواهد بود. Lemmatization به کاهش inflected fromهای کلمات به شکل ریشه مشترک آنها کمک می کند و به کارهایی مانند تجزیه و تحلیل متن، بازیابی اطلاعات و یادگیری ماشین کمک می کند.
* Stemmer: یک stemmer تابعی است که کلمات را به شکل پایه یا ریشه خود کاهش می دهد که به عنوان stem شناخته می شود. بر خلاف lemmatizerها، stemmerها بافت یا بخشی از گفتار یک کلمه را در نظر نمی گیرند. آنها قواعد ساده زبانی را برای حذف پیشوندها یا پسوندها از کلمات اعمال می کنند، که اغلب منجر به stem هایی می شود که کلمات واقعی فرهنگ لغت نیستند. به عنوان مثال، stem "running"، "run" و stem "better" "bet" خواهد بود. Stemming معمولاً برای normalize کردن متن برای کارهایی مانند indexing، جستجو و بازیابی اطلاعات استفاده می شود.
* Chunker: یک chunker تابعی است که کلمات را بر اساس ساختار نحوی آنها در واحدهای معنی دار به نام chunk گروه بندی می کند. Chunk ها می توانند از ترکیب های مختلفی از کلمات مانند noun phrases، verb phrases یا prepositional phrases تشکیل شوند. Chunking معمولاً برای استخراج اطلاعات خاص از متن استفاده می شود، مانند named entities یا key phrases.
* tagger: یک tagger، تابعی است که تگ های دستوری یا Part Of Speech (POS) را به هر کلمه در یک جمله اختصاص می دهد. تگ های POS نشان دهنده نقش و دسته بندی یک کلمه در یک جمله هستند، مانند اسم، فعل، صفت یا قید. برچسب‌گذاری یک مرحله ضروری در بسیاری از وظایف NLP از جمله تجزیه، تحلیل احساسات و استخراج اطلاعات است.
* postagger: postagger که مخفف عبارت part-of-speech tagger است، نوع خاصی از برچسب گذاری است که بر اختصاص تگ های POS به کلمات در یک جمله تمرکز دارد. ساختار نحوی یک جمله را تجزیه و تحلیل می کند و هر کلمه را با برچسب POS مربوطه آن برچسب گذاری می کند.
* embedder: یک تابع یا الگوریتمی است که کلمات یا جملات را به نمایش های dense و numerical تبدیل می کند که به آن word embeddings یا sentence embeddings می گویند. Enbedding‌ها اطلاعات معنایی و متنی کلمات یا جملات را دریافت می‌کنند و ماشین‌ها را قادر می‌سازند تا زبان طبیعی را درک و پردازش کنند. آنها به طور گسترده در وظایف مختلف NLP مانند تجزیه و تحلیل احساسات، ترجمه ماشینی و طبقه بندی اسناد استفاده می شوند.
* wordembedder: wordembedder نوع خاصی از embedder است که word embeddingها را تولید می کند. بر اساس روابط معنایی بین کلمات، کلمات جداگانه را به بردارهای dense و numerical در فضایی با ابعاد بالا نگاشت می کند. این تعبیه‌ها را می‌توان برای اندازه‌گیری شباهت کلمات، انجام word analogies یا به عنوان ویژگی‌های ورودی برای وظایف NLP که downstream هستند، استفاده کرد.
* parser: parser تابع یا الگوریتمی است که ساختار دستوری یک جمله را تجزیه و تحلیل می کند و به هر کلمه برچسب های نحوی مانند فاعل، مفعول یا فعل اختصاص می دهد. هدف آن درک روابط بین کلمات و چگونگی تشکیل یک جمله معنادار است. Parsing برای کارهایی مانند درک جمله، پاسخ به سؤال و تجزیه و تحلیل نحوی بسیار مهم است.

**منابع:**

* <https://whites.agency/blog/open-domain-question-answering-introduction-to-the-topic/>
* <https://hyperskill.org/learn/step/27243>
* <https://medium.com/analytics-vidhya/open-domain-question-answering-series-part-1-introduction-to-reading-comprehension-question-1898c8c9560e>
* <https://medium.com/analytics-vidhya/open-domain-question-answering-series-part-1-introduction-to-reading-comprehension-question-1898c8c9560e>
* <https://www.mdpi.com/2076-3417/9/18/3698>
* <https://conservancy.umn.edu/items/90239401-18bf-4956-86c1-502b2a2ad328>
* <https://www.baeldung.com/cs/rnns-transformers-nlp>
* <https://www.linkedin.com/pulse/deep-dive-positional-encodings-transformer-neural-network-ajay-taneja/>
* <https://www.linkedin.com/pulse/transformer-architectures-dummies-part-2-decoder-only-bhaskar-t-hj9xc/>
* <https://blog.stackademic.com/mastering-nlp-simplified-guide-to-abstractive-vs-802e5e6c0a26>
* <https://huggingface.co/spaces/exbert-project/exbert>