دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

دانشکده مهندسی کامپیوتر

پروژه تحقیقاتی درس شبکه‌های عصبی

مدل برت و نسخه‌های بهبود یافته‌ی آن

نگارش

زهرا زنجانی

استاد درس

دکتر رضا صفابخش

بهار ۱۴۰۲


بسم الله الرحمن الرحيم

چکیده

مدل برت[1] و مدل‌های زبانی از پیش آموزش‌دیده، پردازش زبان طبیعی [2] با گرفتن تعبیه‌های متنی کلمه و امکان یادگیری انتقالی متحول کرده‌اند. توانایی مدل برت برای درک تفاوت های ظریف زبان و استفاده از حجم زیادی از داده های بدون برچسب برای پیش آموزش، دقت و عملکرد وظایف پردازش زبان طبیعی را به طور قابل توجهی بهبود بخشیده و از مدل های قبلی پیشی گرفته و به عملکرد در سطح انسانی دست یافته است.در این گزارش به اهمیت مدل‌های برت و پیشرفت و بهینه سازی های معرفی شده در سایر مدل های برپایه ی برت مانند البرت[3] و روبرتا[4] می‌پردازیم.

واژه‌های کلیدی:

مدل‌های زبانی ازپیش آموزش دیده ، برت

[1] Bert

[2] pretrained language model

[3] ALBert

[4] RoBerta

فهرست مطالب

عنوان صفحه

فهرست اشکال

شکل صفحه

فهرست جداول

جدول صفحه

فصل اول

مقدمه

در سال‌های اخیر، مدل‌های زبانی از پیش آموزش‌دیده به عنوان یک پیشرفت بزرگ در پردازش زبان طبیعی پدیدار شده‌اند که با بهبود قابل‌توجه دقت مدل‌های پردازش زبان طبیعی و ساخت برنامه‌های با کیفیت بالا برای توسعه‌دهندگان و محققان، انقلابی در این زمینه ایجاد کرده‌اند. مدل‌های از پیش آموزش دیده مبتنی بر معماری ترنسفورمر ، مانند جی‌پی‌تی [1] و برت [2]، روی مقادیر زیادی از داده‌های متنی از قبل آموزش دیده‌اند که به آن‌ها امکان می‌دهد طیف گسترده ای از الگوها و ویژگی‌های زبانی را بیاموزند. پیش آموزش و تنظیم دقیق مدل‌ها را برای وظایف خاص پردازش زبان طبیعی، مانند طبقه‌بندی متن، تحلیل احساسات و ترجمه ماشینی با حداقل داده‌های آموزشی اضافی ممکن می‌سازد. در نتیجه، مدل‌های از پیش آموزش‌دیده، ساخت برنامه‌های پردازش زبان طبیعی با کیفیت بالا را برای توسعه‌دهندگان آسان‌تر کرده و پیشرفت در این زمینه را تسریع کرده‌اند که منجر به پیشرفت‌هایی در زمینه‌هایی مانند درک زبان و تولید زبان طبیعی شده است. در بین این مدل‌ها، برت [3]، روبرتا [4]، ایکس‌ال‌نت [5]، البرت [6] و الکترا [7] جزو پرکاربردترین مدل‌ها هستند.

برت که توسط گوگل در سال ۲۰۱۸ معرفی شد، نقطه عطف مهمی در مدل‌های زبانی از پیش آموزش‌دیده بود. مکانیزم توجه به خود دو سویه به مدل امکان می‌دهد بافت کلمات یک جمله را با دقت بیشتری ثبت کند و معماری ترنسفورمر به برت اجازه می‌دهد کل جملات را پردازش کند و روابط بین کلمات را درک کند. علاوه بر این، برت با استفاده از مدل‌سازی زبان ماسک‌دار روی حجم زیادی از داده‌ها پیش آموزش دیده است که به آن امکان می‌دهد طیف گسترده‌ای از ویژگی‌ها و الگوهای زبانی را بیاموزد و پیش‌بینی‌های بسیار دقیقی درباره معنای جملات جدید انجام دهد. به طور کلی، پیشرفت‌های برت تأثیر قابل توجهی بر پردازش زبان طبیعی داشته است و فرصت‌های جدیدی را برای محققان و توسعه‌دهندگان در این زمینه باز کرده است.

روبرتا با استفاده از داده‌های آموزشی اضافی و تکنیک‌های بهینه سازی برت را بهبود می‌بخشد، نتایج بهتری در وظایف خاص پردازش زبان طبیعی به دست می‌آورد. ایکس‌ال‌نت یک رویکرد پیش‌آموزش مبتنی بر جایگشت را معرفی می‌کند که به مدل اجازه می‌دهد وابستگی‌های بین کلمات را بهتر ثبت کند و منجر به بهبود بیشتر در عملکرد شود. البرت با استفاده از تکنیک‌های به اشتراک‌گذاری پارامتر تعداد پارامترهای مدل را کاهش می‌دهد و آموزش آن را کارآمدتر و سریع تر می‌کند.

اسپن‌برت و برت ساختاری اهداف پیش آموزش دیگری را دربرمی‌گیرند که به مدل اجازه می‌دهد تا رابطه بین بخش‌های مختلف متن را بهتر به تصویر بکشد. اسپن‌برت هم بر روی داده‌های جمله و هم در سطح موجودیت‌های جمله از قبل آموزش داده شده است تا ارتباط بین بخش‌های مختلف متن را بهتر به تصویر بکشد، در حالی که برت ساختاری از یک ساختار سلسله مراتبی برای مدل سازی روابط بین کلمات و

[1] GPT
[2] BERT
[3] BERT
[4] RoBERTa
[5] XLNet
[6] ALBERT
[7] ELECTRA

جملات استفاده می‌کند.الکترا با تولید داده‌های آموزشی مصنوعی از طریق یک تکنیک جدید به نام پیش آموزش متمایزگر[8] برت را بهبود می‌بخشد.
در این گزارش، مروری بر این مدل‌ها ارائه می‌کنیم، در مورد معماری، وظایف، اهداف پیش آموزش و عملکرد آن‌ها در وظایف مختلف پردازش زبان طبیعی بحث می‌کنیم. همچنین بهبودهایی را که هر مدل نسبت به برت ایجاد کرده است بررسی خواهیم کرد تا درک جامعی از این مدل‌ها و کاربردهای آنها در پردازش زبان طبیعی ارائه کنیم.ادامه ساختار این پروژه به شرح زیر است:

- در بخش دوم معماری برت و نحوه نحوه‌ی پیش آموزش و تنظیم پارامترهای برت آن برای کاربردهای پردازش زبان طبیعی تشریح خواهد شد.
- در بخش سوم روش‌های بهبود پیش آموزش مدل برت مطرح می‌شود و ایده‌های ارایه شده توسط مدل‌های روبرتا، البرت،برت ساختاری و سایر مدل‌ها مطرح می‌شود.
- در بخش چهارم ایده‌های جهت بهبود معماری برت و مدل‌های برپایه برت بررسی می‌شود.
- در بخش پنجم نتایج مدل‌های بیان شده با یکدیگر مقایسه می‌شود.
- نهایتا بخش ششم مروری بر مطالب این پروژه خواهد بود.

[8] discriminative pre-training

فصل دوم

مدل برت

تکنیک پیش‌آموزش برت انقلابی در زمینه پردازش زبان طبیعی ایجاد کرده است. برت یک معماری شبکه عصبی است که روی مقادیر زیادی از داده‌های متنی بدون برچسب از قبل آموزش داده شده است و به آن امکان می‌دهد بازنمایی‌های متنی کلماتی را که می‌توانند برای طیف وسیعی از وظایف پردازش زبان طبیعی تنظیم شوند، یاد بگیرد.در این فصل قرار است ساختار و نوآوری‌های مدل برت را مورد بررسی قرار می‌دهیم.

۲-۱ مدل‌های از پیش آموزش دیده

در تحقیقات اخیر نشان داده است که پیش آموزش مدل زبان برای بهبود بسیاری از کاربرد پردازش زبان طبیعی مؤثر است[۴] [۱۲]. این مدل‌ها روی حجم زیادی از دادگان بدون برچسب آموزش می‌بینند و الگوهای پیچیده و درک عمومی زبان را می‌آموزند.این مدل‌ها نیاز به برچسب گذاری گسترده را کاهش می‌دهند و امکان تجزیه و تحلیل دقیق‌تر و ظریف‌تر متن را فراهم کرده و موجب پیشرفت‌هایی را در زمینه‌های پردازش زبان طبیعی مانند بازیابی اطلاعات، ربات‌های گفتگو و ترجمه ماشینی شده‌اند.

دو استراتژی برای استفاده از مدل‌های زبانی از پیش آموزش دیده وجود دارد: رویکرد مبتنی بر ویژگی و رویکرد مبتنی بر تنظیم دقیق وزن‌ها . رویکرد مبتنی بر ویژگی مانند مدل المو[1] بازنمایی‌های از پیش آموزش دیده را به عنوان ویژگی‌های اضافی در معماری‌های مخصوص هر وظیفه ترکیب می‌کند. رویکرد مبتنی بر تنظیم دقیق وزن‌ها مانند برت و جی پی تی[2] بدون ایجاد تغیرات قابل توجه در مدل زیربنای آموزش با تنظیم دقیق پارامترها استفاده از مدل را برای هر کاربرد خاص میسر می‌سازد.

۲-۲ ساختار کلی مدل برت

معماری مدل برت شامل کدگذار ترنسفورمر دو طرفه چند لایه است پیاده سازی این مدل تقریباً مشابه نسخه اصلی ترنسفورمر است.ورودی برت دنباله ای از نشانه ها است، مانند کلمات یا زیرکلمه‌ها، که هر کدام با یک بردار با اندازه ثابت از تعبیه‌ها نشان داده می‌شوند. این تعبیه‌ها سپس به لایه‌های ترانسفورماتور وارد می‌شوند، که مکانیسم‌های خودتوجهی را برای محاسبه نمایش‌های متنی هر نشانه در دنباله اعمال می‌کنند. خروجی برت به کاربرد پردازش زبان طبیعی خاصی که برای آن استفاده می‌شود بستگی دارد. به عنوان مثال، برای طبقه بندی متن، آخرین حالت پنهان نشانه $[CLS]$ که به ابتدای دنباله ورودی اضافه می‌شود، به عنوان ورودی لایه طبقه بندی استفاده می شود. برای پاسخگویی به سؤال، خروجی ممکن است شامل موقعیت های شروع و پایان پاسخ در توالی ورودی باشد.
مدل برت با استفاده از مکانیزم توجه دو طرفه محدودیت مدل‌های قبلی برای استفاده از قدرت بازنمایی از

[1] ELMO
[2] □□□□□□

پیش آموزش دیده را رفع می‌کند. در مدل‌های قبلی مانند جی‌پی‌تی که از معماری چپ به راست استفاده می‌کنند. هر توکن فقط می‌تواند به توکن‌های قبلی دسترسی داشته باشد در حالی که در کاربردهایی مانند پاسخ‌گویی به سوال گنجاندن اطلاعات توکن‌ها از هردو جهت لازم است. بنابر این این مدل از بازنمایی کدگذار دوجهته ی ترانسفور استفاده می‌کند و محدودیت مدل یک سویه را کاهش می‌دهد[؟].

۲-۳ آموزش مدل برت

مدل برت مخفف بازنمایی کدگذار دوجهته از مدل ترنسفورمر برای آموزش بازنمایی‌های دوسویه عمیق از متن بدون برچسب با شرطی‌سازی مشترک در زمینه چپ و راست در همه لایه‌ها طراحی شده است. این مدل را می‌توان با یک لایه خروجی اضافی تنظیم کرد تا مدل‌های پیشرفته‌ای را برای طیف وسیعی از کارها، مانند پاسخ‌گویی به سؤال و استنتاج زبان ایجاد کند و نیاز به بسیاری از معماری‌های خاص مهندسی شده برای هر کاربرد پردازش زبان طبیعی را کاهش می‌دهد. چهارچوب کلی آموزش مدل برت به دو قسمت تقسیم می‌شود: پیش آموزش و تنظیم دقیق پارامترها.

۲-۳-۱ پیش آموزش برت

مرحله ی پیش آموزش شامل دو مرحله ی اصلی است: مدل سازی زبان ماسک شده و پیش بینی جمله‌ی بعدی . در مدل سازی زبان ماسک شده حدود ۱۵ درصد کلمات ورودی به صورت تصادفی ماسک می‌شوند و مدل تلاش می‌کند که کلمات ماسک شده را براساس کلمات اطراف ماسک پیش‌بینی کند و کل جمله را بازسازی کند. این فرایند به مدل کمک می‌کند وابستگی‌های متنی و بازنمایی زبانی غنی کل جمله را بیاموزد. در مرحله ی پیش‌بینی جمله‌ی بعدی ، مدل برت جفت جملات را از پیکره ورودی می‌گیرد، این جفت جملات ممکنه است به صورت جملات متوالی و یا به صورت جملات تصادفی انتخاب شده باشند. در این مرحله مدل یاد می‌گیرد که پیش بینی کند که آیا جمله‌ی دوم پیرو جمله‌ی اول در متن اصلی است یا خیر. این کار به برت کمک می‌کند تا روابط و انسجام بین جملات را درک کند و توانایی آن را برای درک معناشناسی سطح گفتمان افزایش دهد. پیش‌بینی جمله‌ی بعدی و درک روابط بین جملات در بسیاری از وظایف پردازش زبان طبیعی مانند پاسخ به سوالات و استنتاج زبان طبیعی مورد نیاز است[۷].

۲-۳-۲ تنظیم دقیق پارامترها در برت

مرحله‌ی تنظیم دقیق پارامترها به گونه ای طراحی شده است که مدل را قادر می‌سازد تا با تعویض ورودی و خروجی‌ها به طور موثر بر روی وظایف مختلف پردازش زبان طبیعی پایین دستی کار کند. در مرحله ی قبلی مدل ارتباط بین کلمات و جملات را یاد گرفته است. تنظیم دقیق پارامترها در وظایف خاص به مدل اجازه می‌دهد تا این دانش آموخته شده را منتقل کند و آن را با وظیفه هدف تطبیق دهد.لایه‌های

ویژه هر وظیفه در این مرحله به مدل افزوده می‌شوند. این لایه‌ها بر اساس نیازهای خاص وظیفه هدف طراحی شده‌اند. به عنوان مثال، برای تجزیه و تحلیل احساسات، ممکن است یک لایه طبقه بندی برای پیش بینی برچسب‌های احساسات اضافه شود. معماری و ساختار این لایه‌ها را می‌توان متناسب با کار در دست سفارشی سازی کرد و به برت اجازه می‌دهد تا وظایف مختلفی مانند طبقه‌بندی متن، تشخیص موجودیت نام‌گذاری شده و پاسخ‌گویی به سؤال را انجام دهد[۷].

۲-۴ تفاوت المو ، جی‌پی‌تی و برت

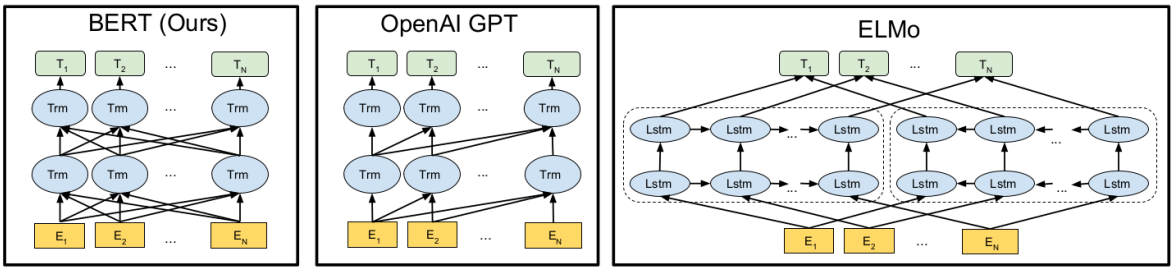
در اینجا ما تفاوت‌های مدل‌های یادگیری برت با مدل‌های قبلی مانند المو و جی‌پی‌تی را بررسی می‌کنیم. مقایسه‌های بین معماری‌های مدل به صورت بصری در شکل ۲-۱نشان داده شده است. این مدل‌ها علاوه بر معماری در رویکردهای پیش آموزش نیز تفاوت دارند، رویکرد المو یک رویکرد مبتنی بر ویژگی است. قابل مقایسه‌ترین روش پیش آموزش موجود با برت جی‌پی‌تی است که یک مدل زبانی مبتنی بر ترنسفورمر از چپ به راست را بر روی یک کورپوس[۳] بزرگ آموزش می‌بینند. در واقع، بسیاری از تصمیمات طراحی در برت عمداً برای نزدیک کردن آن به جی‌پی‌تی تا حد امکان اتخاذ شده است تا بتوان این دو روش را به حداقل ممکن مقایسه کرد. بحث اصلی این کار این است که دو سویه بودن مکانیزم توجه و دو وظیفه پیش آموزش ارائه‌شده در بخش ؟؟این فصل اکثر پیشرفت‌های تجربی را شامل می‌شوند، اما توجه داریم که چندین تفاوت دیگر بین نحوه آموزش برت و جی‌پی‌تی وجود دارد:

- جی‌پی‌تی در کورپوس بوک[۴] (۸۰۰ میلیون کلمه) آموزش دیده است. برت بر روی کورپوس بوک (۸۰۰ میلیون کلمه) و ویکی‌پدیا (۲۵۰۰ میلیون کلمه) آموزش دیده است.
- جی‌پی‌تی از یک جداکننده جمله ($[SEP]$) و نشانه طبقه بندی کننده ($[CLS]$) استفاده می‌کند که فقط در زمان تنظیم دقیق استفاده می‌شوند. برت تعبیه‌های $[SEP]$، $[CLS]$ را در طول پیش آموزش یاد می‌گیرد.
- جی‌پی‌تی برای $1M$ گام با اندازه دسته ای ۳۲۰۰۰ کلمه آموزش داده شد. برت برای $1M$ گام با اندازه دسته ای ۱۲۸۰۰۰ کلمه آموزش دیده است.
- جی‌پی‌تی از نرخ یادگیری یکسان ۵-۵برای تمام آزمایش‌های تنظیم دقیق استفاده کرد. برت یک نرخ یادگیری دقیق برای کار خاص را انتخاب می‌کند که بهترین عملکرد را در مجموعه توسعه دارد.

[3] Corpus

[4] book corpus

آزمایش‌های فرسایشی [۵] نشان می‌دهد اکثر پیشرفت‌ها در واقع از دو وظیفه پیش آموزش و دو جهتی بودن مکانیزم توجه ناشی می‌شود.



شکل ۲-۱: [۷]مقایسه‌ی برت و جی‌پی‌تی و المو

تفاوت در معماری مدل قبل از آموزش برت از یک ترانسفورماتور دو طرفه استفاده می کند. جی‌پی‌تی از ترنسفورمر چپ به راست استفاده می کند. المو از الحاق حافظه‌ی کوتاه‌مدت ماندگار [۶] چپ به راست و راست به چپ به طور مستقل آموزش دیده برای ایجاد ویژگی‌هایی برای وظایف پایه‌ی پردازش زبان طبیعی استفاده می‌کند. در میان این سه، تنها نمایش برت به طور مشترک در هر دو زمینه چپ و راست در همه لایه‌ها مشروط می‌شوند. علاوه بر تنظیمات معماری، برت و جی‌پی‌تی مبتنی بر رویکردهای تنظیم دقیق هستند، در حالی که المو یک رویکرد مبتنی بر ویژگی است.[۷]

۲-۵ کاربرد برت در پردازش زبان طبیعی

برت، به دلیل توانایی آن در ایجاد بازنمایی عمیق و غنی از زبان، به ابزاری محبوب برای وظایف پردازش زبان طبیعی تبدیل شده است. برت برای طیف گسترده‌ای از وظایف پردازش زبان طبیعی، از جمله تجزیه و تحلیل احساسات، پاسخ‌گویی به سؤال، شناسایی موجودیت نام‌گذاری شده، ترجمه ماشینی و طبقه‌بندی متن و غیره استفاده شده است. موفقیت آن در این وظایف به دلیل پیش‌آموزش آن بر روی مقادیر زیادی از داده‌های متنی است که به آن اجازه می‌دهد تا ظرافت‌های ظریف زبان و زمینه را به تصویر بکشد. علاوه بر این، توانایی برت برای تنظیم دقیق برای کارهای خاص، آن را به ابزاری همه کاره برای بسیاری از برنامه‌های پردازش زبان طبیعی تبدیل کرده است. برت با عملکرد پیشرفته و تطبیق پذیری خود، احتمالاً همچنان منبع ارزشمندی برای کارهای مرتبط با زبان در آینده خواهد بود.

در پایان، برت یا بازنمایی رمزگذار دوطرفه از ترنسفورمر، با ارائه ابزاری قدرتمند برای تولید بازنمایی‌های عمیق و غنی از زبان، حوزه پردازش زبان طبیعی را متحول کرده است. توانایی آن برای پیش آموزش بر روی مقادیر زیادی از داده‌های متنی، همراه با قابلیت تنظیم دقیق آن، آن را به ابزاری همه کاره برای طیف گسترده‌ای از وظایف پردازش زبان طبیعی تبدیل کرده است که در بسیاری از معیارها به عملکردی پیشرفته دست می‌یابد. برت به طور گسترده توسط محققان و متخصصان در زمینه پردازش زبان طبیعی

[5] ablation experiments

پذیرفته شده است و احتمالاً تأثیر آن در سال‌های آینده همچنان محسوس خواهد بود. برت با توانایی خود در به تصویر کشیدن ظرافت‌های زبان و زمینه، گام بزرگی به جلو در توانایی ما برای درک و تجزیه و تحلیل زبان انسانی است.

فصل سوم

روش‌های بهبود بازنمایی مدل برت

پیش‌آموزش کامل شبکه به یک سری پیشرفت ها در یادگیری بازنمایی زبان منجر شده است. شواهد حاصل از این پیشرفت ها نشان می دهد که یک شبکه بزرگ برای دستیابی به عملکرد پیشرفته از اهمیت حیاتی برخوردار است.بازنمایی‌های متنی ازپیش آموزش دیده را می‌توان با استفاده از روش‌های بدون نظارت مانند مدل‌سازی زبان یا روش‌های نظارت شده مانند ترجمه‌ی ماشینی بدست آورد.در این فصل ما به روش‌های بدون نظارت می‌پردازیم.

روش‌های مختلفی برای بهبود بازنمایی مدل برت ارایه شده است. مدل ارنی[1]به جای ماسک کردن تصادفی توکن‌ها از استراتژی ماسک کردن دانش، یعنی ماسک کردن در سطح موجودیت‌ها و عبارت‌ها استفاده می‌کند.پیشرفت کلیدی که ارنی نسبت به برت ارائه می‌کند، توانایی آن در ادغام منابع دانش خارجی، مانند نمودارهای دانش و ویکی‌پدیا، برای بهبود فرآیند پیش آموزش است. این روش باعث می‌شود نزدیکی معنایی و روابط گفتمانی در مدل بهتر یاد گرفته شود [۱۵].
مدل اسپن‌برت [2] بر اساس مدل ارنی ساخته شده است و آن را با افزودن یک استراتژی ماسک کردن جدید گسترش می‌دهد [۶]. اسپن‌برت از دو روش برای پیش آموزش استفاده می شود:

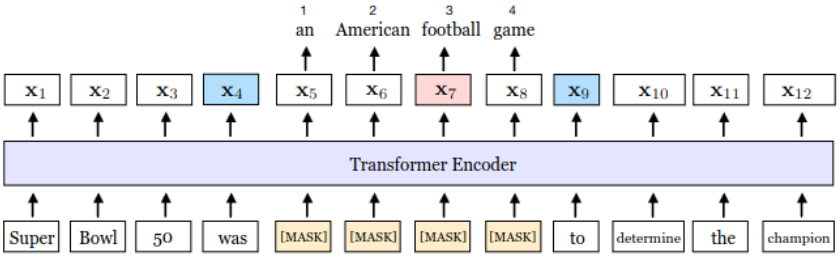
۱. هدف مرزی اسپن[3] که شامل پیش بینی مرزهای موجودیت در یک بخش از متن است. به طور خاص، این مدل برای پیش‌بینی موقعیت شروع و پایان همه موجودیت‌های ممکن در متن ورودی آموزش داده می‌شود. این کار برای کمک به مدل طراحی شده است که یاد بگیرد و با موجودیت های تودرتو یا همپوشانی که می تواند یک مشکل چالش برانگیز در پردازش زبان طبیعی باشد، بشناسد.

۲. مدل‌سازی زبان ماسک شده [4] مانند مدل‌های برت و ارنی، توکن های مجزا به طور تصادفی در طول پیش آموزش پنهان می‌شوند، اما در مدل‌سازی اسپن‌برت از چندین نشانه $[MASK]$ برای نمایش گستره های متوالی متن استفاده می کند. همانطور که در شکل ۳-۱مشاهده می‌شود، هنگام پوشاندن جمله ورودی، اسپن‌برت یک بازه تصادفی از نشانه های متوالی را از جمله انتخاب می کند و سپس همه نشانه های موجود در آن بازه را با نشانه های ویژه $[MASK]$ جایگزین می کند.

مدل برت ساختاری [5] با استفاده از ساختار سلسله مراتبی زبان توانایی مدل در درک ساختار جملات پیچیده را بهبود می‌بخشد. این مدل یک هدف ساختاری کلمه را پیشنهاد می کند که به طور تصادفی ترتیب تریگرم [6] را برای بازسازی و یک هدف ساختاری جمله را تغییر می دهد که ترتیب دو بخش متوالی

[1] ERNIE
[2] Span-Bert
[3] Span Boundary Objective(SBO)
[4] Masked Language Modeling(MLM)
[5] StructBERT
[6] trigram

ارجاع به بده تصاویر
به تصویر اشاره کن



شکل ۳-۱: تصویری از آموزش اسپن‌برت[۶]

دنباله‌ی $anAmericanfootballgame$ ماسک شده است.هدف مرزی اسپن از نمایش های خروجی تعبیه‌های مرزی،$x4$ و$x9$ (به رنگ آبی)، برای پیش بینی هر تعبیه ماسک شده استفاده می کند.[۶]

را پیش بینی می‌کند[؟]. مدل برت ساختاری دو لایه جدید را معرفی می کند که در سطوح مختلف سلسله مراتب نحوی عمل می کنند:

۱. لایه ی توجه به خود چند گرانروی[۷]: این لایه بر خلاف مکانیسم توجه به خود در برت، که بر روی یک دنباله از کلمات با طول ثابت عمل می کند، این در سطوح مختلف گرانروی، از جمله سطوح کلمه، عبارت، و جمله عمل می کند. این لایه به مدل اجازه می دهد تا ساختار سلسله مراتبی زبان را یاد بگیرید و روابط بین کلمات را در سطوح مختلف سلسله مراتب نحو رمزگذاری کند.

۲. لایه خودتوجهی چند سر با آگاهی نحوی[۸]: این لایه در سطح ساختار نحوی عمل می کند و از توجه به خود برای به تصویر کشیدن روابط بین کلماتی که با وابستگی های نحوی به هم مرتبط هستند استفاده می کند.به طور خاص، این لایه از اطلاعات تجزیه وابستگی و تجزیه سازنده برای شناسایی روابط نحوی بین کلمات و رمزگذاری این اطلاعات در جاسازی‌های مدل استفاده می‌کند.

انسجام متن

مدل روبرتا [۹] نسخه‌ی بهبود یافته‌ی مدل برت است که چند تغیر جهت بهبود مدل برت اعمال کرده است و به بهبودهای قابل توجهی در عملکرد دست یافته است [۱۱]. این تغیرات عبارتند از:

۱. روبرتا مدل را برای مدت طولانی تری با دسته های بزرگتر و داده های بیشتر آموزش داد. این بدان معنی است که مدل فرصت های بیشتری برای یادگیری داشته باشد.

۲. این مدل روی دنباله‌های طولانی‌تر آمورش می‌بیند.در برت، حداکثر طول توالی برای آموزش ۵۱۲ توکن بود. با این حال، در روبرتا، حداکثر طول دنباله برای تمرین نیز ۵۱۲ توکن بود، اما با طول دنباله در طول آموزش پویا است. این بدان معنی است که روبرتا می‌تواند توالی های طولانی‌تری

[7] Multi-Granularity Self-Attention

[8] syntactic-aware Multi-Head Self-Attention

[9] RoBerta

را در طول آموزش ببیند و طول هر دنباله می تواند در طول تمرین متفاوت باشد. این به روبرتا اجازه داد تا الگوها و وابستگی های پیچیده‌تری را در دنباله‌های متن طولانی‌تر یاد بگیرد.

۳. روبرتا به صورت پویا موقعیت‌های ماسک‌دار را در طول آموزش تغییر می‌دهد. در برت از موقعیت‌های ماسک‌دار یکسان برای همه نمونه‌های آموزشی استفاده می‌شود که منجر به حفظ موقعیت‌ها توسط مدل و عدم یادگیری تعمیم به موقعیت‌های جدید می‌شود. روبرتا با تغییر پویا موقعیت‌های ماسک‌دار تعمیم بهتری را می‌آموزد.

۴. روبرتا هدف پیش‌بینی جمله‌ی بعدی را حذف کرد،این هدف در برت برای کمک به مدل برای درک رابطه بین دو جمله استفاده می‌شود. نویسندگان این مقاله استدلال می‌کنند که ، پیش‌بینی جمله‌ی بعدی کمتر از آنچه در ابتدا تصور می شد مفید است، و حذف آن به روبرتا اجازه داد تا بر جنبه های دیگر درک زبان تمرکز کند.

برخلاف مدل روبرتا البرت استدلال می‌کند که هدف پیش‌بینی جمله‌ی بعدی فاقد مشکل است، زیرا نمونه‌های منفی با جفت کردن بخش‌هایی از اسناد مختلف ایجاد می‌شوند، این پیش‌بینی موضوع و پیش‌بینی انسجام را در یک کار واحد ترکیب می‌کند. آلبرت در عوض از یک هدف پیش بینی ترتیب جمله استفاده می کند. این هدف با برداشتن دو بخش متوالی و نمونه های منفی با معکوس کردن ترتیب دو بخش متوالی از یک سند، نمونه های مثبت را به دست می آورد[۸].

$XLNet$ یک مدل زبانی مشابه برت است، اما از یک رویکرد آموزشی متفاوت به نام مدل‌سازی زبان جایگشت برای بهبود عملکرد خود در وظایف مختلف پردازش زبان طبیعی استفاده می‌کند.این مدل روی دو نقطه ضعف مدل برت کار می‌کند[۱۹].

۱. برت با هر توکن ماسک شده به گونه‌ای رفتار می‌کند که گویی به طور تصادفی انتخاب شده است و توکن‌هایی که ماسک می‌شوند مستقل از یکدیگر هستند. این فرض می تواند مشکل ساز باشد، زیرا معنای یک جمله به بافت یا رابطه بین کلمات متعدد بستگی دارد و مستقل بودن هر کلمه می‌تواند منجر به خطا یا عدم دقت در پیش‌بینی مدل شود.

۲. نمادهایی مانند $[MASK]$ توسط برت در طول آموزش معرفی می‌شوند، اما هرگز در داده‌های واقعی رخ نمی‌دهند، و در نتیجه بین پیش آموزش و تنظیم دقیق پارامترها اختلاف وجود دارد.

برای رفع این مشکلات $XLNet$ یک روش رگرسیون خودکار جدید مبتنی بر مدل‌سازی زبان جایگشت[۱۶] پیشنهاد می‌کند که تابع هدف تخمین درست نمایی بیشینه برای آن به صورت زیر محاسبه می شود:

$$\max_{\theta} \mathbb{E}_{z \in Z_n} \sum_{j=1}^{n} \log p_\theta(t_{z,j} | t_{z,1}, t_{z,2}, ..., t_{z,j-1}) \qquad (۱-۳)$$

همان طور که در تصویر ۳-۲ مشاهده می‌شود، برای هر دنباله $XLNet$ یک جایگشت $[z_1, z_2, ...z_n]$ از همه‌ی مجموعه جایگشت‌های Z_n که در آن $|Z_N| = N!$ را نمونه گیری می‌کند. احتمال دنباله با توجه به Z فاکتورگیری می‌شود و توکن z_j ام مشروط به تمام نشانه‌های قبلی $t_{z,1}, t_{z,2}, ..., t_{z,j-1}$ با توجه به ترتیب جایگشت z است. مدل‌سازی زبان جایگشت با ایجاد همه جایگشت‌های ممکن یک جمله یا دنباله نشانه‌ها، و آموزش مدل برای پیش‌بینی ترتیب اصلی نشانه‌ها، بدون توجه به موقعیت آنها در دنباله، کار می‌کند. این به مدل اجازه می‌دهد تا روابط متنی بین نشانه‌ها را بیاموزد، حتی زمانی که برخی از آنها ماسک یا خراب شده باشند.

ارجاع به بده تصاویر

این مدل از مکانیزم توجه دوسویه و ترنسفورمر-ایکس‌ال [۱۰] استفاده می‌کند تا موقعیت های هدف z_j را در نظر بگیرد و وابستگی های دوربرد را یاد بگیرد. از آنجایی که کاردینالیته Z_N فاکتوریل است، بهینه سازی ساده و بی تکلف چالش برانگیز خواهد بود. بنابراین،$XLNet$بخشی از ورودی را شرط می کند و بقیه ورودی را تولید می کند تا مقیاس فضای جستجو را کاهش دهد:

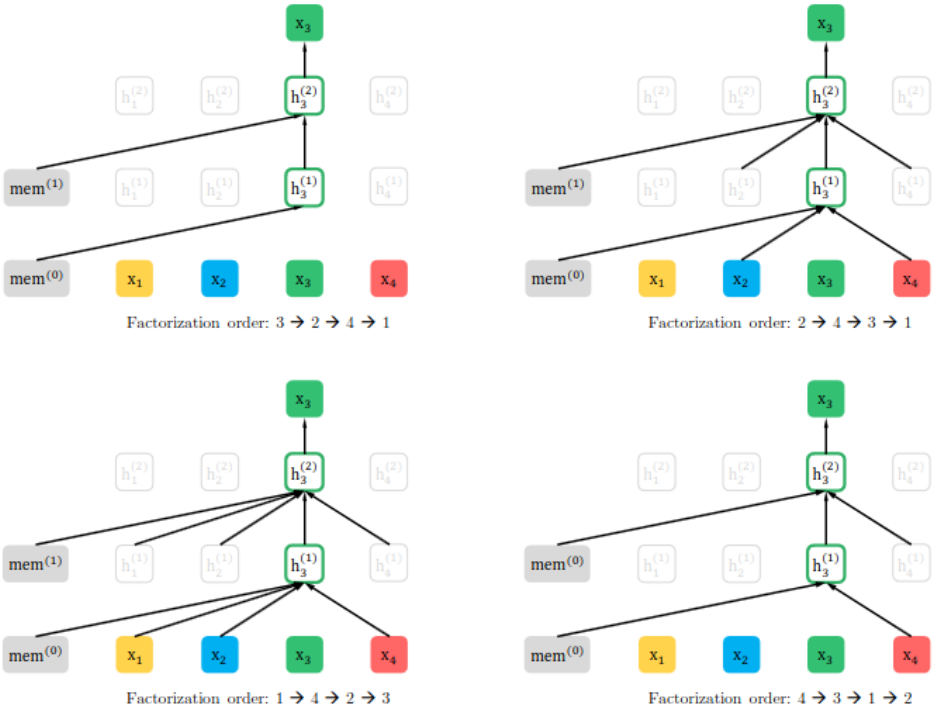
$$\max_{\theta} \mathbb{E}_{z \in Z_n} \sum_{j=c+1}^{n} \log p_\theta(t_{z,j} | t_{z,1}, t_{z,2}, ..., t_{z,j-1}) \qquad (۲-۳)$$

در اینجا c نقطه برش دنباله است.به طور کلی مقایسه مستقیم$XLNet$ با برت به دلیل تغییرات متعدد در تابع هزینه و معماری دشوار است.

الکترا یک روش پیش‌آموزش موثرتر از برت را پیشنهاد می کند. الکترا به جای خراب کردن برخی از موقعیت های ورودی با $[MASK]$، این مدل برخی از نشانه های ورودی را با جایگزین های قابل قبول آنها که از یک شبکه مولد کوچک نمونه برداری شده اند جایگزین می کند. الکترا یک تفکیک کننده را آموزش می دهد تا پیش بینی کند که آیا هر توکن در ورودی خراب توسط ژنراتور جایگزین شده است یا خیر. سپس از تمایزکننده ازپیش آموزش‌دیده می‌توان در کارهای پایین دستی برای تنظیم دقیق استفاده کرد، و بر اساس بازنمایی‌های از پیش آموزش دیده شده توسط مولد بهبود می‌یابد[۲].

مدل بارت تابع نویز اضافی فراتر از مدل‌سازی زبان ماسک شده را برای مدل‌های توالی به دنباله قبل از آموزش معرفی می‌کند[۹]. ابتدا، توالی ورودی با استفاده از یک تابع نویز دلخواه خراب می شود. سپس، ورودی خراب توسط یک شبکه ترنسفورمر که با استفاده از معلم اجباری آموزش دیده بازسازی می شود. بارت طیف گسترده ای از عملکردهای نویز را ارزیابی می کند، از جمله پوشاندن توکن، حذف توکن، پر کردن متن، چرخش سند، به هم ریختن جمله (به هم زدن تصادفی ترتیب کلمه یک جمله). بهترین عملکرد با استفاده از هم زدن جمله و پر کردن متن به دست می آید[۱۰].

[10] transformer-XL



شکل ۳-۲: مدل‌سازی زبان جایگشت $XLNet$ [۱۹]

تصویری از هدف مدل‌سازی زبان جایگشت برای پیش‌بینی $x3$ با توجه به دنباله ورودی یکسان x اما با ترتیب‌های فاکتورگیری متفاوت [۱۹]

در نتیجه، توسعه مدل‌های مختلف مبتنی بر معماری برت مانند البرت، روبرتا، ایکس‌ال‌نت و اسپن‌برت منجر به پیشرفت‌های قابل توجهی در بازنمایی و پیش آموزش مدل‌های زبانی شده است. این مدل‌ها عملکرد پیشرفته‌ای را در طیف گسترده‌ای از وظایف پردازش زبان طبیعی، از جمله پاسخ‌گویی به سؤال، شناسایی موجودیت نام‌گذاری شده، و طبقه‌بندی متن، و غیره نشان داده‌اند. این مدل‌ها قابلیت‌های مدل‌های زبانی را گسترش داده‌اند و پتانسیل پیش آموزش را برای بهبود عملکرد در طیف وسیعی از وظایف پردازش زبان طبیعی نشان داده‌اند. توسعه و اصلاح مداوم این مدل‌ها احتمالاً در سال‌های آینده باعث پیشرفت در زمینه پردازش زبان طبیعی خواهد شد.

فصل چهارم

بهبود ساختار و کارآمدی برت

مدل‌هایی بزرگتری که عملکرد بهتری دارند هزینه آموزش آن‌ها بیشتر است، مدل‌های کدگذار می‌توانند به طور موثر مقیاس شوند. بنابراین مدل‌های کارآمد ارایه شده منجر به بهبود مدل‌های ناکارآمد با همان اندازه می‌شوند. در این بخش، مروری بر چندین تلاش با هدف کاهش بودجه محاسباتی (مصرف زمان و حافظه) در طول آموزش و استنتاج مدل‌های مبتنی بر برت ارائه می‌کنیم[۱۸].

یک مانع برای پاسخ به این سوال محدودیت حافظه سخت افزار موجود است. با توجه به اینکه مدل‌های پیشرفته کنونی اغلب صدها میلیون یا حتی میلیاردها پارامتر دارند، وقتی سعی می‌کنیم مدل‌های خود را مقیاس‌بندی کنیم، به راحتی می‌توان به این محدودیت‌ها دست یافت. سرعت آموزش همچنین می‌تواند به طور قابل توجهی در آموزش توزیع شده با مشکل مواجه شود، زیرا سربار ارتباط مستقیماً با تعداد پارامترهای مدل متناسب است.

راه حل‌های موجود برای مشکلات فوق‌الذکر شامل موازی سازی مدل[۱۳] ، [۱۴]و مدیریت هوشمندانه حافظه [۱]، [۵] می‌باشد.این راه حل‌ها فقط مشکل محدودیت حافظه را حل می‌کنند، و برای مشکل سربار ارتباط کاری نمی‌کنند.

۱-۴ راهکار مدل البرت

مدل البرت که پارامترهای قابل توجهی کمتری نسبت به معماری برت سنتی دارد، به تمام مشکلات ذکر شده می‌پردازد. این مدل برای کاهش مصرف حافظه و افزایش سرعت آموزش از دو تکنیک برای کاهش تعداد پارامتر استفاده می‌کند.

۱. استفاده از تکنیک پارامترسازی فاکتوریزه شده که تعداد پارامترها را در لایه بازنمایی مدل با فاکتورسازی ماتریس بازنمایی به دو ماتریس کوچکتر کاهش می‌دهد و باعث کاهش مصرف حافظه و بهبود روش ذخیره بازنمایی‌هامی‌شود.

۲. اشتراک گذاری پارامترهای متقابل با در چندین لایه که به مدل اجازه می‌دهد کارآمدتر و موثرتر یاد بگیرد، زیرا می‌تواند از مجموعه پارامترهای مشابه در چندین لایه استفاده مجدد کند. این تکنیک همچنین به کاهش تعداد کلی پارامترها در مدل کمک می‌کند که به نوبه خود باعث کاهش مصرف حافظه و زمان مورد نیاز آموزش می‌شود.

برای بهبود بیشتر عملکرد البرت، یک تابع هزینه برای پیش‌بینی ترتیب جمله معرفی می‌کنیم. تابع هزینه پیش‌بینی ترتیب جمله اولیه[۱] بر انسجام بین جمله تمرکز دارد و برای رسیدگی به ناکارآمدی تابع هزینه‌ی پیش‌بینی جمله بعدی [۲]پیشنهاد شده در برت پایه طراحی شده است.

[1] sentence-order prediction (SOP) loss

[2] next sentence prediction(NSP) loss

در نتیجه این تصمیمات طراحی، می‌توان مدل‌های البرت بسیار بزرگ‌تری که پارامترهای کمتر وعملکرد بهتری نسبت به مدل بزرگ برت دارند پیاده سازی شوند.پایه‌ی معماری مدل البرت مانند برت می‌باشد و از همان لایه‌های مدل برت استفاده شده است[۸].

۴-۲ راهکار مدل بارت

مدل بارت[۳] یک مدل دنباله به دنباله است که می‌تواند در هر دو حالت خودبازگشتی و غیر خودبازگشتی آموزش داده شود. همانطور که در شکل ۴-۱ مشاهده می‌کنید این مدل شامل یک کدگذار و کدگشای ترنسفورمر است، بخش کدگذار مدل متن ورودی را می‌گیرد و بازنمایی با طول ثابت از ورودی تولید می‌کند که برای تولید متن خروجی به کدگشا وارد می‌شود. بارت را می‌توان با استفاده از اهداف رمزگذاری خودکار حذف نویز آموزش داد، به این معنی که برای بازسازی متن اصلی از نسخه خراب متن، مانند تعویض تصادفی کلمات در یک جمله، آموزش دیده است[۹]. معماری این مدل بر پایه ی مدل استاندارد ترنسفومر [۱۷] می‌باشد و مانند جی‌پی‌تی از تابع فعال‌سازی گلو[۴] به جای رلو[۵] استفاده می‌کند. معماری بارت ارتباط نزدیکی با معماری مورد استفاده در برت دارد، امابه طور کلی تعداد پارامترهای مدل بارت ۱۰٪ از برت بیشتر است. تفاوت‌های این دو مدل شامل موارد زیر می‌باشد:

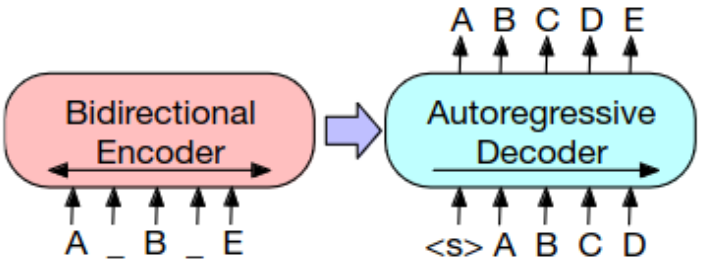
۱. هر لایه‌ی کدگشا علاوه بر اعمال مکانیزم خود توجه درون خودش، یک توجه متقابل بر روی لایه ی مخفی نهایی کدگذار انجام می‌دهند.

۲. برت از یک شبکه‌ی پیشخور[۶] اضافه قبل از پیش‌بینی کلمه استفاده می‌کند و این قسمت در مدل بارت حذف شده است.

۴-۳ راهکار مدل ایکس‌ال‌نت

ایکس‌ال‌نت[۷] یک مدل زبانی پیشرفته است که از مکانیسم دو جریانی خودتوجهی[۸] و ترنسفورمر-ایکس‌ال[۹] استفاده می‌کند تا وابستگی‌های دوربرد را یاد بگیرد و مدل برت را ارتقا بدهد و وابستگی‌های پیچیده‌تری را بین کلمات در یک جمله ثبت کند.یکی دیگر از مزیت‌های این مدل عدم نیاز به ماسک کردن کلمات

[3] BART (Bidirectional and Auto-Regressive Transformer)
[4] GeLU
[5] ReLU
[6] feedfoward
[7] XLNET
[8] wo-Stream Self-Attention mechanism
[9] transformer-XL



شکل ۴-۱: ساختار مدل بارت [۹]

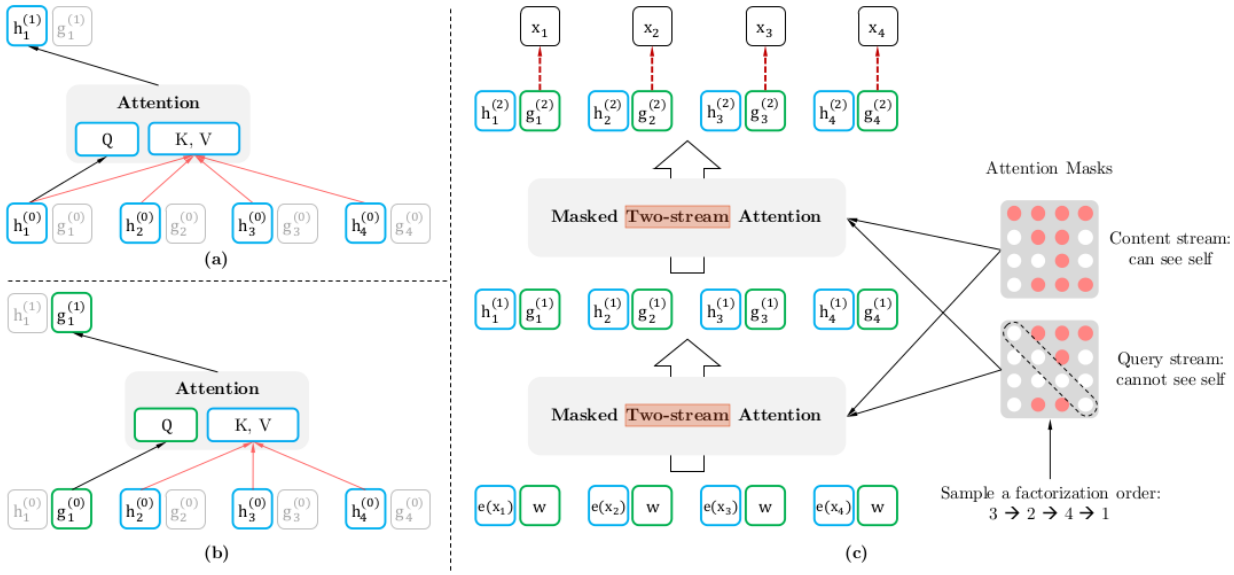
ورودی‌های کدگذار نیازی به همسویی با خروجی‌های کدگشا ندارند، که امکان تبدیل نویز دلخواه را فراهم می‌کند. در اینجا، یک سند با جایگزین کردن دهانه‌های متن با نمادهای ماسک خراب شده است. سند خراب (سمت چپ) با یک مدل دو طرفه کدگذاری می‌شود و سپس احتمال سند اصلی (سمت راست) با کدگشای خودبازگشتی محاسبه می‌شود. برای تنظیم دقیق، یک سند خراب به رمزگذار و رمزگشا وارد می‌شود و ما از نمایش‌هایی از حالت پنهان نهایی کدگشا استفاده می‌کنیم[۹].

در طول آموزش است. به طور کلی مدل ایکس‌ال‌نت از برت کندتر است و به منابع محاسباتی بیشتری نیاز دارد[۱۹].

مکانیسم دو جریانی خودتوجهی در ایکس‌ال‌نت به استفاده از دو جریان توجه مجزا برای مدل‌سازی دو نوع وابستگی موجود در زبان اشاره دارد: وابستگی‌های رو به جلو و عقب. به طور خاص، ایکس‌ال‌نت از دو مکانیسم توجه استفاده می‌کند که به طور موازی عمل می‌کنند: یک مکانیسم استاندارد ماسک‌دار توجه به خود، که به تمام نشانه‌های قبلی در دنباله توجه می‌کند، و یک مکانیسم جدید توجه بدون ماسک، که به تمام نشانه‌های آینده در دنباله توجه می‌کند. با ترکیب این دو مکانیسم توجه، ایکس‌ال‌نت می‌تواند وابستگی‌های رو به جلو و عقب را در دنباله ثبت کند، که به مدل اجازه می‌دهد تا روابط پیچیده‌تری را بین توکن‌ها مدل کند (شکل ۴-۲).

ترنسفورمر-ایکس‌ال جزء کلیدی ایکس‌ال‌نت است،این معماری یک نسخه اصلاح شده از ترنسفورمر پایه [۱۷] است و برای رفع محدودیت مواجه با وابستگی‌های طولانی مدت ارائه شده است. برای غلبه بر این محدودیت، ترنسفورمر ایکس‌ال مکانیزم بازگشتی را معرفی می‌کند که اجازه می‌دهد اطلاعات بین بخش‌های مختلف دنباله ورودی منتقل شود. به طور خاص، از تکنیکی به نام «تکرار در سطح بخش[۱۰]» استفاده می‌کند تا به مدل امکان استفاده‌ی مجدد از حالت‌های پنهان بخش‌های قبلی توالی ورودی را بدهد، که مدل را قادر می‌سازد وابستگی‌های طولانی مدت را یاد بگیرد.

[10] segment-level recurrence



شکل ۴-۲: [۱۹] مکانیزم دو جریانی خودتوجهی

قسمت a: مکانیزم توجه به خود استاندارد، قسمت b: توجه جریان پرس و جو[۱۱]، که اطلاعات دسترسی در مورد محتوای xzt را ندارد. (ج): مروری بر آموزش مدل‌سازی زبان جایگشت با توجه دو جریانی[۱۹] .

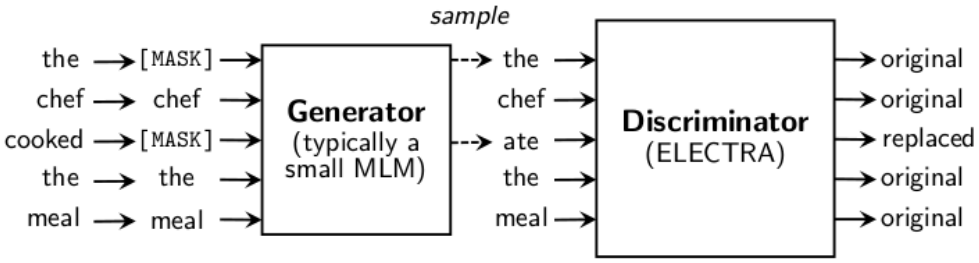
۴-۴ مدل الکترا

الکترا از معماری مشابه شبکه‌ی گن[۱۲] ولی به صورت غیر تقابلی استفاده می‌کند، که در آن یک مدل مولد برای ایجاد توکن‌های مصنوعی آموزش داده می‌شود، و یک مدل متمایزگر برای تمایز بین توکن‌های اصلی و مصنوعی آموزش داده می‌شود. سپس پارامترهای متمایزگر برای کارهای پایین دستی تنظیم دقیق می‌شود. این معماری که در شکل ؟؟ نشان داده شده است، برخلاف مدل برت است که بر پایه ی معماری کدگذار-کدگشا می‌باشد[۳].

مدل مولد یک مدل مبتنی بر ترانسفورماتور است که دنباله ای از توکن های ماسک شده را به عنوان ورودی می‌گیرد و دنباله ای از توکن های بدون ماسک مربوطه را خروجی می‌کند. هدف مولد یادگیری پیش‌بینی تعبیه‌های اصلی است که با ماسک جایگزین شده‌اند. تفکیک کننده همچنین یک مدل مبتنی بر ترانسفورماتور است که آموزش دیده است تا توکن های موجود در داده ها را از تعبیه‌هایی که با نمونه های مولد جایگزین شده‌اند، تشخیص دهد.

اگرچه روش آموزش شبیه شبکه‌ی گن است، اما چندین تفاوت کلیدی وجود دارد که موجب بهبود مدل می‌شود.اول، اگر مولد توکن درستی را تولید کند، آن توکن به جای «جعلی» «واقعی» در نظر گرفته می‌شود. مهم‌تر از آن، مولد با حداکثر احتمال آموزش داده می‌شود تا اینکه به صورت خصمانه برای فریب دادن تمایزکننده آموزش داده شود. اما آموزش تقابلی چالش برانگیز است زیرا پس انتشار خطا از طریق نمونه برداری از مولد غیرممکن است[۳].

[12] GAN



شکل ۴-۳: نمای کلی از تشخیص توکن جایگزین شده در مدل الکترا[۳]

شکل ۲: نمای کلی از تشخیص توکن جایگزین شده. مولد می‌تواند هر مدلی باشد که توزیع خروجی را روی توکن‌ها تولید می‌کند، اما ما معمولاً از یک مدل زبان ماسک‌دار کوچک استفاده می‌کنیم که به طور مشترک با تمایزکننده آموزش داده می‌شود.اگرچه این مدل ها مانند یک گن ساختاربندی شده‌اند، اما به دلیل دشواری به کارگیری گن در متن،به جای رویکرد تقابلی، یک مولد را با احتمال حداکثری آموزش داده می‌شود.پس از آموزش اولیه، مولد را بیرون می‌اندازیم و تنها متمایزگر را در کارهای پایین دست تنظیم می‌کنیم[۳].

توسعه مدل‌های مختلف مبتنی بر معماری برت، مانند بارت و الکترا عصر جدیدی از نوآوری در پردازش زبان طبیعی را آغاز کرده است. این مدل‌ها تکنیک‌ها و روش‌های جدیدی بهبودهای معماری را معرفی کرده‌اند که منجر به دستاوردهای عملکردی قابل توجهی در طیف وسیعی از وظایف پردازش زبان طبیعی شده‌اند. توسعه و اصلاح مداوم این مدل‌ها احتمالاً همچنان مرزهای آنچه را که در پردازش زبان طبیعی ممکن است پیش می‌برد و در نهایت منجر به مدل‌های زبانی پیشرفته‌تر و توانمندتر می‌شود که می‌توانند زبان انسانی را بهتر درک و تولید کنند.

فصل پنجم

نتایج

جدول ۵-۱: مقایسه‌ی

مجموعه داده	ماشین پشتیبان	k- نزدیک ترین همسایه	بیز ساده لوحانه	درخت تصمیم	LSTM	ماتع	کامپیوتر عصبی متمایز	ماتع + الگوریتم ازدحام ذرات
MNIST	۹۴/۱۶	۹۶/۹	۵۶/۱۵	۶۵/۴۰	۹۶/۴۸	۹۶/۹۶	۹۹/۱۲	۹۹/۷۳
ORL	۸۴/۶۸	۹۲/۵	۷۷/۲۵	۸۸/۵	۹۴/۲	۹۵/۱۱	۹۷/۲۱	۹۷/۹
Leter	۸۴/۱۱	۸۹/۸۱	۹۳/۲۱	۸۳/۶	۹۵/۵	۹۶/۰۱	۹۸/۱۶	۹۹/۰۲
Ionosphere	۸۰/۳	۷۷/۷۸	۸۲/۶۲	۸۴/۵	۹۱/۱۶	۹۳/۴۱	۹۶/۰۲	۹۷/۱

$SQuAD2.0$	$SQuAD1.1$				
$Model$ $DevF1$	$DevEM$ $TestEM$	$DevF1$ $TestF1$	$TestEM$	$TestF1$	$DevEM$
برت	۶.۸۴	۸.۹۱	۳.۸۳	۵.۹۰	۵.۷۶
۶.۸۳	۷.۷۵	۳.۸۲			
روبرتا	۳.۸۹	۱.۹۵	۴.۸۶	۵.۹۲	۸.۸۱
۹.۸۸	۸.۸۰	۱.۸۷			
ایکس‌ال‌نت	۰.۸۶	۸.۹۲	۶.۸۳	۲.۹۰	۶.۷۹
۴.۸۶	۷.۷۸	۵.۸۵			
البرت	۷.۸۸	۹.۹۴	۸.۸۵	۲.۹۲	۴.۸۰
۰.۸۷	۶.۷۹	۰.۸۶			
اسپن‌برت	۱.۹۱	۷.۹۵	۹.۸۸	۸.۹۳	۱.۸۴
۶.۸۹	۱.۸۳	۶.۸۸			
برت‌ساختاری	۲.۹۲	۰.۹۶	۰.۸۹	۱.۹۴	۴.۸۵
۰.۹۱	۳.۸۴	۷.۸۹			
الکترا	۸.۹۰	۳.۹۵	۶.۸۷	۹.۹۲	۸.۸۲
۴.۸۸	۷.۸۱	۳.۸۷			

جدول ۵-۲: □□□□□□□□□□□ □□ □□□□□□ □□ □□□□□۱.۱ □□□□□□ □□□ ۲.۰□□□□□□

□□□□□□۱.۱

□□□□□□۲.۰

□□□□□□ □□□ □□ □□□ □۱□ □□□□ □□ □□□□ □۱□ □□□ □□ □□□ □□□ □۱□ □□□□ □□ □□□□ □۱□

□□□□□	۶.۸۴	۸.۹۱	۳.۸۳	۵.۹۰	۵.۷۶	۶.۸۳	۷.۷۵	۳.۸۲	□□□□□□□□	۳.۸۹	۱.۹۵	۴.۸۶	۵.۹۲	۸.۸۱
	۹.۹۴	۷.۸۸	□□□□□□□	۵.۸۵	۷.۷۸	۴.۸۶	۶.۷۹	۲.۹۰	۶.۸۳	۸.۹۲	۰.۸۶	□□□□□□	۱.۸۷	۸.۸۰	۹.۸۸
	۶.۸۸	۱.۸۳	۶.۸۹	۱.۸۴	۸.۹۳	۹.۸۸	۷.۹۵	۱.۹۱	□□□□□□□□□	۰.۸۶	۶.۷۹	۰.۸۷	۴.۸۰	۲.۹۲	۸.۸۵
□□□□□□□□□□□	۹.۹۲	۶.۸۷	۳.۹۵	۸.۹۰	□□□□□□□□	۷.۸۹	۳.۸۴	۰.۹۱	۴.۸۵	۱.۹۴	۰.۸۹	۰.۹۶	۲.۹۲	□□□□□□□□□□□	
				۳.۸۷	۷.۸۱	۴.۸۸	۸.۸۲								

جدول ۵-۳: □□□□□□□□□□ □□ □□□□□□ □□ ۱.۱□□□□□ □□□ ۲.۰□□□□□

$SQuAD2.0$		$SQuAD1.1$		
$F1$	EM	$F1$	EM	$Model$
۱.۸۴	۷.۷۷	۹.۹۰	۱.۸۴	برت
۲.۸۸	۳.۸۱	۰.۹۵	۹.۸۸	روبرتا
۷.۸۷	۹.۸۰	۲.۹۲	۸.۸۵	ایکس‌ال‌نت
۸.۸۶	۲.۸۰	۶.۹۳	۹.۸۶	البرت

جدول ۵-۴: □□□□□□□□□□ □□ □□□□□□ □□ ۱.۱□□□□□ □□□ ۲.۰□□□□□

فصل ششم

جمع‌بندی

این گزارش نشان می‌دهد که مدل‌های مبتنی برت، مانند البرت، روبرتا، ایکس‌ال‌نت، و اسپن‌برت، پتانسیل پیش آموزش را در تقویت وظایف پردازش زبان طبیعی مانند طبقه‌بندی متن، شناسایی موجودیت‌های نام‌گذاری شده و پاسخ‌گویی به سؤال ثابت کرده‌اند.علاوه بر این،بارت و الکترا تکنیک‌های نوآورانه و پیشرفت‌های معماری را معرفی کرده‌اند که منجر به دستاوردهای عملکردی قابل توجهی در طیف گسترده‌ای از وظایف مدل‌سازی زبان شده‌اند.

توسعه و اصلاح مستمر این مدل‌ها احتمالا به جلو راندن مرزه ای ممکن در مدل سازی زبان ادامه خواهد داد. با پیشرفت‌ها و پیشرفت‌های مداوم، می‌توان انتظار داشت که مدل‌های زبانی پیشرفته‌تر و توانمندتری را ببینیم که می‌توانند زبان انسانی را بهتر درک و تولید کنند.

در پایان، این گزارش بر اهمیت تحقیق و توسعه پایدار در پردازش زبان طبیعی برای آزاد کردن پتانسیل کامل مدل‌های زبانی و کاربردهای آنها در حوزه‌های مختلف تاکید می‌کند. یافته‌های این گزارش نشان می‌دهد که پیش آموزش و بهبود بازنمایی رویکردهای موثری برای افزایش عملکرد وظایف پردازش زبان طبیعی هستند. انتظار می‌رود که توسعه و اصلاح مداوم مدل‌های مبتنی بر برت و جانشینان آن‌ها باعث پیشرفت بیشتر در زمینه پردازش زبان طبیعی در سال‌های آینده شود.

منابع و مراجع

[1] Chen, Tianqi, Xu, Bing, Zhang, Chiyuan, and Guestrin, Carlos. Training deep nets with sublinear memory cost, 2016.

[2] Clark, Kevin, Luong, Minh-Thang, Le, Quoc V., and Manning, Christopher D. Electra: Pre-training text encoders as discriminators rather than generators. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020.

[3] Clark, Kevin, Luong, Thang, Le, Quoc V., and Manning, Christopher. Electra: Pre-training text encoders as discriminators rather than generators. In ICLR, 2020.

[4] Dai, Andrew M and Le, Quoc V. Semi-supervised sequence learning. In Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc.

[5] Gomez, Aidan N., Ren, Mengye, Urtasun, Raquel, and Grosse, Roger B. The reversible residual network: Backpropagation without storing activations, 2017.

[6] Joshi, Mandar, Chen, Danqi, Liu, Yinhan, Weld, Daniel S., Zettlemoyer, Luke, and Levy, Omer. Spanbert: Improving pre-training by representing and predicting spans. Trans. Assoc. Comput. Linguistics, 8:64–77, 2020.

[7] Kenton, Jacob Devlin Ming-Wei Chang and Toutanova, Lee Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT, pages 4171–4186, 2019.

[8] Lan, Zhenzhong, Chen, Mingda, Goodman, Sebastian, Gimpel, Kevin, Sharma, Piyush, and Soricut, Radu. Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations, 2020.

[9] Lewis, Mike, Liu, Yinhan, Goyal, Naman, Ghazvininejad, Marjan, Mohamed, Abdelrahman, Levy, Omer, Stoyanov, Veselin, and Zettlemoyer, Luke. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[10] Liu, Qi, Kusner, Matt J, and Blunsom, Phil. A survey on contextual embeddings. arXiv preprint arXiv:2003.07278, 2020.

[11] Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, and Stoyanov, Veselin. Roberta: A robustly optimized bert pretraining approach. 2019.

[12] Peters, Matthew E., Neumann, Mark, Iyyer, Mohit, Gardner, Matt, Clark, Christopher, Lee, Kenton, and Zettlemoyer, Luke. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[13] Shazeer, Noam, Cheng, Youlong, Parmar, Niki, Tran, Dustin, Vaswani, Ashish, Koanantakool, Penporn, Hawkins, Peter, Lee, HyoukJoong, Hong, Mingsheng, Young, Cliff, et al. Mesh-tensorflow: Deep learning for supercomputers. Advances in neural information processing systems, 31, 2018.

[14] Shoeybi, Mohammad, Patwary, M, Puri, R, LeGresley, P, Casper, J, Catanzaro, B Megatron-LM, et al. Training multi-billion parameter language models using model parallelism. arXiv preprint cs.CL/1909.08053, 2019.

[15] Sun, Yu, Wang, Shuohuan, Li, Yukun, Feng, Shikun, Chen, Xuyi, Zhang, Han, Tian, Xin, Zhu, Danxiang, Tian, Hao, and Wu, Hua. ERNIE: Enhanced representation through knowledge integration. number arXiv:1904.09223. arXiv.

[16] Uria, Benigno, Côté, Marc-Alexandre, Gregor, Karol, Murray, Iain, and Larochelle, Hugo. Neural autoregressive distribution estimation. The Journal of Machine Learning Research, 17(1):7184–7220, 2016.

[17] Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Lukasz, and Polosukhin, Illia. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[18] Xia, Patrick, Wu, Shijie, and Van Durme, Benjamin. Which* bert? a survey organizing contextualized encoders. arXiv preprint arXiv:2010.00854, 2020.

[19] Yang, Zhilin, Dai, Zihang, Yang, Yiming, Carbonell, Jaime G., Salakhutdinov, Ruslan, and Le, Quoc V. Xlnet: Generalized autoregressive pretraining for language understanding. In NeurIPS, pages 5754–5764, 2019.

واژه‌نامه‌ی فارسی به انگلیسی

آ

اسپن‌برت . . . . . spanbert

ب

برت ساختاری . . . . . structbert

پ

پارامترسازی . . . . . parameterization

پردازش زبان طبیعی . . natural language processing (nlp)

پس انتشار . . . . . back propagate

پیش آموزش . . . . . pretraining

پیشخور . . . . . Feed-forward

پیکره‌بندی . . . . . configuration

ت

تخمین درست نمایی بیشینه Maximum likelihood estimation (MLE)

تریگرم . . . . . trig-ram

تعبیه . . . . . Embedding

تنظیم دقیق پارامترها . . . . . Fine-tune

ج

جایگشت . . . . . Permutation

چ

چند گرانروی . . . . . Multi-Granularity

ح

حافظه‌ی کوتاه مدت ماندگار . . . . . long short-term memory(LSTM)

خ

خودبازگشتی . . . . . auto-regressive

د

دوسویه . . . . . representation

ش

شرطی‌سازی مشترک . . . . . jointly conditioning

غ

غیرخودبازگشتی . non-auto-regressive

ف

فاکتوریزه شده ..... Factorized

ک

کاردینالیته ..... cardinality

کدگذار ..... encoder

کدگشا ..... decoder

کورپوس ..... Corpus

م

ماسک کردن ..... masking

متمایزگر ..... discriminator

محک ..... Benchmark

مدل زبانی ماسک شده ..... Masked Language Model (MLM)

موجودیت ..... entity

مولد ..... generator

واژه‌نامه‌ی انگلیسی به فارسی

A

Auto-regressive . . . . . خودبازگشتی

B

Back propagate . . . . . پس انتشار

Benchmark . . . . . محک

Bidirectional . . . . . دوسویه

C

Cardinality . . . . . کاردینالیته

Configuration . . . . . پیکره‌بندی

Corpus . . . . . کورپوس

D

Decoder . . . . . کدگشا

Discriminator . . . . . متمایزگر

E

Embedding . . . . . تعبیه

Encoder . . . . . کدگذار

Entity . . . . . موجودیت

F

Factorized . . . . . فاکتوریزه شده

Feed-forward . . . . . پیشخور

Fine-tune . . . . . تنظیم دقیق پارامترها

G

Generator . . . . . مولد

J

Jointly . . . . . شرطی‌سازی مشترک conditioning

L

long . . . . . حافظه‌ی کوتاه مدت ماندگار short-term memory(LSTM)

M

Masked . . . . . مدل زبانی ماسک شده language model (mlm)

Masking . . . . . ماسک کردن

Maximum likelihood estimation (mle) تخمین درست نمایی بیشینه

Multi-granularity . . . . . چند گرانروی

N

Natural language processing (nlp) . پردازش زبان طبیعی

Non-auto-regressive . غیرخودبازگشتی

P

Parameterization . . . . . پارامترسازی

Permutation . . . . . جایگشت

Pretraining . . . . . پیش آموزش

R

Representation . . . . . دوسویه

S

Spanbert . . . . . اسپن‌برت

Structbert . . . . . برت ساختاری

T

Trig-ram . . . . . تریگرم