for Transformers Bidirectional Deep of Pre-training :**BERT** Understanding Language

Toutanova Kristina Lee Kenton Chang Ming-Wei Devlin Jacob
Language AI Google
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

ما یک مدل جدید برای نمایش زبانی معرفی می‌کنیم با نام **BERT**، که مخفف Bidirectional En-coder Representations from Transformers است. برخلاف مدل‌های اخیر نمایش زبان (؟؟)، BERT به‌گونه‌ای طراحی شده است که نمایش‌های دوجهته‌ی عمیقی را از متون بدون برچسب، با شرط‌بندی هم‌زمان بر بافت‌های سمت چپ و راست در تمام لایه‌ها، پیش‌آموزش دهد. در نتیجه، مدل ازپیش‌آموزش‌داده‌شده‌ی BERT می‌تواند تنها با افزودن یک لایه‌ی خروجی اضافی برای طیف گسترده‌ای از وظایف، نظیر پاسخ‌گویی به پرسش و استنتاج زبانی، به مدل‌های پیشرفته‌ی روز تبدیل شود، بی‌آن‌که نیاز به تغییرات عمده در معماری خاص هر وظیفه باشد.

BERT از نظر مفهومی ساده و از نظر تجربی قدرتمند است. این مدل در یازده وظیفه‌ی پردازش زبان طبیعی (NLP) نتایج جدیدی در سطح بهترین روش‌های موجود به دست آورده است، از جمله ارتقای امتیاز GLUE به ٪۸۰٫۵ (افزایش مطلق ۷٫۷ واحد درصد)، افزایش دقت MultiNLI به ٪۸۶٫۷ (افزایش مطلق ۴٫۶ واحد درصد)، افزایش امتیاز SQuAD v1.1 در معیار Test F1 به ۹۳٫۲ (افزایش مطلق ۱٫۵ واحد درصد) و همچنین افزایش امتیاز SQuAD v2.0 Test F1 به ۸۳٫۱ (افزایش مطلق ۵٫۱ واحد درصد).

Introduction ۱

پیش‌آموزش مدل زبان برای بهبود بسیاری از وظایف پردازش زبان طبیعی (؟؟؟؟). مؤثر نشان داده شده است. این وظایف شامل وظایف سطح جمله مانند استنتاج زبان طبیعی (؟؟) و بازنویسی (؟)،است که هدف آنها پیش‌بینی روابط بین جملات با تجزیه و تحلیل جامع آنها است، و همچنین وظایف سطح توکن مانند تشخیص موجودیت‌های نامگذاری شده و پاسخ به سوال، که در آنها مدل‌ها برای تولید خروجی دقیق در سطح توکن مورد نیاز هستند (؟؟).

دو راهکار موجود برای به‌کارگیری نمایش‌های زبانی ازپیش‌آموزش‌دیده در وظایف پایین‌دستی وجود دارد: *مبتنی بر ویژگی* و *تنظیم دقیق* (*fine-tuning*). رویکرد مبتنی بر ویژگی، مانند مدل ELMo (؟)، از معماری‌های خاص هر وظیفه استفاده می‌کند که نمایش‌های ازپیش‌آموزش‌دیده را به‌عنوان ویژگی‌های اضافی در نظر می‌گیرند. در مقابل، رویکرد تنظیم دقیق، مانند مدل Generative Pre-trained Transformer (OpenAI GPT) (؟)، تنها تعداد کمی پارامتر وابسته به وظیفه را معرفی می‌کند و با تنظیم تمام پارامترهای ازپیش‌آموزش‌دیده، روی وظایف پایین‌دستی آموزش داده می‌شود. هر دو رویکرد در مرحله‌ی پیش‌آموزش از تابع هدف یکسانی استفاده می‌کنند، جایی که مدل‌های زبانی یک‌جهته برای یادگیری نمایش‌های کلی زبان به‌کار می‌روند.

ما استدلال می‌کنیم که روش‌های فعلی توانایی نمایش‌های ازپیش‌آموزش‌دیده را به‌ویژه در رویکردهای تنظیم دقیق محدود می‌کنند. محدودیت اصلی آن است که مدل‌های زبانی استاندارد یک‌جهته هستند و این امر دامنه‌ی معماری‌هایی را که می‌توان در مرحله‌ی پیش‌آموزش به‌کار برد، کاهش می‌دهد. برای نمونه، در مدل OpenAI GPT، نویسندگان از معماری چپ‌به‌راست استفاده کرده‌اند که در آن هر توکن تنها می‌تواند به توکن‌های پیشین خود در لایه‌های خودتوجهی Transformer (؟) توجه کند. چنین محدودیتی برای وظایف در سطح جمله بهینه نیست و می‌تواند هنگام استفاده از رویکردهای مبتنی بر تنظیم دقیق در وظایفی مانند پاسخ‌گویی به پرسش، که در آن‌ها ترکیب بافت از هر دو جهت ضروری است، اثر منفی شدیدی داشته باشد.

در این مقاله، ما با معرفی BERT ، روش‌های مبتنی بر تنظیم دقیق را بهبود می‌دهیم. BERT که مخفف Bidirectional Encoder Representations from Transformers است، محدودیت ذکرشده‌ی یک‌جهته‌بودن را با بهره‌گیری از هدف پیش‌آموزش به‌نام masked lan-guage model (MLM) رفع می‌کند؛ هدفی که از آزمون Cloze (؟) الهام گرفته است. در مدل زبانی پوشیده، برخی از توکن‌های ورودی به‌صورت تصادفی ماسک می‌شوند و هدف، پیش‌بینی شناسه‌ی واژگانی اصلیِ توکنِ ماسک‌شده تنها بر اساس بافت آن است. برخلاف پیش‌آموزش مدل‌های زبانی چپ‌به‌راست، هدف MLM امکان ترکیب بافت چپ و راست را فراهم می‌کند، که در نتیجه می‌توان یک Transformer دوبخشی (دوجهته) عمیق را پیش‌آموزش داد. علاوه بر مدل زبانی پوشیده، ما از وظیفه‌ی دیگری به‌نام next sentence prediction نیز استفاده می‌کنیم که به‌صورت هم‌زمان نمایش‌های جفت‌جمله را پیش‌آموزش می‌دهد.

- ما اهمیت پیش‌آموزش دوجهته را برای نمایش‌های زبانی نشان می‌دهیم. برخلاف ؟ که از مدل‌های زبانی

یک‌جهته برای پیش‌آموزش استفاده می‌کند، BERT با بهره‌گیری از مدل‌های زبانی پوشیده (Masked Lan-guage Models) امکان ایجاد نمایش‌های دوجهته و عمیق از پیش‌آموزش‌دیده را فراهم می‌سازد. این رویکرد همچنین در تضاد با روش ؟ است که از ترکیب سطحی دو مدل زبانی چپ‌به‌راست و راست‌به‌چپِ آموزش‌داده‌شده به‌صورت مستقل استفاده می‌کند.

- ما نشان می‌دهیم که نمایش‌های ازپیش‌آموزش‌دیده نیاز به بسیاری از معماری‌های پیچیده و خاصِ هر وظیفه را کاهش می‌دهند. BERT نخستین مدل نمایش مبتنی بر تنظیم دقیق است که در مجموعه‌ی گسترده‌ای از وظایف در سطح جمله و سطح توکن به عملکرد در حد بهترین روش‌های روز دست می‌یابد و از بسیاری از معماری‌های خاصِ وظیفه پیشی می‌گیرد.
- BERT در یازده وظیفه‌ی پردازش زبان طبیعی (NLP) پیشرفت قابل‌توجهی نسبت به روش‌های پیشین ارائه می‌دهد. کد منبع و مدل‌های ازپیش‌آموزش‌داده‌شده در https://github.com/google-research/bert در دسترس هستند.

۲ Work Related

تاریخچه‌ای طولانی در زمینه‌ی پیش‌آموزش نمایش‌های زبانی عمومی وجود دارد. در این بخش، به‌طور خلاصه به بررسی رایج‌ترین رویکردهای مورد استفاده در این حوزه می‌پردازیم.

۱.۲ رویکردهای بدون‌نظارت مبتنی بر ویژگی

یادگیری نمایش‌هایی از واژه‌ها که بتوانند در دامنه‌های گوناگون به‌کار گرفته شوند، طی دهه‌ها یکی از حوزه‌های فعال پژوهش بوده است. این تلاش‌ها شامل روش‌های غیرعصبی (؟؟؟) و همچنین روش‌های مبتنی بر شبکه‌های عصبی (؟؟) می‌شوند. تعبیه‌های واژه‌ای ازپیش‌آموزش‌دیده، بخش جدایی‌ناپذیر سامانه‌های پردازش زبان طبیعی مدرن محسوب می‌شوند و نسبت به تعبیه‌هایی که از ابتدا آموزش داده می‌شوند، بهبود قابل‌توجهی ارائه می‌دهند (؟). برای پیش‌آموزش بردارهای تعبیه‌ی واژه، اهداف مدل‌سازی زبان از چپ به راست (؟) و نیز اهدافی که به تمایز واژه‌های درست از نادرست در بافت چپ و راست کمک می‌کنند (؟) به‌کار رفته‌اند.

این رویکردها به سطوح درشت‌دانه‌تری نیز تعمیم یافته‌اند، از جمله تعبیه‌های جمله‌ای (؟؟) یا تعبیه‌های بند (؟). برای آموزش نمایش‌های جمله، پژوهش‌های پیشین از اهدافی مانند رتبه‌بندی جمله‌های بعدیِ کاندید (؟؟)، تولید واژه‌های جمله‌ی بعدی از چپ به راست با توجه به نمایش جمله‌ی پیشین (؟)، یا اهداف برگرفته از خودرمزگذارهای حذف نویز (؟) بهره گرفته‌اند.

مدل ELMo و نسخه‌های پیشین آن (؟؟) پژوهش‌های سنتی تعبیه‌ی واژه را در بُعدی متفاوت گسترش می‌دهند. این مدل‌ها ویژگی‌های *وابسته به بافت* را از دو مدل زبانیِ چپ‌به‌راست و راست‌به‌چپ استخراج می‌کنند. نمایش بافتی هر توکن از به‌هم‌پیوستنِ نمایش‌های چپ‌به‌راست و راست‌به‌چپ به‌دست می‌آید. هنگامی‌که تعبیه‌های بافتی واژه در معماری‌های خاصِ وظیفه ادغام می‌شوند، ELMo عملکرد به‌روز و پیشرفته‌ای را در چندین معیار مهم NLP از جمله پاسخ‌گویی به پرسش‌ها (؟)، تحلیل احساسات (؟)، و شناسایی نام موجودیت‌ها (؟) ارائه می‌دهد (؟).

؟ روشی را برای یادگیری نمایش‌های بافتی پیشنهاد کردند که در آن مدل، یک واژه را بر اساس بافت چپ و راست آن با استفاده از LSTM پیش‌بینی می‌کند. مشابه ELMo، مدل آن‌ها نیز مبتنی بر ویژگی بوده و به‌طور عمیق دوسویه نیست. همچنین، ؟ نشان دادند که وظیفه‌ی cloze (پیش‌بینی واژه‌های حذف‌شده) می‌تواند موجب افزایش پایداری مدل‌های تولید متن شود.

۲.۲ رویکردهای تنظیم دقیق بدون‌ناظر

مشابه رویکردهای مبتنی بر ویژگی، نخستین پژوهش‌ها در این زمینه تنها پارامترهای تعبیه‌ی واژه را از متون بدون برچسب پیش‌آموزش می‌دادند (؟).

در سال‌های اخیر، رمزگذارهای جمله یا سند که نمایش‌های بافتیِ از توکن‌ها تولید می‌کنند، از داده‌های بدون برچسب پیش‌آموزش یافته و سپس برای وظایف پایین‌دستیِ نظارت‌شده تنظیم دقیق شده‌اند (؟؟؟). مزیت اصلی این رویکردها آن است که تنها تعداد اندکی از پارامترها باید از ابتدا یاد گرفته شوند. تا حدی به دلیل همین مزیت، مدل OpenAI GPT (؟) توانست در بسیاری از وظایف سطح جمله از مجموعه‌داده‌ی معیار GLUE (؟) به نتایج برتر پیشین دست یابد. برای پیش‌آموزش این مدل‌ها، از اهداف مدل‌سازی زبانی چپ‌به‌راست و خودرمزگذار (auto-encoder) استفاده شده است (؟؟؟).

۳.۲ یادگیری انتقالی از داده‌های نظارت‌شده

پژوهش‌هایی نیز انجام شده که نشان می‌دهند انتقال دانش از وظایف نظارت‌شده با مجموعه‌داده‌های بزرگ — مانند استنتاج زبان طبیعی (؟) و ترجمه‌ی ماشینی (؟) — می‌تواند مؤثر باشد. در حوزه‌ی بینایی رایانه‌ای نیز پژوهش‌ها اهمیت یادگیری انتقالی از مدل‌های بزرگ ازپیش‌آموزش‌دیده را نشان داده‌اند، به‌طوری‌که یکی از راهکارهای مؤثر، تنظیم دقیق مدل‌هایی است که با مجموعه‌داده‌ی ImageNet (؟؟) پیش‌آموزش یافته‌اند.

۳ مدل BERT

در این بخش، مدل BERT و جزئیات پیاده‌سازی آن را معرفی می‌کنیم. چارچوب ما شامل دو مرحله است: *پیش‌آموزش* و *تنظیم دقیق*. در مرحله‌ی پیش‌آموزش، مدل بر روی داده‌های بدون برچسب و در قالب چندین وظیفه‌ی پیش‌آموزش مختلف آموزش می‌بیند. در مرحله‌ی تنظیم دقیق، مدل BERT ابتدا با پارامترهای ازپیش‌آموزش‌یافته مقداردهی اولیه می‌شود و سپس تمامی پارامترها با استفاده از داده‌های برچسب‌دارِ وظیفه‌ی پایین‌دستی تنظیم دقیق می‌شوند. هر وظیفه‌ی پایین‌دستی، مدل تنظیم‌شده‌ی مخصوص به خود را دارد، هرچند همه‌ی

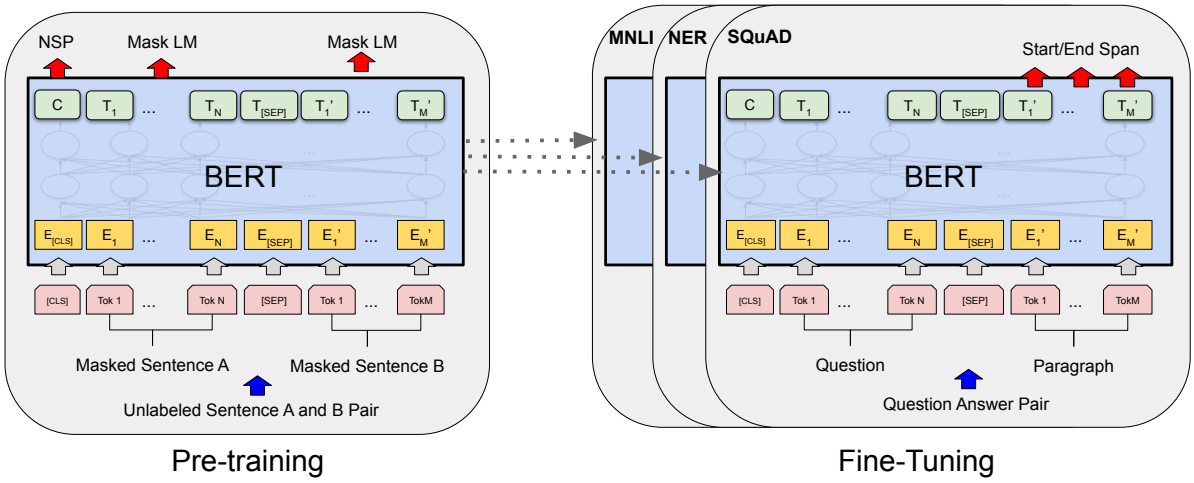


Figure ۱: فرآیند کلی پیش‌آموزش و تنظیم دقیق (fine-tuning) در مدل BERT. به‌جز لایه‌های خروجی، در هر دو مرحله‌ی پیش‌آموزش و تنظیم دقیق از معماری‌های یکسانی استفاده می‌شود. پارامترهای ازپیش‌آموزش‌دیده برای مقداردهی اولیه‌ی مدل‌ها در وظایف پایین‌دستی مختلف به‌کار می‌روند. در طی تنظیم دقیق، تمام پارامترها به‌روزرسانی می‌شوند. نماد [CLS] یک نشانگر ویژه است که در ابتدای هر نمونه ورودی افزوده می‌شود، و [SEP] یک توکن جداساز ویژه است (برای مثال، جهت جداسازی پرسش‌ها و پاسخ‌ها).

آن‌ها با پارامترهای پیش‌آموزش‌یافته‌ی یکسان مقداردهی اولیه می‌شوند. مثال پرسش‌وپاسخ در شکل ۱ به عنوان نمونه‌ی جاری در این بخش به کار گرفته می‌شود.

یکی از ویژگی‌های متمایز BERT ، معماری یکپارچه‌ی آن در میان وظایف مختلف است. تفاوت میان معماری پیش‌آموزش و معماری نهایی وظایف پایین‌دستی بسیار اندک است.

معماری مدل معماری مدل BERT یک رمزگذار Transformer دوسویه‌ی چندلایه است که بر اساس پیاده‌سازی اصلی معرفی‌شده در ؟ ساخته شده و در کتابخانه‌ی tensor2tensor[۱] منتشر شده است. از آن‌جا که استفاده از Transformerها امروزه رایج شده و پیاده‌سازی ما تقریباً مشابه نسخه‌ی اصلی است، از ارائه‌ی توضیحات پیش‌زمینه‌ای مفصل خودداری می‌کنیم و خوانندگان را به منبع اصلی ؟ و همچنین منابع آموزشی عالی نظیر «The Annotated Transformer»[۲] ارجاع می‌دهیم.

در این پژوهش، تعداد لایه‌ها (یعنی بلوک‌های Transformer) را با L، اندازه‌ی نهان را با H و تعداد سرهای خودتوجهی را با A نمایش می‌دهیم.[۳] نتایج اصلی ما بر روی دو اندازه‌ی مدل گزارش می‌شوند: $\text{BERT}_{\text{BASE}}$ (L=12، H=768، A=12 و در مجموع ۱۱۰ میلیون پارامتر) و $\text{BERT}_{\text{LARGE}}$ (L=24، H=1024، A=16 و در مجموع ۳۴۰ میلیون پارامتر).

مدل $\text{BERT}_{\text{BASE}}$ به‌منظور مقایسه، با همان اندازه‌ی مدل OpenAI GPT انتخاب شد. با این حال، تفاوت اساسی آن است که BERT از خودتوجهی دوسویه بهره می‌برد، در حالی که GPT از خودتوجهی محدود استفاده می‌کند که در آن هر توکن تنها می‌تواند به بافت سمت چپ خود توجه کند.[۴]

نمایش‌های ورودی/خروجی برای اینکه BERT بتواند طیف گسترده‌ای از وظایف پایین‌دستی را انجام دهد، نمایش ورودی ما قادر است به‌طور غیرمبهم هم یک جمله‌ی منفرد و هم یک جفت جمله (مثلاً ⟨ پرسش، پاسخ ⟩) را در یک توالی توکن نمایش دهد. در سرتاسر این پژوهش، «جمله» می‌تواند هر بازه‌ی دلخواهی از متن متوالی باشد و لزوماً یک جمله‌ی زبان‌شناختی واقعی نیست. «توالی» به توالی ورودی توکن‌ها به BERT گفته می‌شود، که می‌تواند یک جمله یا دو جمله‌ی بسته‌بندی‌شده کنار هم باشد.

ما از WordPiece embeddings (؟) با یک واژه‌نامه ۳۰,۰۰۰ توکنی استفاده می‌کنیم. توکن اول هر توالی همیشه یک توکن ویژه‌ی طبقه‌بندی ([CLS]) است. وضعیت نهان نهایی متناظر با این توکن به عنوان نمایش تجمیعی توالی برای وظایف طبقه‌بندی به کار می‌رود. جفت جملات در یک توالی واحد بسته‌بندی می‌شوند. ما جملات را به دو روش متمایز می‌کنیم. اول، آن‌ها را با یک توکن ویژه ([SEP]) جدا می‌کنیم. دوم، به هر توکن یک نمایش یادگرفته‌شده اضافه می‌کنیم که مشخص می‌کند آیا توکن متعلق به جمله‌ی A است یا جمله‌ی B. همان‌طور که در شکل ۱ نشان داده شده، نمایش ورودی را با E، بردار نهان نهایی توکن ویژه‌ی [CLS] را با $C \in \mathbb{R}^H$ و بردار نهان نهایی توکن ورودی i^{th} را با $T_i \in \mathbb{R}^H$ نمایش می‌دهیم.

---

[۱] https://github.com/tensorflow/tensor2tensor

[۲] http://nlp.seas.harvard.edu/2018/04/03/attention.html

[۳] در تمامی موارد، اندازه‌ی لایه‌ی پیش‌خور یا فیلتر را برابر $4H$ در نظر می‌گیریم؛ یعنی ۳۰۷۲ برای H=768 و ۴۰۹۶ برای H=1024.

[۴] در ادبیات پژوهشی، معمولاً به Transformer دوسویه «رمزگذار Transformer» و به نسخه‌ی محدود به بافت چپ «رمزگشای Transformer» گفته می‌شود، چرا که نسخه‌ی دوم برای تولید متن به کار می‌رود.

برای هر توکن، نمایش ورودی آن با جمع کردن نمایش‌های توکن، بخش و موقعیت متناظر ساخته می‌شود. یک تصویرسازی از این ساختار در شکل ۲ دیده می‌شود.

۱.۳ پیش‌آموزش BERT

بر خلاف ؟ و ؟، ما از مدل‌های زبان سنتی چپ‌ـبه‌ـراست یا راست‌ـبه‌ـچپ برای پیش‌آموزش BERT استفاده نمی‌کنیم. در عوض، BERT را با استفاده از دو وظیفه‌ی بدون نظارت پیش‌آموزش می‌کنیم که در این بخش شرح داده شده‌اند. این مرحله در بخش چپ شکل ۱ نشان داده شده است.

وظیفه ۱#: مدل زبان ماسک‌شده (Masked LM) به‌طور شهودی، منطقی است که باور کنیم یک مدل عمیق دوجهتی به‌طور قابل توجهی قدرتمندتر از یک مدل چپ‌ـبه‌ـراست یا ترکیب سطحی مدل‌های چپ‌ـبه‌ـراست و راست‌ـبه‌ـچپ است. متأسفانه، مدل‌های زبان شرطی استاندارد تنها می‌توانند چپ‌ـبه‌ـراست *یا* راست‌ـبه‌ـچپ آموزش ببینند، زیرا شرطی‌سازی دوجهتی باعث می‌شود هر واژه به‌طور غیرمستقیم «خودش را ببیند» و مدل به‌سادگی بتواند واژه هدف را در یک زمینه‌ی چندلایه‌ای پیش‌بینی کند.

برای آموزش یک نمایش عمیق دوجهتی، ما به سادگی درصدی از توکن‌های ورودی را به‌طور تصادفی ماسک می‌کنیم و سپس آن توکن‌های ماسک‌شده را پیش‌بینی می‌کنیم. ما به این فرآیند «مدل زبان ماسک‌شده» (MLM) می‌گوییم، هرچند در منابع ادبیات غالباً به آن وظیفه‌ی *Cloze* گفته می‌شود (؟). در این حالت، بردارهای نهان نهایی متناظر با توکن‌های ماسک‌شده به یک خروجی softmax روی واژگان داده می‌شوند، مشابه یک مدل زبان استاندارد. در تمام آزمایش‌های ما، ۱۵% از تمام توکن‌های WordPiece هر توالی به‌طور تصادفی ماسک می‌شوند. بر خلاف denois-ing auto-encoders (؟)، ما تنها واژه‌های ماسک‌شده را پیش‌بینی می‌کنیم و کل ورودی را بازسازی نمی‌کنیم.

اگرچه این کار به ما امکان ایجاد یک مدل پیش‌آموزش دوجهتی را می‌دهد، اما یک نکته منفی دارد: ایجاد ناسازگاری بین پیش‌آموزش و ریزتنظیم، زیرا توکن [MASK] در هنگام ریزتنظیم ظاهر نمی‌شود. برای کاهش این مشکل، ما همیشه واژه‌های «ماسک‌شده» را با توکن واقعی [MASK] جایگزین نمی‌کنیم. تولیدکننده داده‌های آموزشی به‌طور تصادفی ۱۵% از موقعیت‌های توکن را برای پیش‌بینی انتخاب می‌کند. اگر توکن iـام انتخاب شد، ما توکن iـام را به یکی از موارد زیر جایگزین می‌کنیم: (۱) توکن [MASK] در ۸۰% مواقع، (۲) یک توکن تصادفی در ۱۰% مواقع، (۳) توکن iـام بدون تغییر در ۱۰% مواقع. سپس، T_i برای پیش‌بینی توکن اصلی با استفاده از cross entropy loss به کار می‌رود. ما انواع مختلف این روش را در ضمیمه پ.۲ مقایسه کرده‌ایم.

وظیفه ۲#: پیش‌بینی جمله بعدی (NSP) بسیاری از وظایف پایین‌دستی مهم مانند پرسش و پاسخ (QA) و استنتاج زبان طبیعی (NLI) بر اساس فهم *رابطه* بین دو جمله هستند که به‌طور مستقیم توسط مدل‌سازی زبان ثبت نمی‌شود. برای آموزش مدلی که روابط بین جملات را درک کند، ما پیش‌آموزش را برای وظیفه‌ی دودویی «پیش‌بینی جمله بعدی» انجام می‌دهیم که می‌توان آن را به‌سادگی از هر مجموعه‌ی متنی تک‌زبانه ایجاد کرد. به طور مشخص، هنگام انتخاب جملات A و B برای هر نمونه‌ی پیش‌آموزش، ۵۰% مواقع B جمله واقعی بعد از A است (برچسب‌گذاری شده به‌عنوان IsNext) و ۵۰% مواقع جمله‌ای تصادفی از مجموعه انتخاب می‌شود (برچسب‌گذاری شده به‌عنوان NotNext). همان‌طور که در شکل ۱ نشان داده شده، بردار C برای پیش‌بینی جمله بعدی (NSP) به کار می‌رود.[۵] با وجود سادگی، نشان می‌دهیم در بخش ۱.۵ که پیش‌آموزش برای این وظیفه برای QA و NLI بسیار مفید است. [۶] وظیفه NSP ارتباط نزدیکی با اهداف یادگیری نمایش در ؟ و ؟ دارد. با این حال، در کارهای پیشین، تنها نمایش‌های جمله به وظایف پایین‌دستی منتقل می‌شوند، در حالی که BERT تمام پارامترها را برای مقداردهی اولیه‌ی مدل وظیفه انتهایی انتقال می‌دهد.

داده‌های پیش‌آموزش روند پیش‌آموزش عمدتاً از ادبیات موجود درباره پیش‌آموزش مدل زبان پیروی می‌کند. برای مجموعه داده‌های پیش‌آموزش از BooksCorpus (۸۰۰ میلیون واژه) (؟) و ویکی‌پدیای انگلیسی (۲,۵۰۰ میلیون واژه) استفاده می‌کنیم. برای ویکی‌پدیا تنها بخش‌های متنی استخراج شده و لیست‌ها، جداول و سرفصل‌ها نادیده گرفته می‌شوند. استفاده از یک مجموعه داده سطح سند به جای مجموعه داده سطح جمله‌ی مرتب‌شده مانند Billion Word Benchmark (؟) برای استخراج توالی‌های طولانی متوالی حیاتی است.

۲.۳ BERT Fine-tuning

Fine-tuning ساده است، زیرا مکانیزم خودـتوجه (self-attention) در معماری Transformer به BERT امکان می‌دهد تا بسیاری از وظایف پایین‌دستی—چه شامل یک متن باشند و چه جفت متن—را با جایگزینی ورودی‌ها و خروجی‌های مناسب مدل کند. برای کاربردهایی که شامل جفت متن هستند، یک الگوی رایج این است که ابتدا جفت متن‌ها را به‌صورت مستقل کدگذاری کرده و سپس از توجه متقابل دوطرفه (bidirectional cross attention) استفاده کنند، مانند ؟؟. در مقابل، BERT از مکانیزم خودـتوجه برای یکپارچه‌سازی این دو مرحله استفاده می‌کند؛ زیرا کدگذاری یک جفت متن الحاق‌شده (concatenated) با خودـتوجه، به‌طور مؤثری شامل توجه متقابل *دوطرفه* بین دو جمله می‌شود.

برای هر وظیفه، ما به‌سادگی ورودی‌ها و خروجی‌های مربوط به آن وظیفه را به BERT متصل کرده و تمام پارامترها را به‌صورت fine-tune end-to-end می‌کنیم. در ورودی، جملهٔ A و جملهٔ B از مرحلهٔ پیش‌آموزش (pre-training) معادل موارد زیر هستند: (۱) جفت جملات در بازنویسی معنایی، (paraphrasing) (۲) جفت فرضیه—

[۵] مدل نهایی به دقت ۹۷%-۹۸% در NSP دست می‌یابد.

[۶] بردار C بدون ریزتنظیم، نمایش جمله معنی‌داری نیست، زیرا با NSP آموزش داده شده است.

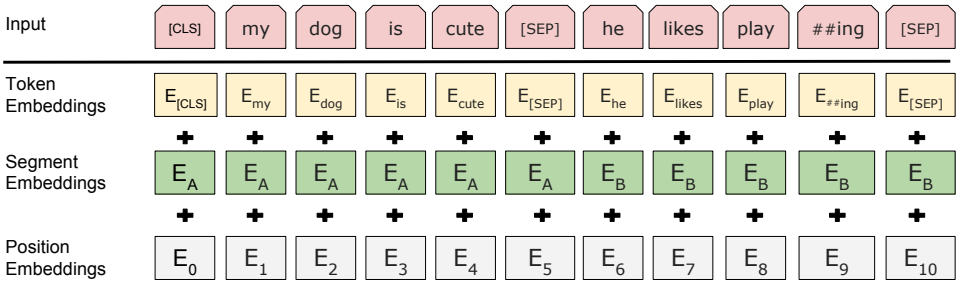


Figure ۲: نمایش ورودی BERT. نمایش‌های ورودی مجموع نمایش‌های توکن، نمایش‌های بخش‌بندی و نمایش‌های موقعیت هستند.

پیش‌فرض (hypothesis—premise) در استنتاج (entail-ment)، (۳) جفت پرسش—متن (question—passage) در پاسخ‌دهی به پرسش (question answering)، و (۴) یک جفت متن—تهگون (degenerate) در طبقه‌بندی متن یا برچسب‌گذاری دنباله (sequence tagging). در خروجی، نمایش‌های توکن‌ها به یک لایهٔ خروجی برای وظایف سطح‌ـتوکن (token-level)، مانند برچسب‌گذاری دنباله یا پاسخ‌دهی به پرسش، ارسال می‌شوند؛ و نمایش [CLS] به یک لایهٔ خروجی برای طبقه‌بندی، مانند استنتاج یا تحلیل احساسات (sentiment analysis)، فرستاده می‌شود.

در مقایسه با پیش‌آموزش، fine-tuning هزینهٔ نسبتاً کمی دارد. تمام نتایج ارائه‌شده در این مقاله را می‌توان حداکثر در یک ساعت با یک واحد پردازشی ابری (Cloud TPU) یا چند ساعت با یک GPU، با شروع از همان مدل پیش‌آموزش‌دیده، بازتولید (replicate) کرد.[۷] جزئیات مربوط به هر وظیفه را در زیربخش‌های مربوطه در بخش ۴ توصیف می‌کنیم. برای جزئیات بیشتر می‌توانید به پیوست آ.۵ مراجعه کنید.

۴ experiments

در این بخش، نتایج fine-tuning مدل BERT را روی ۱۱ وظیفهٔ پردازش زبان طبیعی ارائه می‌دهیم.

۱.۴ GLUE

معیار ارزیابی عمومی درک زبان (GLUE) (؟) مجموعه‌ای از وظایف متنوع درک زبان طبیعی است. توضیحات دقیق‌تر مجموعه‌داده‌های GLUE در پیوست ب.۱ آورده شده است.

برای ریزتنظیم روی GLUE، دنباله ورودی (برای یک جمله یا جفت جمله) را همان‌طور که در بخش ۳ توضیح داده شد، نمایش می‌دهیم و از بردار پنهان نهایی $C \in \mathbb{R}^H$ مربوط به اولین توکن ورودی ([CLS]) به عنوان نمایش کلی استفاده می‌کنیم. تنها پارامترهای جدیدی که در طول ریزتنظیم معرفی می‌شوند، وزن‌های لایه دسته‌بندی $W \in \mathbb{R}^{K \times H}$ هستند، که در آن K تعداد برچسب‌ها است. ما از بردار C و وزن W برای محاسبه یک خطای دسته‌بندی استاندارد استفاده می‌کنیم، یعنی $\log(\text{softmax}(CW^T))$.

برای همه وظایف GLUE، از اندازه بسته ۳۲ استفاده کرده و مدل را به مدت ۳ اپوک روی داده‌ها ریزتنظیم می‌کنیم. برای هر وظیفه، بهترین نرخ یادگیری ریزتنظیم را (از میان $5e-5$، $4e-5$، $3e-5$، و $2e-5$) روی مجموعه Dev انتخاب کردیم. علاوه بر این، برای BERT$_{\text{LARGE}}$ متوجه شدیم که ریزتنظیم گاهی روی داده‌های کوچک ناپایدار است، بنابراین چندین راه‌اندازی تصادفی انجام دادیم و بهترین مدل را روی مجموعه Dev انتخاب کردیم. در راه‌اندازی‌های تصادفی، از همان نقطه بررسی پیش‌آموزش شده استفاده می‌کنیم، اما جابه‌جایی داده‌های ریزتنظیم و مقداردهی اولیه لایه دسته‌بندی متفاوت انجام می‌دهیم.[۹]

نتایج در جدول ۱ ارائه شده است. هم BERT$_{\text{BASE}}$ و هم BERT$_{\text{LARGE}}$ تمام سیستم‌ها را در تمام وظایف با فاصله قابل توجهی پشت سر گذاشته‌اند و به ترتیب ۴.۵% و ۰.۷% بهبود دقت متوسط نسبت به وضعیت قبلی ارائه می‌دهند. توجه داشته باشید که BERT$_{\text{BASE}}$ و GPT OpenAI از نظر معماری مدل تقریباً یکسان هستند، به جز ماسک‌گذاری توجه. برای بزرگ‌ترین و پرگزارش‌ترین وظیفه GLUE، یعنی BERT MNLI، به بهبود ۴.۶% در دقت مطلق دست یافته است. در جدول رده‌بندی رسمی GLUE[۱۰]، BERT$_{\text{LARGE}}$ امتیاز ۵.۸۰ را کسب کرده است، در حالی که GPT OpenAI در زمان نگارش این مقاله امتیاز ۸.۷۲ را دارد.

۲.۴ SQuAD v۱.۱

مجموعه داده Stanford Question Answering Dataset (SQuAD v۱.۱) شامل ۱۰۰ هزار جفت پرسش/پاسخ جمع‌آوری شده از طریق جمع‌سپاری است (؟). در این وظیفه، با داشتن یک پرسش و یک متن از ویکی‌پدیا که شامل پاسخ است، هدف پیش‌بینی بازه متنی پاسخ در متن داده شده می‌باشد.

همان‌طور که در شکل ۱ نشان داده شده است، در وظیفه پاسخ به پرسش، سوال و پاراگراف ورودی را به‌صورت یک توالی بسته‌بندی شده نمایش می‌دهیم، به‌طوری که سوال از A embedding و پاراگراف از B embedding استفاده

[۷] به عنوان مثال، مدل BERT برای مجموعهٔ SQuAD را می‌توان در حدود ۳۰ دقیقه با یک TPU Cloud آموزش داد تا به نمرهٔ F۱ برابر با ۹۱.۰٪ در مجموعهٔ توسعه (Dev) دست یابد.

[۸] See (۱۰) in https://gluebenchmark.com/faq.

[۹] توزیع مجموعه داده‌های GLUE شامل برچسب‌های Test نمی‌شود و ما تنها یک بار ارسال ارزیابی به سرور GLUE برای هر یک از BERT$_{\text{BASE}}$ و BERT$_{\text{LARGE}}$ انجام دادیم.

[۱۰] https://gluebenchmark.com/leaderboard

Average	RTE	MRPC	STS-B	CoLA	SST-۲	QNLI	QQP	MNLI-(m/mm)	System
-	۵k.۲	۵k.۳	vk.۵	۵k.۸	۶۷k	۱۰۸k	۳۶۳k	۳۹۲k	
۰.۷۴	۷.۶۱	۰.۸۶	۰.۸۱	۰.۳۵	۲.۹۳	۳.۸۲	۱.۶۶	۱.۶/۸۰.۸۰	SOTA Pre-OpenAI
۰.۷۱	۸.۵۶	۹.۸۴	۳.۷۳	۰.۳۶	۴.۹۰	۸.۷۹	۸.۶۴	۱.۴/۷۶.۷۶	BiLSTM+ELMo+Attn
۱.۷۵	۰.۵۶	۳.۸۲	۰.۸۰	۴.۴۵	۳.۹۱	۴.۸۷	۳.۷۰	۴.۱/۸۱.۸۲	GPT OpenAI
۶.۷۹	۴.۶۶	۹.۸۸	۸.۸۵	۱.۵۲	۵.۹۳	۵.۹۰	۲.۷۱	۴.۶/۸۳.۸۴	BASEBERT
۱.۸۲	**۱.۷۰**	**۳.۸۹**	**۵.۸۶**	**۵.۶۰**	**۹.۹۴**	**۷.۹۲**	**۱.۷۲**	**۹.۷/۸۵.۸۶**	LARGEBERT

The .(https://gluebenchmark.com/leaderboard) server evaluation the by scored results، Test GLUE :۱ Table different slightly is column “Average” The examples. training of number the denotes task each below number are GPT OpenAI and BERT ^set. WNLI problematic the exclude we since score، GLUE official the than reported are correlations Spearman MRPC، and QQP for reported are scores F۱ task. single single-model، of one as BERT use that entries exclude We tasks. other the for reported are scores accuracy and STS-B، for components. their

می‌کند. تنها در هنگام fine-tuning یک بردار شروع $S \in \mathbb{R}^H$ و یک بردار پایان $E \in \mathbb{R}^H$ معرفی می‌کنیم. احتمال اینکه واژه i شروع بازه پاسخ باشد، به‌صورت حاصلضرب داخلی بین T_i و S محاسبه شده و سپس یک softmax روی تمام واژه‌های پاراگراف اعمال می‌شود:

$$P_i = \frac{e^{S \cdot T_i}}{\sum_j e^{S \cdot T_j}}.$$

فرمول مشابه برای انتهای بازه پاسخ استفاده می‌شود. امتیاز یک بازه پیشنهادی از موقعیت i تا موقعیت j به‌صورت

$$S \cdot T_i + E \cdot T_j$$

تعریف شده است و بازه‌ای با بیشترین امتیاز که $j \geq i$ باشد به‌عنوان پیش‌بینی انتخاب می‌شود. هدف آموزش، مجموع لگاریتم احتمال موقعیت‌های شروع و پایان صحیح است. ما برای ۳ epoch با نرخ یادگیری $5e-5$ و اندازه batch برابر با ۳۲، fine-tuning انجام می‌دهیم.

جدول ۲ نتایج برتر در leaderboard و همچنین نتایج سیستم‌های منتشرشده را نشان می‌دهد (؟؟؟؟). نتایج برتر SQuAD leaderboard توصیف سیستم‌های عمومی به‌روز را ندارند[۱۱] و می‌توانند از هر داده عمومی هنگام آموزش استفاده کنند. بنابراین ما در سیستم خود از افزایش داده متواضعانه استفاده می‌کنیم؛ ابتدا روی TriviaQA (؟) fine-tuning انجام می‌دهیم و سپس روی SQuAD ادامه می‌دهیم.

بهترین سیستم ما از سیستم برتر leaderboard با +۵.۱ F۱ در ترکیب مدل‌ها و +۳.۱ F۱ به‌عنوان یک سیستم واحد پیشی می‌گیرد. در واقع، مدل BERT تنها ما، سیستم ترکیبی برتر را از نظر Score F۱ پشت سر می‌گذارد. بدون داده‌های TriviaQA، fine-tuning تنها ۱.۰-۴.۰ F۱ کاهش داریم و هنوز هم از تمام سیستم‌های موجود به‌طور قابل توجهی پیشی می‌گیریم[۱۲].

[۱۱] QANet در ؟ توصیف شده است، اما سیستم پس از انتشار به‌طور قابل توجهی بهبود یافته است.

[۱۲] داده‌های TriviaQA که استفاده کردیم شامل پاراگراف‌هایی از

v۲.۰ SQuAD ۳.۴

وظیفه SQuAD ۰.۲ تعریف مسئله SQuAD ۱.۱ را گسترش می‌دهد و اجازه می‌دهد که هیچ پاسخ کوتاهی در پاراگراف ارائه‌شده وجود نداشته باشد، که مسئله را واقع‌گرایانه‌تر می‌کند.

ما از رویکرد ساده‌ای برای گسترش مدل BERT نسخه v۱.۱ SQuAD برای این وظیفه استفاده می‌کنیم. سوال‌هایی که پاسخی ندارند، به‌عنوان سوال‌هایی در نظر گرفته می‌شوند که بازه پاسخ آن‌ها شروع و پایان آن روی توکن [CLS] است. فضای احتمالات برای موقعیت‌های شروع و پایان بازه پاسخ گسترش می‌یابد تا شامل موقعیت توکن [CLS] شود. برای پیش‌بینی، امتیاز بازه بدون پاسخ را با

$$s_{\texttt{null}} = S \cdot C + E \cdot C$$

با امتیاز بهترین بازه غیر تهی

$$\hat{s_{i,j}} = \max_{j \geq i} S \cdot T_i + E \cdot T_j$$

مقایسه می‌کنیم. یک پاسخ غیر تهی پیش‌بینی می‌شود اگر

$$\hat{s_{i,j}} > s_{\texttt{null}} + \tau$$

که آستانه τ روی مجموعه Dev انتخاب می‌شود تا F۱ به حداکثر برسد. ما برای این مدل از داده‌های TriviaQA استفاده نکردیم. Fine-tuning برای ۲ epoch با نرخ یادگیری $5e-5$ و اندازه batch برابر با ۴۸ انجام شد.

نتایج نسبت به ورودی‌های قبلی leaderboard و کارهای منتشرشده برتر (؟؟) در جدول ۳ نشان داده شده‌اند، با این تفاوت که سیستم‌هایی که BERT را به‌عنوان یکی از مؤلفه‌های خود استفاده کرده‌اند، حذف شده‌اند. ما بهبود +۱.۵ F۱ نسبت به سیستم برتر قبلی مشاهده می‌کنیم.

SWAG ۴.۴

داده‌مجموعه Genera- Adversarial With Situations tions (SWAG) شامل ۱۱۳ هزار مثال تکمیل جمله ـ جفتی

TriviaQA-Wiki است که از ۴۰۰ توکن اول اسناد تشکیل شده‌اند و حداقل یکی از پاسخ‌های ممکن ارائه‌شده در آن‌ها موجود است.

Test		Dev		سامانه (System)
F۱	EM	F۱	EM	
برترین سامانه‌های جدول امتیازات (۱۰ دسامبر ۲۰۱۸)				
۹۱/۲	۸۲/۳	–	–	انسان (Human)
۹۱/۷	۸۶/۰	–	–	#۱ تجمیعی (Ensemble) - nlnet
۹۰/۵	۸۴/۵	–	–	#۲ تجمیعی (Ensemble) - QANet
منتشرشده‌ها (Published)				
۸۵/۸	–	۸۵/۶	–	(تکی) BiDAF+ELMo
۸۸/۵	۸۲/۳	۸۷/۹	۸۱/۲	(تجمیعی) Reader R.M.
روش ما (Ours)				
–	–	۸۸/۵	۸۰/۸	(تکی) BERT$_{\text{BASE}}$
–	–	۹۰/۹	۸۴/۱	(تکی) BERT$_{\text{LARGE}}$
–	–	۹۱/۸	۸۵/۸	(تجمیعی) BERT$_{\text{LARGE}}$
۹۱/۸	**۸۵/۱**	**۹۱/۱**	**۸۴/۲**	(تکی + TriviaQA) BERT$_{\text{LARGE}}$
۹۳/۲	**۸۷/۴**	**۹۲/۲**	**۸۶/۲**	(تجمیعی + TriviaQA) BERT$_{\text{LARGE}}$

Table ۲: نتایج مجموعه‌داده SQuAD 1.1. مدل تجمیعی BERT شامل ۷ سامانه است که از نقاط بررسی (Checkpoints) پیش‌تمرین متفاوت و بذرهای تنظیم متفاوت در ریزتنظیم استفاده می‌کنند.

Test		Dev		سامانه (System)
F۱	EM	F۱	EM	
برترین سامانه‌های جدول امتیازات (۱۰ دسامبر ۲۰۱۸)				
۸۹/۵	۸۶/۹	۸۹/۰	۸۶/۳	انسان (Human)
۷۸/۰	۷۴/۸	–	–	#۱ تکی - (F-Net) MIR-MRC
۷۷/۱	۷۴/۲	–	–	#۲ تکی - nlnet
منتشرشده‌ها (Published)				
۷۴/۹	۷۱/۴	–	–	(تجمیعی) unet
۷۴/۴	۷۱/۴	–	–	(تکی) SLQA+
روش ما (Ours)				
۸۳/۱	۸۰/۰	۸۱/۹	۷۸/۷	(تکی) BERT$_{\text{LARGE}}$

Table ۳: نتایج مجموعه‌داده SQuAD 2.0. در این جدول، سامانه‌هایی که از BERT به‌عنوان یکی از اجزای خود استفاده کرده‌اند، لحاظ نشده‌اند.

است که استدلال مبتنی بر common-sense را ارزیابی می‌کند (؟). برای هر جمله، وظیفه این است که محتمل‌ترین ادامه را از میان چهار گزینه انتخاب کنیم.

هنگام fine-tuning روی داده‌مجموعه SWAG، چهار توالی ورودی ساخته می‌شود که هر کدام شامل الحاق جمله داده‌شده (جمله A) و یک ادامه ممکن (جمله B) هستند. تنها پارامتر خاص وظیفه که معرفی می‌شود یک بردار است که حاصل‌ضرب داخلی آن با نمایش توکن [CLS] یعنی C، امتیاز هر گزینه را نشان می‌دهد و با لایه softmax نرمال می‌شود.

مدل برای ۳ epoch با نرخ یادگیری $2e-5$ و اندازه batch برابر با ۱۶ fine-tuning می‌شود. نتایج در جدول ۴ ارائه شده‌اند. BERT$_{\text{LARGE}}$ سیستم baseline نویسندگان ESIM+ELMo را با +۱%.۲۷ و GPT OpenAI را با ۳%.۸ پیشی می‌گیرد.

۵ مطالعات Ablation

در این بخش، آزمایش‌های ablation را روی جنبه‌های مختلف BERT انجام می‌دهیم تا اهمیت نسبی آن‌ها را بهتر درک کنیم. مطالعات ablation اضافی را می‌توان در ضمیمه پ یافت.

Test	Dev	System
۷.۵۲	۹.۵۱	ESIM+GloVe
۲.۵۹	۱.۵۹	ESIM+ELMo
۰.۷۸	–	GPT OpenAI
–	۶.۸۱	BERT$_{\text{BASE}}$
۳.۸۶	**۶.۸۶**	BERT$_{\text{LARGE}}$
۰.۸۵	–	†(expert) Human
۰.۸۸	–	†annotations) ۵) Human

Table ۴: SWAG dev and test accuracies. †Human performance is measured with ۱۰۰ samples, as reported in the paper. SWAG the in ported re- Human† accuracies. Test and Dev

۵.۱ تأثیر وظایف Pre-training

ما اهمیت دو جهتی بودن عمیق BERT را با ارزیابی دو هدف pre-training نشان می‌دهیم، با استفاده از همان داده‌های pre-training، fine-tuning طرح و ابرپارامترها همانند BERT$_{\text{BASE}}$:

بدون NSP: یک مدل دو جهتی که با استفاده از "masked LM" (MLM) آموزش داده شده اما بدون وظیفه "next

Set Dev					
SQuAD (F۱)	SST-۲ (Acc)	MRPC (Acc)	QNLI (Acc)	MNLI-m (Acc)	Tasks
۵.۸۸	۷.۹۲	۷.۸۶	۴.۸۸	۴.۸۴	$_{\text{BASE}}$BERT
۹.۸۷	۶.۹۲	۵.۸۶	۹.۸۴	۹.۸۳	NSP No
۸.۷۷	۱.۹۲	۵.۷۷	۳.۸۴	۱.۸۲	NSP No & LTR
۹.۸۴	۶.۹۱	۷.۷۵	۱.۸۴	۱.۸۲	BiLSTM +

using tasks pre-training the over Ablation :۵ Table trained is NSP" "No architecture. $_{\text{BASE}}$BERT the "LTR task. prediction sentence next the without with- LM left-to-right a as trained is NSP" No & GPT. OpenAI like prediction، sentence next the out on BiLSTM initialized randomly a adds BiLSTM" +" fine- during model NSP" No + "LTR the of top tuning.

(NSP) prediction" sentence است.

LTR و بدون NSP: یک مدل فقط با زمینه چپ که با استفاده از LM استاندارد Left-to-Right (LTR) آموزش داده شده، به جای .MLM محدودیت فقط چپ همچنین در fine-tuning اعمال شد، زیرا حذف آن موجب ایجاد ناسازگاری بین pre-train و fine-tune می‌شد که عملکرد در وظایف downstream را کاهش می‌داد. علاوه بر این، این مدل بدون وظیفه NSP پیش‌آموزش داده شد. این مستقیماً قابل مقایسه با GPT OpenAI است، اما با استفاده از مجموعه داده آموزشی بزرگ‌تر ما، نمایش ورودی ما و طرح fine-tuning ما.

ابتدا تأثیر وظیفه NSP را بررسی می‌کنیم. در جدول ۵ نشان می‌دهیم که حذف NSP عملکرد را به‌طور قابل توجهی در QNLI، MNLI و SQuAD ۱.۱ کاهش می‌دهد. سپس تأثیر آموزش نمایش‌های دو جهتی را با مقایسه "بدون NSP" با "LTR و بدون NSP" ارزیابی می‌کنیم. مدل LTR در تمام وظایف نسبت به مدل MLM عملکرد ضعیف‌تری دارد و کاهش‌های قابل توجهی در MRPC و SQuAD مشاهده می‌شود.

برای SQuAD، واضح است که مدل LTR در پیش‌بینی توکن‌ها عملکرد ضعیفی خواهد داشت، زیرا حالات پنهان سطح توکن هیچ زمینه‌ای از سمت راست ندارند. برای تلاش جدی در تقویت سیستم LTR، یک BiLSTM با مقداردهی اولیه تصادفی بر روی آن اضافه کردیم. این کار نتایج را در SQuAD به‌طور قابل توجهی بهبود می‌بخشد، اما نتایج همچنان بسیار ضعیف‌تر از مدل‌های دو جهتی پیش‌آموزش داده‌شده هستند. استفاده از BiLSTM عملکرد را در وظایف GLUE کاهش می‌دهد.

ما می‌دانیم که همچنین می‌توان مدل‌های جداگانه LTR و RTL آموزش داد و هر توکن را به‌صورت الحاق دو مدل نمایش داد، همانند کاری که ELMo انجام می‌دهد. با این حال: (a) این دو برابر هزینه یک مدل دو جهتی واحد است؛ (b) این برای وظایفی مانند QA غیرشهودی است، زیرا مدل RTL نمی‌تواند پاسخ را بر اساس سؤال شرطی کند؛ (c) این به‌طور قطعی کمتر از یک مدل دو جهتی عمیق قدرتمند است، زیرا نمی‌تواند از هر دو زمینه چپ و راست در هر لایه استفاده کند.

۲.۵ تأثیر اندازه مدل

در این بخش، تأثیر اندازه مدل بر دقت وظایف فاین‌تیونینگ را بررسی می‌کنیم. ما چندین مدل BERT با تعداد متفاوتی از لایه‌ها، واحدهای پنهان و سرهای توجه آموزش دادیم، در حالی که سایر ابرپارامترها و روش آموزش همان‌طور که قبلاً توضیح داده شد، حفظ شدند.

نتایج در برخی از وظایف انتخابی GLUE در جدول ۶ نشان داده شده است. در این جدول، میانگین دقت مجموعه Dev از ۵ شروع تصادفی فاین‌تیونینگ گزارش شده است. می‌توان دید که مدل‌های بزرگ‌تر به‌طور قطعی باعث بهبود دقت در تمامی چهار مجموعه داده می‌شوند، حتی برای MRPC که تنها ۳۶۰۰ نمونه آموزشی برچسب‌گذاری شده دارد و تفاوت قابل توجهی با وظایف پیش‌آموزشی دارد. همچنین شاید تعجب‌آور باشد که ما قادر به دستیابی به چنین بهبودهای قابل توجهی بر روی مدل‌هایی هستیم که نسبت به ادبیات موجود، قبلاً نسبتاً بزرگ هستند. به عنوان مثال، بزرگ‌ترین ترنسفورمر مورد بررسی در ؟ دارای (L=۶، H=۱۰۲۴، A=۱۶) و ۱۰۰ میلیون پارامتر برای انکودر است، و بزرگ‌ترین ترنسفورمر یافت شده در ادبیات دارای (L=۶۴، H=۵۱۲، A=۲) و ۲۳۵ میلیون پارامتر است (؟). در مقایسه، $_{\text{BASE}}$BERT دارای ۱۱۰ میلیون پارامتر و $_{\text{LARGE}}$BERT دارای ۳۴۰ میلیون پارامتر است.

مدت‌هاست که مشخص شده افزایش اندازه مدل منجر به بهبود مداوم در وظایف مقیاس بزرگ مانند ترجمه ماشینی و مدل‌سازی زبان می‌شود، همان‌طور که توسط پریپلکسی داده‌های آموزشی نگه‌داشته‌شده در جدول ۶ نشان داده شده است. با این حال، ما بر این باوریم که این نخستین کار است که به‌طور قانع‌کننده نشان می‌دهد مقیاس‌دهی به اندازه‌های مدل بسیار بزرگ نیز منجر به بهبودهای قابل توجه در وظایف با مقیاس بسیار کوچک می‌شود، به شرط آنکه مدل به اندازه کافی پیش‌آموزش دیده باشد. ؟ نتایج مختلطی درباره تأثیر افزایش اندازه bi-LM پیش‌آموزشی از دو لایه به چهار لایه در وظایف پایین‌دستی ارائه دادند و ؟ به‌طور گذرا ذکر کردند که افزایش اندازه بعد پنهان از ۲۰۰ به ۶۰۰ مفید بود، اما افزایش بیشتر تا ۱۰۰۰ به بهبودهای بیشتری منجر نشد. هر دوی این کارهای پیشین از روش مبتنی بر ویژگی استفاده کرده‌اند—— ما فرض می‌کنیم زمانی که مدل مستقیماً روی وظایف پایین‌دستی فاین‌تیون می‌شود و تنها از تعداد بسیار کمی پارامتر اضافی با مقداردهی تصادفی استفاده می‌کند، مدل‌های خاص وظیفه می‌توانند از نمایش‌های پیش‌آموزش‌دیده بزرگ‌تر و بیانگرتر بهره‌مند شوند حتی زمانی که داده‌های وظیفه پایین‌دستی بسیار کم باشد.

۳.۵ روش مبتنی بر ویژگی با BERT

تمام نتایج BERT که تاکنون ارائه شده‌اند، از روش فاین‌تیونینگ استفاده کرده‌اند، جایی که یک لایه ساده دسته‌بندی به مدل پیش‌آموزش‌دیده اضافه شده و همه پارامترها به‌طور مشترک روی یک وظیفه پایین‌دستی فاین‌تیون می‌شوند.

ابرپارامترها (Hyperparams)			دقت مجموعه‌ی توسعه (Dev Set (Accuracy			
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-۲
۳	۷۶۸	۱۲	۵/۸۴	۷۷/۹	۷۹/۸	۸۸/۴
۶	۷۶۸	۳	۵/۲۴	۸۰/۶	۸۲/۲	۹۰/۷
۶	۷۶۸	۱۲	۴/۶۸	۸۱/۹	۸۴/۸	۹۱/۳
۱۲	۷۶۸	۱۲	۳/۹۹	۸۴/۴	۸۶/۷	۹۲/۹
۱۲	۱۰۲۴	۱۶	۳/۵۴	۸۵/۷	۸۶/۹	۹۳/۳
۲۴	۱۰۲۴	۱۶	۳/۲۳	۸۶/۶	۸۷/۸	۹۳/۷

Table ۶: مطالعه‌ی حذفی (Ablation) بر روی اندازه‌ی مدل BERT. #L = تعداد لایه‌ها؛ #H = اندازه‌ی بردارهای پنهان؛ #A = تعداد سرهای توجه (Attention Heads). عبارت "LM (ppl)" نشان‌دهنده‌ی پیچیدگی مدل زبانی پوشانده‌شده (Masked LM) Perplexity بر روی داده‌های آموزشی کنارگذاشته‌شده است.

با این حال، روش مبتنی بر ویژگی، که در آن ویژگی‌های ثابت از مدل پیش‌آموزش‌دیده استخراج می‌شوند، مزایای خاص خود را دارد. اول، همه وظایف نمی‌توانند به‌راحتی با معماری رمزگذار ترنسفورمر نمایش داده شوند و بنابراین نیاز به اضافه کردن یک معماری مدل خاص وظیفه دارند. دوم، مزایای محاسباتی مهمی وجود دارد، زیرا می‌توان نمایش پرهزینه داده‌های آموزشی را یک بار پیش‌محاسبه کرد و سپس آزمایش‌های متعددی با مدل‌های ارزان‌تر روی این نمایش اجرا نمود.

در این بخش، دو روش را با اعمال BERT روی وظیفه شناسایی موجودیت‌های نام‌دار (NER) مجموعه داده CoNLL-۲۰۰۳ (؟) مقایسه می‌کنیم. در ورودی BERT، از مدل WordPiece با حفظ حروف استفاده می‌کنیم و حداکثر زمینه سند ارائه شده توسط داده‌ها را شامل می‌کنیم. مطابق رویه استاندارد، این وظیفه را به‌صورت برچسب‌گذاری فرموله می‌کنیم، اما از لایه CRF در خروجی استفاده نمی‌کنیم. نمایش اولین زیر–توکن به‌عنوان ورودی برای دسته‌بند سطح توکن روی مجموعه برچسب‌های NER استفاده می‌شود.

برای بررسی اثر فاین‌تیونینگ، روش مبتنی بر ویژگی را با استخراج فعال‌سازی‌ها از یک یا چند لایه *بدون* فاین‌تیون هیچ پارامتری از BERT اعمال می‌کنیم. این تعبیه‌های متنی به‌عنوان ورودی به یک BiLSTM دو لایه ۷۶۸ بعدی با مقداردهی تصادفی قبل از لایه دسته‌بندی استفاده می‌شوند.

نتایج در جدول ۷ ارائه شده‌اند. $\text{BERT}_{\text{LARGE}}$ عملکردی رقابتی با روش‌های پیشرو دارد. بهترین روش، نمایش‌های توکن را از چهار لایه مخفی بالایی ترنسفورمر پیش‌آموزش‌دیده متصل می‌کند، که تنها ۳.۰ F۱ پایین‌تر از فاین‌تیونینگ کل مدل است. این نشان می‌دهد که BERT برای هر دو روش فاین‌تیونینگ و مبتنی بر ویژگی مؤثر است.

۶ نتیجه‌گیری

بهبودهای تجربی اخیر ناشی از یادگیری انتقالی با مدل‌های زبانی نشان داده‌اند که پیش‌آموزش غنی و بدون نظارت، بخشی اساسی از بسیاری از سیستم‌های درک زبان است. به‌ویژه، این نتایج حتی به وظایف با منابع کم اجازه می‌دهد تا از معماری‌های عمیق یک‌جهته بهره‌مند شوند. سهم اصلی ما تعمیم بیشتر این یافته‌ها به معماری‌های دوجهته عمیق است، که امکان استفاده موفقیت‌آمیز از همان مدل پیش‌آموزش‌دیده برای مجموعه گسترده‌ای از وظایف NLP را فراهم می‌کند.

مراجع

پیوست مقاله :"BERT: پیش‌آموزش ترنسفورمرهای دوجهته عمیق برای درک زبان"

ما پیوست را به سه بخش سازماندهی کرده‌ایم:

- جزئیات اضافی پیاده‌سازی BERT در پیوست آ ارائه شده است؛
- جزئیات اضافی مربوط به آزمایش‌های ما در پیوست ب ارائه شده است؛ و
- مطالعات تحلیل تفکیکی (ablation) اضافی در پیوست پ ارائه شده است.

ما مطالعات تحلیل تفکیکی اضافی برای BERT ارائه می‌دهیم که شامل موارد زیر است:

- تأثیر تعداد گام‌های آموزش؛ و
- تحلیل تفکیکی برای روش‌های مختلف ماسک‌گذاری.

BERT for Details Additional آ

Tasks Pre-training the of Illustration ۱.آ

tasks pre-training the of examples provide We following. the in

Procedure Masking the and LM Masked

`is dog my` is sentence unlabeled the Assuming proce- masking random the during and ،`hairy` corre- (which token ۴-th the chose we dure can procedure masking our ،(`hairy` to sponding by illustrated further be

- with word the Replace time: the of ۸۰% → `hairy is dog my` e.g.، token، `[MASK]` the `[MASK] is dog my`

F۱ مجموعه‌ی آزمون (Test)	F۱ مجموعه‌ی توسعه (Dev)	سامانه (System)
۹۲/۲	۹۵/۷	(؟) ELMo
۹۲/۶	–	(؟) CVT
۹۳/۱	–	(؟) CSE
		رویکرد تنظیم دقیق (Fine-tuning approach)
۹۲/۸	۹۶/۶	BERT$_{LARGE}$
۹۲/۴	۹۶/۴	BERT$_{BASE}$
		رویکرد مبتنی بر ویژگی BERT$_{BASE}$
–	۹۱/۰	تعبیه‌ها (Embeddings)
–	۹۵/۶	لایه‌ی ماقبل آخر (Second-to-Last Hidden)
–	۹۴/۹	لایه‌ی آخر (Last Hidden)
–	۹۵/۹	مجموع وزنی چهار لایه‌ی آخر (Weighted Four Last Sum Hidden)
–	۹۶/۱	اتصال چهار لایه‌ی آخر (Concat Four Last Hidden)
–	۹۵/۵	مجموع وزنی هر ۱۲ لایه (Weighted ۱۲ All Sum Layers)

Table ۷: نتایج تشخیص موجودیت نام‌دار (NER) روی مجموعه‌ی CoNLL-۲۰۰۳. ابرپارامترها با استفاده از مجموعه‌ی توسعه (Dev) انتخاب شده‌اند. مقادیر گزارش‌شده‌ی Dev و Test میانگین ۵ اجرای تصادفی با همان ابرپارامترها هستند.

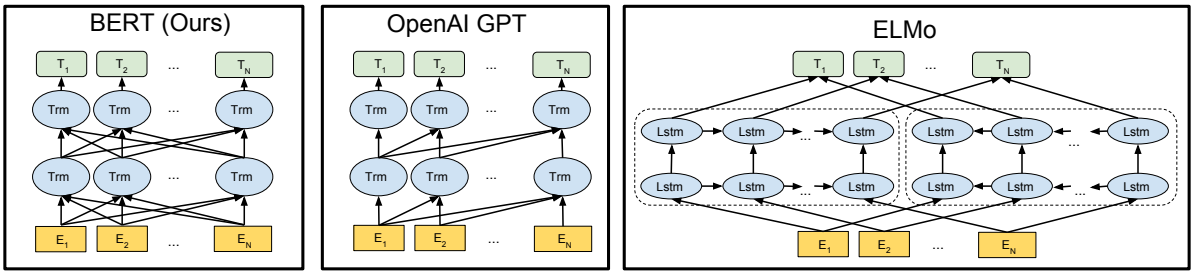


Figure ۳: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among three, BERT has the most powerful representations. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

train- model langauge standard to Compared predictions make only LM masked the ing، sug- which batch، each in tokens of ۱۵% on re- be may steps pre-training more that gests Sec- In converge. to model the for quired con- does MLM that demonstrate we پ.۱ tion left-to-right a than slower marginally verge the but token)، every predicts (which model far model MLM the of improvements empirical cost. training increased the outweigh

sen- next The **Prediction Sentence Next** fol- the in illustrated be can task prediction tence

with word the Replace time: the of ۱۰% ● `my → hairy is dog my` e.g.، word، random a `apple is dog`

un- word the Keep time: the of ۱۰% ● `dog my → hairy is dog my` e.g.، changed، the bias to is this of purpose The `.hairy is` observed actual the towards representation word.

the that is procedure this of advantage The which know not does encoder Transformer have which or predict to asked be will it words forced is it so words، random by replaced been representa- contextual distributional a keep to because Additionally، token. input *every* of tion all of ۵%. ۱ for occurs only replacement random to seem not does this ،(۱۵% of ۱۰% (i.e.، tokens capa- understanding language model's the harm impact the evaluate we ،پ.۲ Section In bility. procedure. this

our in pretraing up speed To length. quence se- with model the pre-train we experiments، steps. the of ۹۰% for ۱۲۸ of length quence se- of steps the of ۱۰% rest the train we Then، embed- positional the learn to ۵۱۲ of quence dings.

Procedure Fine-tuning ۳.آ

hyperparameters model most fine-tuning، For excep- the with pre-training، in as same the are number and rate، learning size، batch the of tion was probability dropout The epochs. training of hyperparam- optimal The .۱.۰ at kept always the found we but task-specific، are values eter well work to values possible of range following tasks: all across

- **size Batch**: ۳۲ ،۱۶
- **rate Learning (Adam)**: ۲e-۵ ۳e-۵، ۵e-۵،
- **epochs of Number**: ۴ ،۳ ،۲

(e.g.، sets data large that observed also We less far were examples) training labeled ۱۰۰k+ small than choice hyperparameter to sensitive it so fast، very typically is Fine-tuning sets. data search exhaustive an run simply to reasonable is model the choose and parameters above the over set. development the on best performs that

،and ELMo BERT، of Comparison ۴.آ GPT OpenAI

pop- recent in differences the studies we Here including models learning representation ular com- The BERT. and GPT OpenAI ELMo، are architectures model the between parisons ad- in that Note .۳ Figure in visually shown BERT differences، architecture the to dition approaches، fine-tuning are GPT OpenAI and approach. feature-based a is ELMo while pre-training existing comparable most The trains which GPT، OpenAI is BERT to method text large a on LM Transformer left-to-right a decisions design the of many fact، In corpus. as it make to made intentionally were BERT in meth- two the that so possible as GPT to close ar- core The compared. minimally be could ods bi-directionality the that is work this of gument Sec- in presented tasks pre-training two the and empiri- the of majority the for account ۱.۳ tion are there that note do we but improvements، cal BERT how between differences other several trained: were GPT and examples. lowing

Input = `[SEP] store [MASK] to went man the [CLS] [SEP] milk [MASK] gallon a bought he`

Label = `IsNext`

Input = `[SEP] store the to [MASK] man the [CLS] [SEP] birds ##less flight are [MASK] penguin`

Label = `NotNext`

Procedure Pre-training ۲.آ

we sequence، input training each generate To which corpus، the from text of spans two sample are they though even “sentences” as to refer we (but sentences single than longer much typically receives sentence first The also). shorter be can the receives second the and embedding `A` the actual the is `B` time the of ۵۰% embedding. `B` time the of ۵۰% and `A` follows that sentence next the for done is which sentence، random a is it sam- are They task. prediction” sentence “next to- ۵۱۲ $\leq$ is length combined the that such pled Word- after applied is masking LM The kens. rate masking uniform a with tokenization Piece to given consideration special no and ،۱۵% of pieces. word partial sequences ۲۵۶ of size batch with train We to- ۱۲۸،۰۰۰ = tokens ۵۱۲ * sequences ۲۵۶) ap- is which steps، ۱،۰۰۰،۰۰۰ for kens/batch) word billion ۳.۳ the over epochs ۴۰ proximately ۱e- of rate learning with Adam use We corpus. of decay weight L۲ ،$\beta_2 = 0.999$ ،$\beta_1 = 0.9$ ،۴ ۱۰،۰۰۰ first the over warmup rate learning ،0.01 We rate. learning the of decay linear and steps، We layers. all on ۱.۰ of probability dropout a use stan- the than rather (؟) activation `gelu` a use train- The GPT. OpenAI following ،`relu` dard like- LM masked mean the of sum the is loss ing prediction sentence next mean the and lihood likelihood.

۴ on performed was $_{\text{BASE}}$BERT of Training TPU ۱۶) configuration Pod in TPUs Cloud was $_{\text{LARGE}}$BERT of Training [۱۳]total). chips chips TPU ۶۴) TPUs Cloud ۱۶ on performed com- to days ۴ took pre-training Each total). plete.

ex- disproportionately are sequences Longer se- the to quadratic is attention because pensive

[۱۳] https://cloudplatform.googleblog.com/۲۰۱۸/۰۶/Cloud-TPU-now-offers-preemptible-pricing-and-global-availability.html

originally were which of descriptions the :؟ in summarized

Infer- Language Natural Multi-Genre **MNLI** entailment crowdsourced large-scale، a is ence sentences، of pair a Given .(؟) task classification sen- second the whether predict to is goal the *neutral* or ،*contradiction* ،*entailment* an is tence one. first the to respect with

clas- binary a is Pairs Question Quora **QQP** if determine to is goal the where task sification semantically are Quora on asked questions two .(؟) equivalent

Inference Language Natural Question **QNLI** Answer- Question Stanford the of version a is a to converted been has which (؟) Dataset ing ex- positive The .(؟) task classification binary do which pairs sentence) (question، are amples ex- negative the and answer، correct the contain same the from sentence) (question، are amples answer. the contain not do which paragraph

is Treebank Sentiment Stanford The **SST-۲** con- task classification single-sentence binary a re- movie from extracted sentences of sisting senti- their of annotations human with views .(؟) ment

Acceptabil- Linguistic of Corpus The **CoLA** task، classification single-sentence binary a is ity English an whether predict to is goal the where .(؟) not or "acceptable" linguistically is sentence

Similar- Textual Semantic The **STS-B** sentence of collection a is Benchmark ity other and headlines news from drawn pairs score a with annotated were They .(؟) sources two the similar how denoting ۵ to ۱ from meaning. semantic of terms in are sentences

Cor- Paraphrase Research Microsoft **MRPC** ex- automatically pairs sentence of consists pus human with sources، news online from tracted pair the in sentences the whether for annotations .(؟) equivalent semantically are

bi- a is Entailment Textual Recognizing **RTE** with but MNLI، to similar task entailment nary [۱۴].(؟) data training less much

[۱۴] re- fine-tuning single-task report only we that Note could approach fine-tuning multitask A paper. this in sults ex- For further. even performance the push potentially RTE on improvements substantial observe did we ample، MNLI. with training multi-task from

- BooksCorpus the on trained is GPT on trained is BERT words)؛ (۸۰۰M and words) (۸۰۰M BooksCorpus the words). (۲،۵۰۰M Wikipedia
- and (`[SEP]`) separator sentence a uses GPT in- only are which (`[CLS]`) token classifier learns BERT time؛ fine-tuning at troduced embeddings `B/A` sentence and `[CLS]` ،`[SEP]` pre-training. during
- batch a with steps ۱M for trained was GPT trained was BERT words؛ ۳۲،۰۰۰ of size ۱۲۸،۰۰۰ of size batch a with steps ۱M for words.
- ۵e- of rate learning same the used GPT BERT experiments؛ fine-tuning all for ۵ learning fine-tuning task-specific a chooses devel- the on best the performs which rate set. opment

we differences، these of effect the isolate To ۱.۵ Section in experiments ablation perform im- the of majority the that demonstrate which two the from coming fact in are provements they bidirectionality the and tasks pre-training enable.

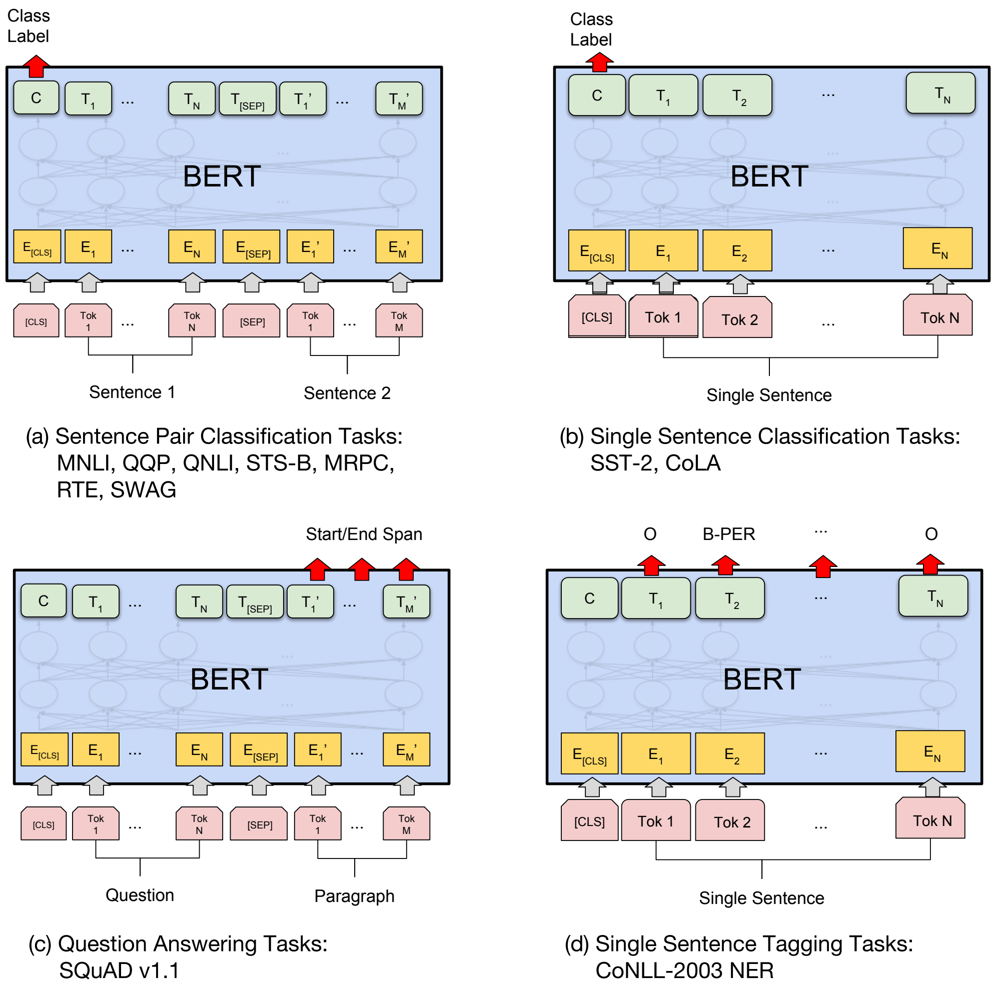
آ.۵ Illustrations of Fine-tuning on Different Tasks

dif- on BERT fine-tuning of illustration The Our .۴ Figure in seen be can tasks ferent incorporat- by formed are models task-specific layer، output additional one with BERT ing be to need parameters of number minimal a so and (a) tasks، the Among scratch. from learned (d) and (c) while tasks sequence-level are (b) represents E figure، the In tasks. token-level are contex- the represents T_i embedding، input the spe- the is [CLS] ،i token of representation tual is [SEP] and output، classification for symbol cial non-consecutive separate to symbol special the sequences. token

ب Setup Experimental Detailed

ب.۱ Experiments. Benchmark GLUE the for Descriptions Detailed

from obtained are ۱Table in results GLUE Our https://gluebenchmark.com/leaderboard https://blog.openai.com/ and GLUE The .language-unsupervised datasets، following the includes benchmark



Tasks. Different on BERT Fine-tuning of Illustrations :۴ Figure

lan- natural small a is NLI Winograd **WNLI** web- GLUE The .(؟) dataset inference guage con- the with issues are there that notes page sys- trained every and [۱۵] dataset، this of struction per- has GLUE to submitted been that's tem accuracy baseline ۱.۶۵ the than worse formed therefore We class. majority the predicting of For GPT. OpenAI to fair be to set this exclude the predicted always we submission، GLUE our class. majority

Studies Ablation Additional پ

Steps Training of Number of Effect ۱.پ

after accuracy Dev MNLI presents ۵ Figure been has that checkpoint a from fine-tuning answer to us allows This steps. k for pre-trained questions: following the

such need really BERT Does Question: .۱ (۱۲۸،۰۰۰ pre-training of amount large a achieve to steps) ۱،۰۰۰،۰۰۰ * words/batch accuracy؟ fine-tuning high

almost achieves BASEBERT Yes، Answer: when MNLI on accuracy additional ۰%.۱ ۵۰۰k to compared steps ۱M on trained steps.

con- pre-training MLM Does Question: .۲ since pre-training، LTR than slower verge each in predicted are words of ۱۵% only word؟ every than rather batch

con- does model MLM The Answer: model. LTR the than slower slightly verge accuracy absolute of terms in However، the outperform to begins model MLM the immediately. almost model LTR

[۱۵] https://gluebenchmark.com/faq

پ.۲ Masking Different for Ablation Procedures

uses BERT that mention we ،۱.۳ Section In tokens target the masking for strategy mixed a language masked the with pre-training when an is following The objective. (MLM) model different of effect the evaluate to study ablation strategies. masking

strate- masking the of purpose the that Note pre- between mismatch the reduce to is gies sym- [MASK] the as fine-tuning، and training stage. fine-tuning the during appears never bol and MNLI both for results Dev the report We fine-tuning both report we NER، For NER. the expect we as approaches، feature-based and feature-based the for amplified be will mismatch chance the have not will model the as approach representations. the adjust to

token. random other

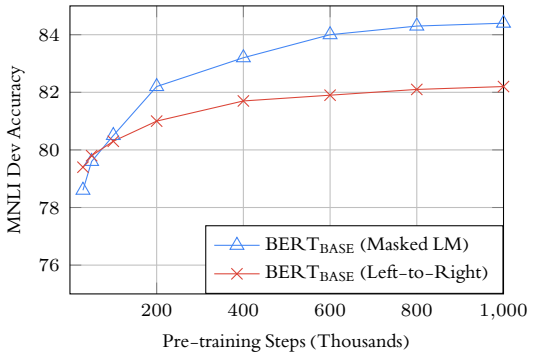
rep- table the of part left the in numbers The strategies specific the of probabilities the resent uses (BERT pre-training MLM during used paper the of part right The .(۱۰% ،۱۰% ،۸۰% feature- the For results. set Dev the represents layers ۴ last the concatenate we approach، based to shown was which features، the as BERT of .۳.۵ Section in approach best the be

fine-tuning that seen be can it table the From strate- masking different to robust surprisingly is the only using expected، as However، gies. applying when problematic was strategy Mask Interest- NER. to approach feature-based the performs strategy Rnd the only using ingly، well. as strategy our than worse much

نتایج مجموعه‌ی توسعه (Dev Set)		
NER		MNLI
مبتنی بر ویژگی (Feature-based)	تنظیم دقیق (Fine-tune)	تنظیم دقیق (Fine-tune)
۹۴/۹	۹۵/۴	۸۴/۲
۹۴/۰	۹۴/۹	۸۴/۳
۹۴/۶	۹۵/۲	۸۴/۱
۹۴/۷	۹۵/۲	۸۴/۴
۹۴/۶	۹۴/۸	۸۳/۷
۹۴/۶	۹۴/۹	۸۳/۶

Table ۸: تحلیل تفکیکی (Ablation) بر روی استراتژی‌های مختلف ماسک‌گذاری.

the In .۸ Table in presented are results The to- target the replace we that means Mask table، Same MLM؛ for symbol [MASK] the with ken Rnd is؛ as token target the keep we that means an- with token target the replace we that means



steps. training of number over Ablation :۵ Figure fine-tuning، after accuracy MNLI the shows This pre- been have that parameters model from starting .k of value the is x-axis The steps. k for trained