۱.

سیستم‌های open-domain:
این سیستم‌ها برای پاسخگویی به سوالات در طیف گسترده‌ای از موضوعات طراحی شده‌اند. آن‌ها می‌توانند به سؤالات مربوط به هر چیزی پاسخ دهند و از مجموعه داده‌های بزرگ و متنوع مانند کل اینترنت استفاده می‌کنند. این سیستم‌ها معمولاً بر تکنیک‌هایی مانند پردازش زبان طبیعی، بازیابی اطلاعات، و یادگیری ماشینی متکی هستند تا مرتبط‌ترین پاسخ‌ها را از منابع گسترده و متنوع بیابند و ارائه دهند.

سیستم‌های close-domain:
این سیستم‌های پاسخگویی به سوال، تخصصی هستند و به یک دامنه یا موضوع خاص محدود می‌شوند. این سیستم‌ها برای پاسخگویی به سوالات تنها در آن دامنه از پیش تعریف شده طراحی شده‌اند. این سیستم‌ها متمرکزتر هستند و اغلب از داده‌ها و پایگاه‌های دانش خاص دامنه استفاده می‌کنند. از آنجا که در یک محدوده محدود عمل می‌کنند، می‌توانند پاسخ‌های دقیق‌تر و جزئی‌تری برای حوزه خاص خود ارائه دهند.

تفاوت‌ها:

- دامنه: سیستم‌های دامنه باز طیف گسترده‌ای از موضوعات را پوشش می‌دهند، در حالی که سیستم‌های دامنه بسته بر یک حوزه خاص تمرکز دارند.
- منابع داده: سیستم‌های دامنه باز از مجموعه داده‌های بزرگ و متنوع استفاده می‌کنند؛ سیستم‌های دامنه بسته از داده‌های خاص و تخصصی دامنه استفاده می‌کنند.
- پیچیدگی: سیستم‌های دامنه باز به دلیل مقدار زیاد اطلاعاتی که باید پردازش و بازیابی کنند، معمولاً پیچیده‌تر هستند.
- دقت: سیستم‌های دامنه بسته می‌توانند در حوزه تخصصی خود دقیق‌تر باشند زیرا بر داده‌های متمرکز و خاص متکی هستند.
- کاربرد: سیستم‌های دامنه باز برای پاسخگویی به سوالات عمومی استفاده می‌شوند، در حالی که سیستم‌های دامنه بسته در حوزه‌های تخصصی مانند پزشکی، حقوقی، یا فنی کاربرد دارند.

.۲

درک مطلب ماشینی:

درک مطلب ماشینی (MRC) یکی از زیرمجموعه‌های پردازش زبان طبیعی (NLP) است که بر آموزش ماشین‌ها برای خواندن، فهمیدن و تفسیر متن نوشته شده تمرکز دارد. هدف MRC این است که ماشین‌ها بتوانند متن را به گونه‌ای پردازش و درک کنند که بتوانند به دقت به سوالات مربوط به محتوای آن پاسخ دهند. این شامل درک زمینه، شناسایی اطلاعات کلیدی و انجام استنتاج بر اساس متن است.

ارتباط با پاسخگویی به سوالات:

درک مطلب ماشینی به پاسخگویی به سوالات (QA) ارتباط نزدیکی دارد زیرا قابلیت پایه‌ای لازم برای سیستم‌های QA موثر را فراهم می‌کند. به طور کلی، MRC به سیستم‌های QA امکان می‌دهد تا متن‌هایی را که نیاز به استخراج پاسخ‌ها از آن‌ها دارند، درک کنند. توضیح نحوه ارتباط آن‌ها:

- فهم متن: MRC سیستم‌های QA را به قابلیت فهم و تفسیر متن مجهز می‌کند و اطمینان حاصل می‌کند که پاسخ‌ها از درک دقیق منبع متنی استخراج می‌شوند.
- استخراج پاسخ: در QA، به ویژه در محیط‌های دامنه باز، سیستم نیاز دارد تا متن‌های مختلف را برای یافتن پاسخ‌های دقیق به سوالات بخواند و درک کند. تکنیک‌های MRC برای استخراج اطلاعات مرتبط از این متن‌ها استفاده می‌شوند.
- آگاهی از زمینه: هم MRC و هم QA نیاز به درک زمینه برای ارائه پاسخ‌های معنادار دارند. MRC توانایی سیستم QA را برای درک زمینه‌ای که در آن سوال پرسیده شده و پاسخ یافت می‌شود، تقویت می‌کند.
- انجام استنتاج: سیستم‌های پیشرفته QA اغلب نیاز دارند تا پاسخ‌هایی که به صراحت بیان نشده‌اند اما می‌توانند از متن داده شده استنتاج شوند، را استنتاج کنند. MRC مکانیزم‌هایی را برای انجام این استنتاج‌ها فراهم می‌کند.

به طور خلاصه، MRC فرآیند زیرساختی است که به سیستم‌های QA امکان می‌دهد متن را بخوانند و درک کنند که برای پاسخ دادن دقیق به سوالات بر اساس آن متن ضروری است.

۳.

سوالات factoid:
این دسته از سوالات، سوالاتی هستند که به دنبال اطلاعات خاص و واقعی هستند. این سوالات معمولاً پاسخ‌های مختصری دارند که می‌توانند به عنوان درست یا غلط تأیید شوند. پاسخ‌ها اغلب کوتاه هستند، مانند یک کلمه یا یک عبارت کوتاه، و معمولاً شامل حقایق مشخصی مانند نام‌ها، تاریخ‌ها، مکان‌ها، اعداد یا جزئیات خاص دیگر هستند.

سوالات non-factoid:
سوالات non-factoid نیاز به پاسخ‌های مفصل‌تر و توضیحی دارند. این سوالات اغلب شامل توضیحات، توصیفات، نظرات یا تحلیل‌ها هستند. پاسخ‌ها معمولاً طولانی‌تر هستند و ممکن است پاسخ صحیح یگانه‌ای نداشته باشند. آن‌ها نیاز به درک و تفسیر اطلاعات پیچیده‌تر دارند.

تفاوت‌ها:
- ماهیت پاسخ‌ها:

factoid: کوتاه، خاص، و واقعی (مثلاً "پاریس"، "الکساندر گراهام بل").
non-factoid: طولانی، مفصل، و توضیحی (مثلاً توضیح فرآیند فتوسنتز).
- پیچیدگی:

factoid: ساده و سرراست.
non-factoid: پیچیده، اغلب نیازمند درک جامع و تفسیر.
- اعتبارسنجی:

factoid: به راحتی قابل تأیید به عنوان درست یا غلط.
non-factoid: ممکن است به راحتی قابل تأیید نباشند.

- بازیابی اطلاعات:

factoid: می‌توان با استفاده از بازیابی اطلاعات از پایگاه‌های داده ساختاریافته یا استخراج ساده متن پاسخ داد.
non-factoid: نیاز به درک عمیق‌تر زبان طبیعی، استدلال، و گاهی اوقات ترکیب اطلاعات از منابع متعدد دارد.

.۴

مزایای ترنسفورمرها در مقایسه با RNNها:

پردازش موازی:
ترنسفورمرها: ترنسفورمرها امکان پردازش موازی توالی‌های ورودی را فراهم می‌کنند که منجر به آموزش سریع‌تر می‌شود. هر توکن در توالی ورودی می‌تواند به طور همزمان پردازش شود.
RNNها: RNNها توالی‌های ورودی را به صورت ترتیبی، توکن به توکن، پردازش می‌کنند که می‌تواند کندتر باشد، به ویژه برای توالی‌های بلند.

مدیریت وابستگی‌های بلندمدت:
ترنسفورمرها: ترنسفورمرها از مکانیزم‌های توجه خود استفاده می‌کنند تا وابستگی‌ها بین توکن‌ها را صرف نظر از فاصله آن‌ها در توالی بگیرند، که آن‌ها را در مدیریت وابستگی‌های بلندمدت بسیار مؤثر می‌کند.
RNNها: RNNها به دلیل مشکل ناپدید شدن گرادیان در مدیریت وابستگی‌های بلندمدت مشکل دارند، هرچند تکنیک‌هایی مانند LSTM و GRU تا حدودی این مشکل را کاهش می‌دهند.

مقیاس‌پذیری:
ترنسفورمرها: ترنسفورمرها با مجموعه داده‌های بزرگ به خوبی مقیاس‌پذیر هستند و می‌توانند توالی‌های بسیار بزرگ را به طور مؤثر مدیریت کنند.
RNNها: RNNها با افزایش طول توالی‌ها می‌توانند از نظر محاسباتی پرهزینه و دشوار برای آموزش شوند.

زمینه دوطرفه(Bidirectional Context):
ترنسفورمرها: ترنسفورمرها می‌توانند زمینه دوطرفه را به طور طبیعی با استفاده از ساختارهای رمزگذار-رمزگشا یا مدل‌هایی مانند BERT که کل توالی را یکجا می‌خوانند، شامل شوند.
RNNها: RNNهای سنتی یکطرفه هستند و توالی‌ها را در یک جهت (به جلو یا عقب) پردازش می‌کنند. هرچند RNNهای دوطرفه وجود دارند، اما نیاز به پردازش جداگانه به جلو و عقب دارند که پیچیدگی را افزایش می‌دهد.

عملکرد در وظایف NLP:
ترنسفورمرها: ترنسفورمرها در بسیاری از وظایف NLP مانند ترجمه ماشینی، خلاصه‌سازی متن و پاسخگویی به سوالات به عملکرد پیشرفته‌ای دست یافته‌اند.
RNNها: در حالی که RNNها در وظایف مختلف خوب عمل می‌کنند، اغلب به سطح عملکردی که ترنسفورمرها در وظایف پیچیده NLP می‌رسند، نمی‌رسند.

معایب ترنسفورمرها در مقایسه با RNNها:

منابع محاسباتی:

ترنسفورمرها: ترنسفورمرها از نظر محاسباتی پرهزینه هستند و نیاز به منابع سخت‌افزاری قابل توجهی دارند، به ویژه برای مدل‌های بزرگ با لایه‌ها و پارامترهای زیاد.
RNNها: RNNها، به ویژه نسخه‌های ساده‌تر، می‌توانند از نظر منابع محاسباتی کمتر پرهزینه باشند.

پیچیدگی مدل:
ترنسفورمرها: معماری ترنسفورمرها پیچیده‌تر است و شامل چندین لایه توجه و شبکه‌های تغذیه جلو است که می‌تواند سخت‌تر برای پیاده‌سازی و تنظیم باشد.
RNNها: RNNها معماری ساده‌تری دارند که آن‌ها را راحت‌تر برای فهمیدن، پیاده‌سازی و عیب‌یابی می‌کند.

کارایی داده:
ترنسفورمرها: ترنسفورمرها معمولاً نیاز به مقادیر زیادی داده برای آموزش موثر دارند که می‌تواند در سناریوهای کم‌داده موجب محدودیت شود.
RNNها: RNNها گاهی اوقات می‌توانند در مجموعه داده‌های کوچکتر بهتر از ترنسفورمرها عمل کنند به دلیل ماهیت ترتیبی و نیازهای آموزشی ساده‌تر.

استفاده از حافظه:
ترنسفورمرها: ترنسفورمرها نیاز به حافظه بیشتری برای ذخیره حالت‌های میانی و وزن‌های توجه دارند که می‌تواند برای توالی‌های بسیار طولانی محدودیت باشد.
RNNها: RNNها با پردازش یک توکن در هر زمان، معمولاً حافظه کمتری برای هر توالی استفاده می‌کنند، اگرچه این می‌تواند بسته به معماری خاص (مانند LSTM یا GRU) متفاوت باشد.

۵.

Positional Embedding:

در مدل‌های مبتنی بر ترنسفورمر، positional embedding تکنیکی است که برای ارائه اطلاعات درباره موقعیت توکن‌ها در یک توالی استفاده می‌شود. برخلاف شبکه‌های عصبی بازگشتی (RNNها)، که به طور ذاتی داده‌ها را به صورت ترتیبی پردازش می‌کنند و ترتیب توکن‌های ورودی را حفظ می‌کنند، ترنسفورمرها همه توکن‌ها را به طور همزمان پردازش می‌کنند. این پردازش همزمان به این معناست که ترنسفورمرها به طور ذاتی ترتیب یا موقعیت توکن‌ها را در یک توالی درک نمی‌کنند.
positional embedding برای حل این مشکل معرفی شده‌اند و اطلاعات موقعیتی را در جاسازی‌های توکن‌ها کدگذاری می‌کنند.

اهمیت positional embedding:

- حفظ اطلاعات ترتیبی:

ترنسفورمرها: از آنجایی که ترنسفورمرها توکن‌های ورودی را به صورت موازی پردازش می‌کنند، نیاز به راهی برای پیگیری ترتیب توکن‌ها دارند. positional embedding اطمینان می‌دهند که مدل می‌تواند بین توکن‌ها بر اساس موقعیتشان در توالی تفاوت قائل شود.

افزایش درک context:
positional embedding به مدل کمک می‌کنند تا زمینه هر توکن را با ارائه اطلاعاتی درباره موقعیت آن نسبت به سایر توکن‌ها درک کند. این امر برای تسک‌هایی مانند ترجمه بسیار مهم است، جایی که معنای یک کلمه می‌تواند به شدت به موقعیت آن در جمله وابسته باشد.

بهبود عملکرد مدل:
افزودن positional embedding به ترنسفورمرها اجازه می‌دهد تا عملکرد بهتری در وظایفی که نیاز به درک ترتیب توالی دارند، داشته باشند. این امر منجر به بهبود در انواع وظایف NLP مانند ترجمه ماشینی، خلاصه‌سازی متن، و مدل‌سازی زبان می‌شود.

به طور خلاصه، positional embedding یکی از اجزای مهم مدل‌های مبتنی بر ترنسفورمر هستند، زیرا اطلاعات ترتیبی لازم را فراهم می‌کنند که به مدل اجازه می‌دهد ترتیب و زمینه توکن‌ها را در یک توالی درک کند. این قابلیت عملکرد و کاربرد ترنسفورمرها را در وظایف مختلف NLP به طور قابل توجهی افزایش می‌دهد.

۶.

ترنسفورمرهای فقط رمزگذار (Encoder-Only):

ترنسفورمرهای فقط رمزگذار از بخش رمزگذار معماری ترنسفورمر استفاده می‌کنند. این مدل‌ها برای پردازش توالی‌های ورودی و تولید نمایش‌های کدگذاری شده از داده طراحی شده‌اند.

ویژگی‌های کلیدی:

Self-Attention: بر روابط درون توالی ورودی تمرکز دارد.
خروجی: نمایش‌های متنی از توکن‌های ورودی تولید می‌کند.(خروجی توالی نیست.)
موارد استفاده: اغلب برای وظایفی که نیاز به درک داده‌های ورودی بدون تولید توالی‌های جدید دارند، استفاده می‌شود. مثال‌ها شامل طبقه‌بندی متن، تحلیل احساسات و شناسایی نام موجودیت‌ها است.
مثال‌ها:

BERT (نمایش‌های رمزگذار دوطرفه از ترنسفورمرها)

ترنسفورمرهای فقط رمزگشا (Decoder-Only):

ترنسفورمرهای فقط رمزگشا از بخش رمزگشای معماری ترنسفورمر استفاده می‌کنند. این مدل‌ها برای تولید توالی‌ها بر اساس یک زمینه یا پرسش طراحی شده‌اند.

ویژگی‌های کلیدی:

Masked Self-Attention: بر تولید توکن بعدی با توجه به توکن‌های قبلاً تولید شده تمرکز دارد.
خروجی: توالی‌های جدید تولید می‌کند (مثل تولید متن، مدل‌سازی زبان).
موارد استفاده: مناسب برای وظایفی که شامل تولید توالی‌ها هستند. مثال‌ها شامل تکمیل متن، مدل‌سازی زبان و تولید متن خودبازگشتی است.
مثال‌ها:

GPT-۲، GPT-۳

ترنسفورمرهای رمزگذار-رمزگشا (Encoder-Decoder):

ترنسفورمرهای رمزگذار-رمزگشا، که به عنوان مدل‌های توالی به توالی نیز شناخته می‌شوند، از هر دو بخش رمزگذار و رمزگشا معماری ترنسفورمر استفاده می‌کنند. این مدل‌ها برای تبدیل یک توالی ورودی به یک توالی خروجی متفاوت طراحی شده‌اند.

ویژگی‌های کلیدی:

رمزگذار: توالی ورودی را پردازش کرده و نمایشی زمینه‌دار ایجاد می‌کند.
رمزگشا: از این نمایش برای تولید توالی خروجی استفاده می‌کند.

Cross-Attention: رمزگشا به خروجی رمزگذار علاوه بر توجه به خود توجه می‌کند.
خروجی: یک توالی را به توالی دیگر تبدیل می‌کند (مثل ترجمه، خلاصه‌سازی).
موارد استفاده: مناسب برای وظایفی که شامل تبدیل یک نوع توالی به نوع دیگر هستند. مثال‌ها شامل ترجمه ماشینی، خلاصه‌سازی متن و پاسخگویی به سوالات است.

۷.

سیستم‌های Extractive:
این سیستم‌ها برای پیدا کردن و استخراج دقیق بخش‌های متنی از یک سند منبع که مستقیماً به یک سوال داده شده پاسخ می‌دهند، طراحی شده‌اند. سیستم بخشی از متن را که شامل پاسخ است شناسایی کرده و آن را عیناً استخراج می‌کند.

ویژگی‌های کلیدی:
منبع پاسخ: مستقیماً از متن ورودی گرفته شده است.
خروجی: پاسخ یک زیررشته از سند ورودی است.
تکنیک‌های استفاده شده: تکنیک‌هایی مانند طبقه‌بندی توکن، پیش‌بینی بازه، و مکانیزم‌های توجه برای شناسایی شروع و پایان بازه پاسخ در متن استفاده می‌شود.

مزایا:
دقت: دقت بالا زمانی که پاسخ به طور واضح در متن بیان شده باشد.
سادگی: نسبت به سیستم‌های abstractive برای پیاده‌سازی ساده‌تر است.

محدودیت‌ها:
خلاقیت: محدود به متن دقیق موجود در سند است؛ نمی‌تواند جملات جدید ایجاد کرده یا اطلاعات را بازنویسی کند.
وابستگی به زمینه: ممکن است در سوالاتی که نیاز به درک فراتر از استخراج ساده دارند، مانند ترکیب اطلاعات از چند جمله، مشکل داشته باشد.

سیستم‌های Abstractive:
این سیستم‌ها پاسخ‌هایی را ایجاد می‌کنند که ممکن است استخراج مستقیم از متن ورودی نباشند. در عوض، این سیستم‌ها جملات جدیدی ایجاد می‌کنند که اطلاعات مرتبط از سند منبع را خلاصه یا بازنویسی می‌کنند تا به سوال پاسخ دهند.

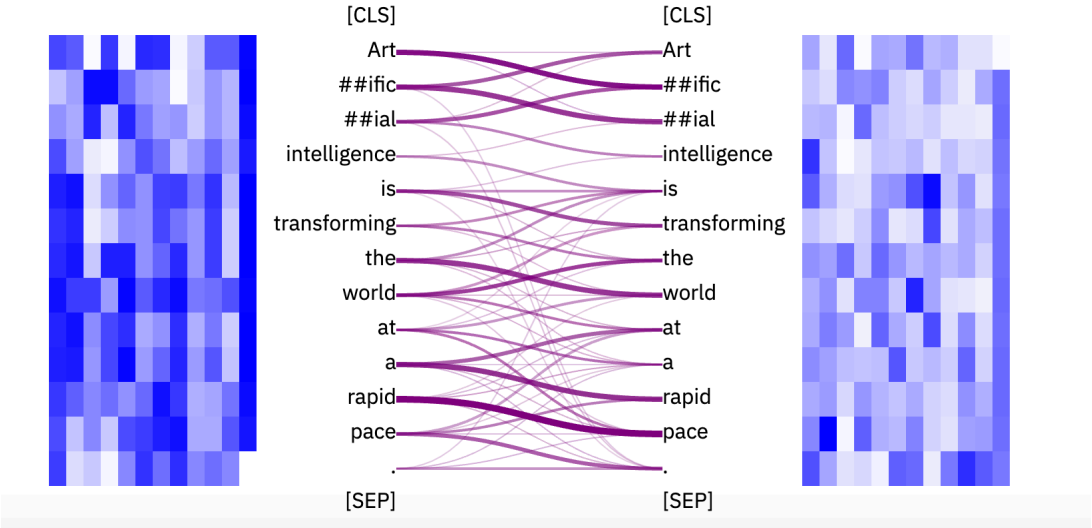
ویژگی‌های کلیدی:
منبع پاسخ: می‌تواند پاسخ‌ها را با استفاده از اطلاعات در متن ایجاد کند اما محدود به بخش‌های دقیق متن نیست.
خروجی: پاسخ یک متن جدید است که ممکن است اطلاعات را از چندین بخش سند ترکیب کند.
تکنیک‌های استفاده شده: تکنیک‌های تولید زبان طبیعی (NLG)، شامل مدل‌های توالی به توالی، ترنسفورمرها، و معماری‌های رمزگذار-رمزگشا.

مزایا:
انعطاف‌پذیری: می‌تواند پاسخ‌های طبیعی‌تر و جامع‌تر ایجاد کند، اطلاعات را به صورت مورد نیاز بازنویسی و ترکیب کند.
درک زمینه‌ای: بهتر در مدیریت سوالاتی که نیاز به درک و خلاصه‌سازی اطلاعات از بخش‌های مختلف متن دارند.

محدودیت‌ها:
پیچیدگی: پیاده‌سازی پیچیده‌تر و نیاز به تکنیک‌های پیشرفته تولید زبان طبیعی دارد.
دقت: ممکن است گاهی پاسخ‌های کمتر دقیقی نسبت به روش‌های استخراجی تولید کند، به ویژه اگر مدل اطلاعات نامربوط یا نادرست تولید کند.

۸.

input: Artificial intelligence is transforming the world at a rapid pace.



تصویر خروجی مکانیزم‌های توجه در مدل BERT را با استفاده از جمله ورودی نشان می‌دهد. بخش مرکزی تصویر جمله ورودی با توکن‌های آن را نمایش می‌دهد. خطوط متصل‌کننده توکن‌ها نحوه توزیع توجه را در جمله برای یک هد توجه خاص نشان می‌دهند.
وزن‌های توجه: خطوط بین توکن‌ها وزن‌های توجه را نشان می‌دهند، که میزان تمرکز یک توکن روی دیگری را نشان می‌دهند. خطوط ضخیم‌تر نشان‌دهنده وزن‌های توجه بالاتر هستند، یعنی توجه قوی‌تر بین آن توکن‌ها.
نقشه‌های توجه: نقشه‌های حرارتی در دو طرف نمرات توجه برای هر توکن را نمایش می‌دهند. سایه‌های آبی تیره‌تر نشان‌دهنده نمرات توجه بالاتر و سایه‌های روشن‌تر نشان‌دهنده نمرات توجه پایین‌تر هستند.

اهمیت توجه چندگانه در تسک NER:
۱. گرفتن روابط متنوع:
توجه چندگانه به مدل اجازه می‌دهد تا به طور همزمان به بخش‌های مختلف توالی ورودی توجه کند و جنبه‌ها و روابط مختلف بین توکن‌ها را بگیرد. برای NER، این امر بسیار حیاتی است زیرا معنا و طبقه‌بندی یک توکن می‌تواند به شدت به زمینه اطراف آن بستگی داشته باشد. هدهای مختلف می‌توانند بر روی نشانه‌های زمینه‌ای مختلف تمرکز کنند و توانایی مدل را در شناسایی دقیق موجودیت‌ها افزایش دهند.
۲. افزایش درک زمینه‌ای:
هر هد توجه می‌تواند بر موقعیت‌های مختلف در توالی تمرکز کند و درک غنی و دقیقی از زمینه ارائه دهد. به عنوان مثال، یک هد ممکن است بر روی کلمات همسایه فوری تمرکز کند در حالی که دیگری می‌تواند وابستگی‌های بلندمدت را بگیرد. این امر به خصوص در NER مهم است که زمینه یک کلمه می‌تواند چندین توکن را شامل شود.

۳. کاهش ابهام:
توجه چندگانه به کاهش ابهام کلماتی که ممکن است معانی یا کارکردهای مختلفی بسته به زمینه داشته باشند کمک می‌کند. با جمع‌آوری اطلاعات از چندین منظر، مدل می‌تواند موجودیت‌ها را بهتر رفع ابهام کند و به شناسایی دقیق‌تر NER منجر شود.
۴. بهبود مقاومت و تعمیم‌دهی:
افزونگی فراهم شده توسط چندین هد توجه، مدل را مقاوم‌تر می‌کند. اگر یک هد نتواند برخی وابستگی‌ها یا روابط را بگیرد، دیگران می‌توانند جبران کنند و به عملکرد قابل اعتمادتر منجر شوند. این قابلیت تعمیم‌دهی مدل را در زمینه‌ها و مجموعه داده‌های مختلف بهبود می‌بخشد.
۵. مدیریت ساختارهای پیچیده:
زبان طبیعی اغلب شامل ساختارهای پیچیده جملات است که موجودیت‌ها تحت تأثیر روابط نحوی و معنایی مختلف قرار می‌گیرند. توجه چندگانه به مدل اجازه می‌دهد تا این روابط پیچیده را به طور همزمان پردازش کند و در شناسایی موجودیت‌ها در جملات پیچیده موثرتر باشد.

۹.
توضیحات مربوط به بخش‌های اضافه شده در فایل ضمیمه شده به صورت کامنت قرار دارد.