

به نام خالق رنگین کمان

ستاره باباجانی – گزارش تمرین سری 5

سوال 1:

- سیستم های پاسخگویی به سوالات open-domain: این سیستم ها برای پاسخگویی به سؤالات از طیف گسترده ای از موضوعات بدون هیچ محدودیتی طراحی شده اند. همچنین این سیستم ها معمولاً از مدل های زبانی در مقیاس بزرگ، مانند مدل های GPT (تبدیل پیش آموزش دهی شده) استفاده می کنند که بر روی مجموعه متنی متنوعی از اینترنت آموزش داده شده اند. آنها متکی به درک زمینه سؤال و ایجاد پاسخ مرتبط بر اساس اطلاعات موجود، بدون محدود شدن به یک دامنه یا پایگاه داده خاص هستند.
- این سیستم ها اغلب از تکنیک هایی مانند بازیابی اطلاعات، درک زبان طبیعی و تولید زبان طبیعی برای ارائه پاسخ های دقیق و آموزنده استفاده می کنند.
- سیستم های پاسخگویی به سوالات close-domain: این نوع از سیستم ها برای پاسخ به سؤالات در یک دامنه یا حوزه موضوعی خاص طراحی شده اند. همچنین معمولاً بر داده های ساختاریافته یا پایگاه های دانش از پیش تعریف شده مرتبط با حوزه خاص متکی هستند. همچنین اغلب در برنامه های کاربردی تخصصی استفاده می شوند که در آن اطلاعات به خوبی تعریف شده و ساختار یافته است، مانند پشتیبانی مشتری، اسناد فنی یا تشخیص پزشکی زیرا می توانند پاسخ های دقیق و قابل اطمینان تری را در حوزه تخصص خود در مقایسه

با سیستم های open-domain ارائه دهند زیرا به مجموعه ای از داده ها یا دانش خاص برای دامنه دسترسی دارند.

تفاوت ها:

1. حوزه: سیستم های open-domain می توانند سؤالات طیف گسترده ای از موضوعات را بدون هیچ محدودیتی پاسخ دهند، در حالی که سیستم های close-domain به یک دامنه یا حوزه موضوعی خاص محدود می شوند.
 2. منبع داده: سیستم های open-domain معمولاً به مجموعه های بزرگی از داده های متنی بدون ساختار، مانند مقالات اینترنتی، کتاب ها و سایر منابع متنی متکی هستند. در صورتیکه سیستم های close-domain بر داده های ساختاریافته یا پایگاه های دانش از پیش تعریف شده مرتبط با حوزه خاصی که در آن فعالیت می کنند، متکی هستند.
 3. انعطاف پذیری در مقابل دقت: سیستم های open-domain با توانایی پاسخ دادن به سؤالات در مورد موضوعات مختلف، انعطاف بیشتری را ارائه می دهند، اما پاسخ های آنها ممکن است همیشه به اندازه پاسخ های سیستم های close-domain دقیق نباشد. در صورتیکه سیستم های close-domain دقت و اطمینان بیشتری را در حوزه تخصصی خود ارائه می دهند، اما فاقد انعطاف پذیری برای رسیدگی به سؤالات خارج از حوزه خاص خود هستند.
- منبع کمک کننده:

<https://open.passingai.com/content/12306.html>

سوال 2: درک مطلب ماشینی (MRC) یک کار در پردازش زبان طبیعی (NLP) است که در آن یک برنامه یا مدل کامپیوتری برای خواندن قسمتی از متن و پاسخ به سؤالات مربوط به آن آموزش می بیند. هدف، توسعه الگوریتم‌هایی است که می‌توانند زبان انسان را به گونه‌ای درک و تفسیر کنند که به آن‌ها امکان پاسخ دقیق به سؤالات بر اساس اطلاعات ارائه شده در متن داده شده را بدهد.

همان طور که گفته شد، هدف MRC این است که با آموزش کامپیوترها برای درک و تفسیر متن نوشته شده، از درک خواندن انسان تقلید کند. این نه تنها شامل شناخت تک تک کلمات بلکه درک روابط بین کلمات، جملات و پاراگراف‌ها برای درک معنای کلی یک قطعه است.

سیستم‌های MRC باید قادر به درک پدیده‌های مختلف زبانی مانند معناشناسی، نحو، وضوح مرجع مشترک و درک زمینه باشند. آن‌ها باید ظرایف زبان، از جمله عبارات اصطلاحی، استعاره‌ها و ابهامات را درک کنند تا معنای متن را دقیق تفسیر کنند.

در زمینه MRC، هدف نهایی پاسخ به سؤالات مطرح شده در مورد یک متن خاص است. این سؤالات می‌توانند از سؤالات واقعی (مثلاً "پایتخت فرانسه چیست؟") تا سؤالات استنتاجی و مبتنی بر استدلال (مثلاً "چرا شخصیت به شیوه‌ای خاص رفتار کرد؟") متغیر باشد. سیستم MRC باید متن را تجزیه و تحلیل کند تا اطلاعات مربوطه را بیابد و پاسخ مناسبی برای سوال ایجاد کند.

ارتباط MRC با QA به شرح زیر است:

1. MRC: Foundation for Question Answering به عنوان وظیفه اساسی

برای سیستم‌های QA عمل می‌کند. با آموزش مدل‌هایی برای درک و پاسخ به

سؤالات مربوط به متن، بلوک‌های سازنده سیستم‌های پیشرفته‌تر پاسخ‌گویی به سؤال را فراهم می‌کند.

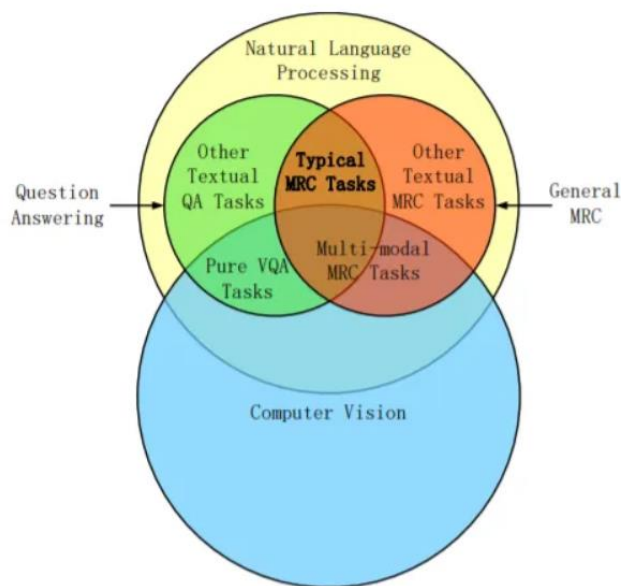
2. تکنیک‌های مشترک: بسیاری از تکنیک‌ها و الگوریتم‌های توسعه‌یافته برای درک مطلب ماشینی، مستقیماً برای وظایف QA قابل استفاده هستند. اینها شامل روش‌هایی برای نمایش متن (به عنوان مثال word embeddings, contextual embeddings)، مکانیسم‌های توجه، مدل‌های دنباله به دنباله، و تکنیک‌های یادگیری انتقال است.

3. ارزیابی: معیارهای ارزیابی و معیارهای مورد استفاده در درک مطلب ماشینی اغلب با معیارهای مورد استفاده در QA همپوشانی دارند. هر دو کار بر اساس معیارهایی مانند دقت، یادآوری، امتیاز F1 و قضاوت انسان برای ارزیابی عملکرد مدل‌ها ارزیابی می‌شوند.

4. برنامه‌های کاربردی در دنیای واقعی: MRC و QA کاربردهای متعددی در دنیای واقعی در حوزه‌های مختلف، از جمله پشتیبانی مشتری، بازیابی اطلاعات، سیستم‌های آموزشی و دستیاران مجازی دارد. سیستم‌هایی که می‌توانند به طور موثر سؤالات مربوط به متن را درک کنند و به آنها پاسخ دهند، می‌توانند برای خودکارسازی وظایف، کمک به کاربران در یافتن اطلاعات و افزایش تعامل انسان و رایانه به کار گرفته شوند.

اما وظایف QA را می‌توان به روش‌های مختلفی حل کرد مثل Retriever-Generator, Retriever-Extractor, Generator. همچنین گاهی اوقات یک کار QA با تکنیک MRC حل می‌شود که این دلیل همگرایی آنها است. اما همه راه حل‌های MRC برای حل مشکلات QA ساخته نشده‌اند. MRC توانایی درک یک متن با خواندن آن است. این توانایی را می‌توان با وظایف مختلف آزمایش کرد، که می‌

تواند در سایر مشکلات NLP نیز تامین شود. نمودار زیر از یک مقاله گرفته شده است که رابطه این دو مورد را با هم به خوبی نشان میدهد:



سوال 3:

- سوالات factoid: سؤالاتی هستند که به دنبال اطلاعات واقعی خاص یا پاسخ های مجزا هستند. این سؤالات معمولاً پاسخ های واضح و مشخصی دارند که اغلب کوتاه هستند و مستقیماً از یک زمینه یا منبع دانش قابل استخراج هستند. نمونه هایی از سوالات فاکتوئید عبارتند از:
 - پایتخت فرانسه کجاست؟
 - آلبرت انیشتین کی به دنیا آمد؟
- پاسخ به این سوالات معمولاً واحدها یا عبارات کوتاه هستند و اغلب می توان آنها را در یک جمله یا یک گزیده کوچک از متن یافت.

• سوالات non-factoid: سوالاتی هستند که در مقایسه با سوالات factoid ای پیچیده تر و ذهنی تر هستند. این سوالات ممکن است نیاز به درک، تفسیر یا استدلال عمیق تری داشته باشد تا پاسخی رضایت بخش ارائه شود. این سوالات معمولاً غیر واقعی اغلب شامل نظر، قضاوت، توضیح یا حدس و گمان هستند. نمونه هایی از سوالات غیر فاکتوئیدی عبارتند از:

○ چرا فرد رفتار خاصی داشت؟

○ مزایا و معایب انرژی های تجدیدپذیر چیست؟

• تغییر آب و هوا چگونه بر تنوع زیستی تاثیر می گذارد؟

پاسخ به این نوع سوالات می تواند چند وجهی باشد، شامل توضیح یا استدلال باشد و ممکن است یک پاسخ صحیح واحد نداشته باشد. در عوض، ممکن است نیاز به تجزیه و تحلیل، سنتز یا استنتاج بر اساس اطلاعات موجود داشته باشن..

تفاوت های کلیدی آنها به شرح زیر است:

1. ماهیت پاسخ ها: سوالات factoid ای به دنبال اطلاعات واقعی خاص با پاسخ

های واضح و مختصر هستند، در حالی که سوالات non-factoid ای ممکن است به نظرات، توضیحات یا تحلیل های ذهنی نیاز داشته باشند.

2. پیچیدگی: سوالات factoid ای معمولاً ساده تر و سرراست تر هستند و بر بازیابی اطلاعات مجزا تمرکز می کنند، در حالی که سوالات non-factoid ای می توانند پیچیده تر و بازتر باشند و به درک و تفسیر عمیق تری نیاز دارند.

3. منابع پاسخ: پاسخ به سوالات factoid ای اغلب می تواند مستقیماً از منابع متنی یا پایگاه های دانش استخراج شود، در حالی که پاسخ به سوالات non-factoid

ای ممکن است مستلزم استدلال، استنتاج یا ترکیب اطلاعات از منابع متعدد باشد.

سوال 4:

مزایای ترانسفورمرها نسبت به RNNها:

1. موازی سازی:

- ترانسفورمرها: می توانند تمام توالی داده ها را به طور همزمان پردازش کنند، که امکان موازی سازی قابل توجه و زمان های آموزشی سریع تر را فراهم می کند.

- RNNها: توالی ها را به صورت متوالی پردازش میکنند، که موازی سازی را دشوار می کند و می تواند منجر به کاهش زمان تمرین شود.

2. رسیدگی به وابستگی های دوربرد :

- ترانسفورمرها: از مکانیسم های توجه به خود استفاده میکنند که می توانند مستقیماً به هر قسمت از دنباله دسترسی داشته باشند و آنها را در گرفتن وابستگی های دوربرد مؤثرتر می کند.

- RNNها: اغلب با وابستگی های دوربرد به دلیل از بین رفتن مشکلات گرادیان و ماهیت متوالی پردازش آنها دست و پنجه نرم می کنند.

3. مقیاس پذیری:

- ترانسفورمرها: با توجه به قابلیت های پردازش موازی خود، می توانند با مجموعه داده های بزرگ و اندازه مدل ها مقیاس موثرتری داشته باشند.
- RNNها: توسط پردازش متوالی آنها محدود می شود، که مقیاس کردن آن را برای مجموعه داده های بزرگ و اندازه مدل دشوارتر می کند.

4. Context Awareness:

- ترانسفورمرها: از مکانیسم های توجه برای تمرکز بر قسمت های مرتبط توالی ورودی استفاده میکنند و آگاهی زمینه بهتری را ارائه میدهند.
- RNN ها: حالت مخفی با اندازه ثابتی دارند که ممکن است تمام اطلاعات زمینه مرتبط را به خصوص برای دنباله های طولانی ثبت نکند.

5. تفسیرپذیری مدل:

- ترانسفورمرها: مکانیسم توجه با نشان دادن اینکه مدل روی کدام قسمت های ورودی تمرکز می کند، قابلیت تفسیرپذیری را فراهم می کند.
- RNN ها: معمولاً کمتر قابل تفسیر است، زیرا تشخیص اینکه چگونه حالت های پنهان با دنباله ورودی ارتباط دارند دشوارتر است.

معایب ترانسفورمرها در مقایسه با RNN ها:

1. منابع محاسباتی:

- ترانسفورمرها: به دلیل پیچیدگی درجه دوم مکانیزم توجه به خود با توجه به طول دنباله، به منابع محاسباتی قابل توجهی، به ویژه برای مدل های بزرگ، نیاز دارند.

- RNN ها: عموماً به منابع محاسباتی کمتری نیاز دارند زیرا داده ها را به صورت متوالی پردازش می کنند و معماری ساده تری دارند.

2. استفاده از حافظه:

- ترانسفورمرها: می توانند حافظه فشرده باشند زیرا مکانیسم توجه به خود مستلزم ذخیره کل دنباله در حافظه است.

- RNN ها: معمولاً از حافظه کمتری استفاده می کنند زیرا آنها فقط نیاز به ذخیره حالت های پنهان فعلی و قبلی دارند.

3. پیچیدگی مدل:

- ترانسفورمرها: معماری پیچیده تری با لایه های متعدد و سرهای توجه دارند که می تواند اجرا و تنظیم آنها را سخت تر کند.
 - RNN ها: معماری ساده تر، که درک، پیاده سازی و تنظیم آن آسان تر است.
4. نیازمندی های داده های آموزشی:

- ترانسفورمرها: اغلب برای عملکرد خوب به حجم زیادی از داده های تمرینی نیاز دارند، مخصوصاً هنگام تمرین از ابتدا.
- RNN ها: می توانند با مجموعه داده های کوچکتر به خوبی عمل کنند، زیرا آنها به پردازش متوالی متکی هستند که می تواند از نظر داده در کارهای خاص کارآمدتر باشد.

5. Real-Time Processing:

- ترانسفورمرها: برای پردازش بلادرنگ کمتر مناسب هستند، زیرا نیاز دارند که کل دنباله قبل از انجام پیش بینی در دسترس باشد.
- RNN ها: می توانند برای پردازش بلادرنگ مناسب تر باشند زیرا می توانند در حین پردازش توالی ها، پیش بینی هایی را در لحظه ایجاد کنند.

سوال 5: رمزگذاری موقعیت یک جزء حیاتی در مدل های مبتنی بر ترانسفورمرها است که به چالش ترکیب اطلاعات ترتیب توالی در مدل می پردازد. ترانسفورمرها توالی های ورودی را به صورت موازی پردازش می کنند، برخلاف RNN هایی که آنها را به صورت متوالی پردازش می کنند. این پردازش موازی به این معنی است که ترانسفورمرها درک

ذاتی از ترتیب توکن ها در یک دنباله ندارند. رمزگذاری موقعیت برای ارائه این اطلاعات سفارش معرفی شده است.

نحوه انجام رمزگذاری موقعیت به شرح زیر است: رمزگذاری‌های موقعیتی به embedding‌های ورودی اضافه می‌شوند تا به مدل اطلاعاتی درباره موقعیت هر نشانه در دنباله بدهند. روش‌های مختلفی برای تولید رمزگذاری‌های موقعیتی وجود دارد، اما یک رویکرد رایج استفاده از توابع سینوسی و کسینوسی فرکانس‌های مختلف است:

- For even indices (i.e., $2i$):

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

- For odd indices (i.e., $2i+1$):

$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

where:

- pos is the position of the token in the sequence.
- i is the dimension of the embedding.
- d is the dimensionality of the model's embeddings.

اهمیت رمزگذاری موقعیت به شرح زیر است:

1. شامل اطلاعات متوالی بودن: از آنجایی که ترانسفورمرها همه نشانه‌ها را به طور همزمان پردازش می‌کنند، رمزگذاری‌های موقعیتی به مدل کمک می‌کند تا ترتیب توکن‌ها را درک کنند، که برای کارهایی مانند مدل‌سازی زبان، ترجمه و موارد دیگر ضروری است.
2. یادگیری موقعیت‌های نسبی را فعال می‌کند: توابع سینوسی و کسینوسی تضمین می‌کنند که رمزگذاری‌های موقعیتی مقادیر منحصر به فردی برای هر موقعیت

دارند و ویژگی های مفیدی مانند خطی بودن را نشان می دهند که به مدل اجازه می دهد موقعیت های نسبی توکن ها را به طور موثر یاد بگیرد.

3. Smooth Transitions: ماهیت پیوسته توابع سینوس و کسینوس Smooth Transitions بین موقعیت ها را فراهم می کند و به مدل اجازه می دهد در طول های دنباله های مختلف تعمیم بهتری داشته باشد.

4. سازگاری با Self-Attention: رمزگذاری های موقعیتی مستقیماً به embedding های ورودی اضافه می شوند و آنها را به طور طبیعی با مکانیسم خودتوجهی در ترانسفورمرها سازگار می کند، که امتیازات توجه را بر اساس این نمایش های ترکیبی محاسبه می کند.

سوال 6:

1. Only-Encoders: معماری آنها صرفاً از مجموعه ای از لایه های رمزگذار تشکیل شده است. هر لایه رمزگذار معمولاً شامل مکانیسم های خودتوجهی و شبکه های عصبی پیش خور است. که اغلب برای کارهایی استفاده می شوند که درک و پردازش ورودی به عنوان یک کل مهم است. وظایف رایج عبارتند از:
 - طبقه بندی متن: تخصیص برچسب به متن، مانند تجزیه و تحلیل احساسات.
 - Named Entity Recognition (NER): شناسایی موجودیت ها در متن.
 - مدل سازی زبان: ایجاد نمایش هایی برای متن که می تواند برای کارهای مختلف پایین دست استفاده شود.
- ویژگی های کلیدی آنها عبارتند از:

- کل دنباله ورودی را به طور همزمان پردازش می کنند.
 - از توجه به خود برای گرفتن وابستگی بین همه نشانه ها استفاده می کنند.
 - نمایش های متنی از نشانه های ورودی را خروجی می دهند.
2. Only-Decoders: معماری آنها صرفاً از مجموعه ای از لایه های رمزگشا تشکیل شده است. هر لایه رمزگشا شامل مکانیسم های توجه به خود، مکانیسم های توجه رمزگذار-رمزگشا (در صورت وجود خروجی های رمزگذار) و شبکه های عصبی پیش خور است. که عمده تاً برای کارهای autoregressive استفاده می شوند که در آن مدل بر اساس توکن های قبلی هر بار یک توکن تولید می کند. وظایف رایج عبارتند از:
- تولید متن: ایجاد توالی متن، مانند مدل سازی زبان (سبک GPT).
 - ترجمه ماشینی: ترجمه متن از یک زبان به زبان دیگر زمانی که مدل به صورت ترتیب به ترتیب آموزش داده شود.
- ویژگی های کلیدی آنها عبارتند از:
- توالی های خروجی توکن به نشانه را به صورت خودرگرسیون تولید می کنند.
 - از توجه به خود برای در نظر گرفتن توکن های تولید شده قبلی استفاده می کنند.
 - خروجی نشانه بعدی در دنباله بر اساس زمینه فعلی.
3. رمزگذار-رمزگشا (ترانسفورمرها): معماری آنها که به سادگی به عنوان مدل ترانسفورمر نیز شناخته می شود، از دو بخش تشکیل شده است: پشته رمزگذار و پشته رمزگشا. رمزگذار توالی ورودی را پردازش می کنند تا نمایش های متنی ایجاد کنند، که سپس برای تولید دنباله خروجی به رمزگشا وارد می شوند. که

برای کارهایی استفاده می‌شوند که نیاز به تبدیل یک دنباله ورودی به یک دنباله خروجی دارند. وظایف رایج عبارتند از:

- ترجمه ماشینی: ترجمه متن از یک زبان به زبان دیگر.
- خلاصه: ایجاد خلاصه ای مختصر از یک متن داده شده.
- Sequence-to-Sequence Tasks: هر کاری که در آن یک توالی ورودی به دنباله خروجی متفاوتی تبدیل می شود، مانند پاسخگویی به سؤال و تولید متن به متن.

ویژگی های کلیدی آنها عبارتند از:

- Encoder: کل توالی ورودی را پردازش می کند و اطلاعات متنی را با استفاده از لایه های خودتوجهی و فید فوروارد ضبط می کند.
- رمزگشا: توالی خروجی را در یک زمان، با استفاده از توجه به خود، توجه رمزگذار-رمزگشا (برای ترکیب خروجی های رمزگذار)، و لایه های پیشخور ایجاد می کند.
- رمزگذار دو جهته: زمینه را از هر دو جهت (چپ به راست و راست به چپ) در دنباله ورودی ضبط می کند.
- رمزگشای یک جهته: توکن ها را به صورت چپ به راست تولید می کند و از تولید خود رگرسیون اطمینان می دهد.

سوال 7:

1. Extractive QA: این سیستم ها گستره ای از متن را مستقیماً از یک سند معین که به سؤال پاسخ می دهد، شناسایی و استخراج می کنند. پاسخ زیر رشته ای از متن اصلی است. که نحوه عملکرد آنها به شرح زیر است:

- ورودی: یک سوال و یک قسمت از متن (زمینه).
- پردازش: مدل متن را پردازش می کند تا موقعیت شروع و پایان دامنه پاسخ را شناسایی کند.
- خروجی: گزیده ای مستقیم از متن که به سوال پاسخ می دهد.

این سیستم ها اغلب پاسخ های بسیار دقیقی را ارائه می دهند زیرا متن دقیق را از منبع بیرون می کشند. اما پاسخ آنها باید کلمه به کلمه در متن وجود داشته باشد. نمی توانند محتوای جدید تولید کنند یا بازنویسی کنند پس به شدت به حضور پاسخ دقیق در متن ارائه شده متکی هستند.

2. Abstractive QA: این سیستم ها متن جدیدی تولید می کنند که به سؤال پاسخ می دهد و احتمالاً از کلمات و عباراتی استفاده می کند که در متن اصلی وجود ندارد. این رویکرد شبیه خلاصه سازی یا نقل قول است. که نحوه عملکرد آنها به شرح زیر است:

- ورودی: یک سوال و یک قسمت از متن (زمینه).
 - پردازش: مدل زمینه را درک می کند و یک پاسخ منسجم و مرتبط ایجاد می کند.
 - خروجی: پاسخی که به تازگی ایجاد شده است که ممکن است اطلاعات را از زمینه ترکیب، بازفراز یا ترکیب کند.
- این سیستم می تواند پاسخ هایی ارائه دهد حتی اگر متن دقیق در متن موجود نباشد، برای ایجاد خلاصه یا پاسخ های بازنویسی مفید است. پس قادر به تولید پاسخ های طبیعی تر و متناسب با زمینه است. هرچند برای پیاده سازی و آموزش پیچیده تر است

که به قابلیت های پیچیده تولید زبان نیاز دارد و پاسخ های ایجاد شده ممکن است خطاها یا اطلاعات توهم آمیزی را ایجاد کند که در متن منبع وجود ندارد.

تفاوت های کلیدی این دو سیستم به شرح زیر هستند:

1. منبع پاسخ:

- استخراجی: پاسخ ها مستقیماً از متن ارائه شده استخراج می شوند.
- انتزاعی: پاسخ ها تولید می شوند و ممکن است مستقیماً در متن ظاهر نشوند.

2. طبیعی بودن:

- استخراجی: پاسخ ها می توانند گزیده ای دقیق باشند که ممکن است در مکالمه کمتر طبیعی باشد.
- انتزاعی: پاسخ ها می توانند به صورت طبیعی تر و منسجم تر بیان شوند.

3. اجرا:

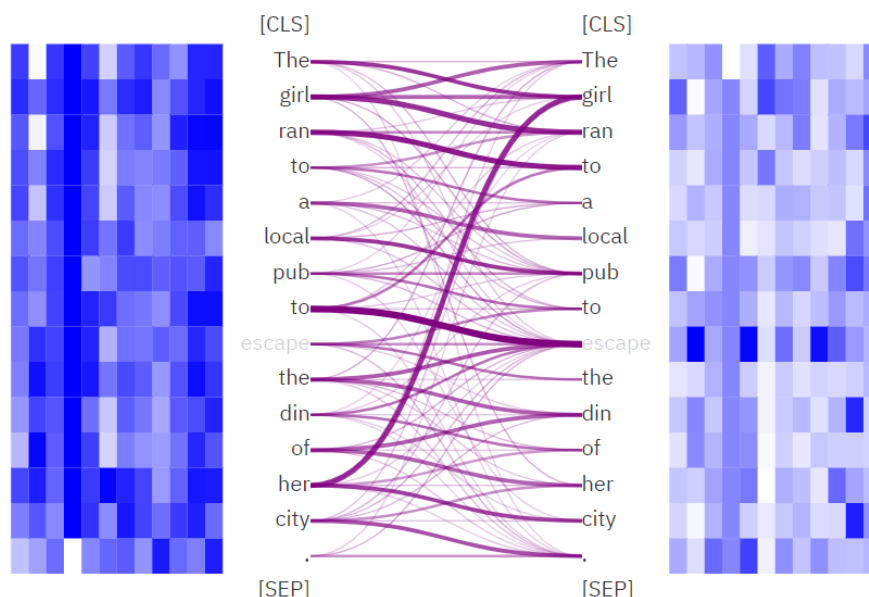
- استخراجی: بر تشخیص دهانه در متن تمرکز می کند.
- انتزاعی: شامل درک و تولید متن جدید است که به تکنیک های پیشرفته تولید زبان طبیعی نیاز دارد.

4. موارد استفاده:

- استخراجی: ایده آل برای کاربردهایی که استخراج دقیق و دقیق بسیار مهم است، مانند اسناد قانونی یا دستورالعمل های فنی.
- انتزاعی: مناسب برای برنامه های بیشتر محاوره ای یا مبتنی بر خلاصه، مانند دستیارهای مجازی یا ابزارهای خلاصه سازی.

Input: The quick brown fox jumps over the lazy dog.

Task: Named Entity Recognition (NER)



BERT از مکانیزم multi-head self-attention استفاده می کند، به این معنی که در هر لایه دارای multiple attention heads است. هر attention head یاد می گیرد که روی قسمت های مختلف توالی ورودی تمرکز کند. در اینجا دلیل فایده داشتن آنها این است:

1. کشف روابط متنوع: هر attention head می تواند یاد بگیرد که انواع مختلفی از روابط بین نشانه ها را ثبت کند. به عنوان مثال، یک head ممکن است بر وابستگی های نحوی (مثلاً روابط فاعل-فعل) تمرکز کند، در حالی که دیگری ممکن است بر روابط معنایی تمرکز کند (مثلاً جفت شدن اسم و صفت).
2. درک زمینه قوی: با داشتن headهای متعدد، مدل می تواند اطلاعات را از دیدگاه های مختلف جمع آوری کند، که منجر به درک قوی تری از زمینه می

شود. این به ویژه برای کارهای پیچیده که درک زمینه های ظریف بسیار مهم است، مهم است.

3. ابهام زدایی از کلمات مبهم: کلمات با معانی چندگانه (مثلاً "بانک" می تواند به معنای یک موسسه مالی یا کنار رودخانه باشد) را می توان به طور مؤثرتر ابهام زد. head های مختلف می توانند بر روی زمینه های مختلف تمرکز کنند تا معنای مورد نظر را بر اساس کلمات اطراف درک کنند.

4. تعمیم بهبودیافته: head های توجه متعدد، مدل را قادر می سازد تا با یادگیری جنبه های مختلف زبان، داده های دیده نشده را بهتر تعمیم دهد. این تنوع خطر تطابق بیش از حد با الگوهای خاص در داده های آموزشی را کاهش می دهد.

5. تفسیرپذیری پیشرفته: head های متعدد بینش بیشتری در مورد آنچه مدل در حال یادگیری است ارائه می دهد. با تجزیه و تحلیل الگوهای توجه head های مختلف، محققان و متخصصان می توانند درک بهتری از نحوه پردازش و اولویت بندی اطلاعات توسط مدل به دست آورند.

سوال 9: کد خواسته شده کامل شد و در پارت های مورد نیاز، کامنت گذاشته شد.