به نام خدا

تمرین سری ششم

درس مبانی پردازش زبان و گفتار

سینا علی‌نژاد

۹۹۵۲۱۴۶۹

سوال ۱-

سناریو ۱: وزن‌های اولیه با مدل BERT از قبل آموزش دیده شده

در این سناریو، شما از وزن‌های یک مدل BERT که قبلاً بر روی مجموعه‌ای از تسک‌ها آموزش دیده است، به عنوان وزن‌های اولیه استفاده می‌کنید. این فرآیند معمولاً به عنوان **Fine-Tuning** شناخته می‌شود.

فرآیند آموزش

- **شروع بهینه‌تر** :مدل با وزن‌های اولیه‌ای شروع می‌کند که از قبل اطلاعات و الگوهای زبان طبیعی را یاد گرفته‌اند. این به مدل کمک می‌کند که با داده‌های جدید سریع‌تر سازگار شود.
- **نیاز به داده کمتر** :به دلیل داشتن وزن‌های از پیش آموزش دیده، مدل نیاز کمتری به داده‌های جدید برای یادگیری الگوهای پایه‌ای زبان دارد.
- **زمان آموزش کمتر** :به دلیل شروع از یک نقطه بهینه‌تر، فرآیند آموزش معمولاً سریع‌تر انجام می‌شود و به تعداد کمتری از دوره‌های آموزشی (epochs) نیاز دارد.

عملکرد پس از آموزش

- **عملکرد بهتر** :مدل به طور کلی عملکرد بهتری خواهد داشت، زیرا از ابتدا با وزن‌های بهینه‌تری شروع کرده که از قبل بر روی تسک‌های مشابه آموزش دیده‌اند.
- **تعمیم بهتر** :مدل توانایی بهتری در تعمیم دادن به داده‌های جدید و نادیده پیدا می‌کند، زیرا الگوهای زبان عمومی را از قبل یاد گرفته است.

سناریو ۲: وزن‌های اولیه تصادفی

در این سناریو، شما از وزن‌های اولیه تصادفی استفاده می‌کنید و مدل را از ابتدا آموزش می‌دهید. این فرآیند به عنوان **Training from Scratch** شناخته میشود.

فرآیند آموزش

- **شروع ضعیف‌تر** :مدل با وزن‌های تصادفی شروع می‌کند که هیچ اطلاعی از الگوهای زبان طبیعی ندارند.
- **نیاز به داده بیشتر** :مدل نیاز دارد که تمامی الگوهای زبان را از ابتدا یاد بگیرد، بنابراین به داده‌های بیشتری برای آموزش نیاز دارد.
- **زمان آموزش بیشتر** :به دلیل شروع از وزن‌های تصادفی، فرآیند آموزش طولانی‌تر خواهد بود و به تعداد بیشتری از دوره‌های آموزشی نیاز دارد.

عملکرد پس از آموزش

- **عملکرد ضعیف‌تر** :مدل به احتمال زیاد عملکرد ضعیف‌تری نسبت به مدلی که از وزن‌های از پیش آموزش دیده استفاده می‌کند خواهد داشت، به خصوص اگر داده‌های آموزشی محدود باشد.

- **تعمیم ضعیف‌تر** :مدل ممکن است در تعمیم دادن به داده‌های جدید و نادیده ضعیف‌تر عمل کند، زیرا الگوهای زبان عمومی را به خوبی یاد نگرفته است.

نتیجه‌گیری

به طور کلی، استفاده از وزن‌های یک مدل BERT از قبل آموزش دیده، به مدل کمک می‌کند تا سریع‌تر و بهتر آموزش ببیند و عملکرد بهتری داشته باشد. در مقابل، آموزش مدل از ابتدا با وزن‌های تصادفی نیاز به داده و زمان بیشتری دارد و معمولاً عملکرد ضعیف‌تری خواهد داشت.

سوال ۲-

چالش Forgetting Catastrophic در فرآیند Fine-Tuning

Forgetting Catastrophic یا فراموشی فاجعه‌بار، یک چالش مهم در فرآیند آموزش مدل‌های شبکه عصبی، به ویژه در فرآیند Fine-Tuning است. این پدیده زمانی رخ می‌دهد که یک مدل شبکه عصبی که قبلاً بر روی یک مجموعه داده آموزش دیده است، با آموزش مجدد بر روی یک مجموعه داده جدید، اطلاعات و دانش قبلی خود را از دست می‌دهد. به عبارت دیگر، مدل توانایی خود را در انجام وظایف اولیه‌ای که بر روی آن‌ها آموزش دیده بود، از دست می‌دهد.

توضیح چالش

مکانیزم Forgetting Catastrophic

- **تغییر وزن‌ها** :در فرآیند Fine-Tuning ، وزن‌های شبکه عصبی به منظور بهینه‌سازی عملکرد مدل بر روی داده‌های جدید تغییر می‌کنند. این تغییرات می‌توانند منجر به از دست رفتن اطلاعات و الگوهای اولیه‌ای شوند که مدل قبلاً یاد گرفته بود.
- **ظرفیت محدود مدل**: مدل های شبکه عصبی ظرفیت محدودی دارند و نمی‌توانند تمامی اطلاعات و الگوهای مربوط به تسک‌های مختلف را به طور همزمان نگه دارند. این محدودیت ظرفیت باعث می‌شود که با اضافه شدن داده‌های جدید، مدل اطلاعات قدیمی را فراموش کند.

روش‌های کاهش و رفع Forgetting Catastrophic

برای کاهش یا رفع چالشForgetting Catastrophic ، رویکردهای مختلفی وجود دارد که در ادامه به چند مورد از آن‌ها اشاره می‌کنیم:

۱. Regularization-Based Methods

روش‌های مبتنی بر منظم‌سازی تلاش می‌کنند تا تغییرات وزن‌ها را در طول فرآیند Fine-Tuning محدود کنند تا مدل اطلاعات قدیمی را فراموش نکند.

Elastic Weight Consolidation (EWC)

این روش یک نوع منظم‌سازی است که تغییرات وزن‌های مهم را محدود می‌کند. ایده اصلی این روش این است که وزن‌هایی که برای تسک اولیه حیاتی هستند، نباید به مقدار زیادی تغییر کنند.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., ... & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 3521-3526.

۲. Rehearsal Methods

روش‌های تمرینی از داده‌های قدیمی در طول آموزش بر روی داده‌های جدید استفاده می‌کنند تا مدل اطلاعات قبلی را حفظ کند.

Joint Training

در این روش، داده‌های قدیمی و جدید به صورت همزمان برای آموزش مدل استفاده می‌شوند. این کار باعث می‌شود که مدل هم اطلاعات قبلی و هم اطلاعات جدید را به خوبی یاد بگیرد.

```
Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2), 123-146.
```

۳. Dynamic Architectures

روش‌های مبتنی بر معماری‌های پویا، معماری مدل را در طول زمان تغییر می‌دهند تا بتوانند اطلاعات جدید را بدون از دست دادن اطلاعات قدیمی یاد بگیرند.

Progressive Neural Networks

این روش با اضافه کردن شبکه‌های جدید برای یادگیری تسک‌های جدید، از فراموشی اطلاعات قدیمی جلوگیری می‌کند.

شبکه‌های جدید می‌توانند از دانش شبکه‌های قدیمی استفاده کنند.

```
Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., ... & Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.
```

۴. Memory-Based Methods

روش‌های مبتنی بر حافظه از یک حافظه خارجی برای ذخیره و بازیابی اطلاعات قدیمی استفاده می‌کنند.

Memory Networks

این روش‌ها از یک حافظه خارجی برای ذخیره اطلاعات مهم از تسک‌های قبلی استفاده می‌کنند و در زمان نیاز به آن‌ها دسترسی پیدا می‌کنند.

```
Graves, A., Wayne, G., & Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.
```

نتیجه‌گیری

Forgetting Catastrophicیک چالش جدی در فرآیند Fine-Tuning مدل‌های شبکه عصبی است که می‌تواند منجر به از دست رفتن اطلاعات و دانش قبلی مدل شود. با استفاده از روش‌های مختلف مانند Regularization-Based Methods، Rehearsal Methods، Dynamic Architectures، و Memory-Based Methods می‌توان این مشکل را تا حدودی رفع کرد.

سوال ۳-

انتقال یادگیری (Transfer Learning) و تفاوت آن با تنظیم دقیق(Fine-Tuning)

انتقال یادگیری (Transfer Learning) و **تنظیم دقیق (Fine-Tuning)** دو رویکرد مهم در آموزش مدل‌های یادگیری ماشین هستند که از اطلاعات و دانش مدل‌های قبلی برای بهبود عملکرد مدل‌های جدید استفاده می‌کنند. در ادامه به توضیح هر یک از این مفاهیم، تفاوت‌های آن‌ها و شرایط استفاده از هر یک می‌پردازیم.

انتقال یادگیری(Transfer Learning)

انتقال یادگیری به فرایندی اشاره دارد که در آن، یک مدل که بر روی یک تسک خاص آموزش دیده است، برای یک تسک دیگر استفاده می‌شود. ایده اصلی این است که دانش کسب شده توسط مدل در یک حوزه می‌تواند به بهبود عملکرد مدل در حوزه‌ای دیگر کمک کند.

کاربردها و شرایط استفاده

- **داده‌های آموزشی محدود** :زمانی که داده‌های آموزشی برای تسک هدف محدود است، می‌توان از مدل‌های از پیش آموزش دیده در یک تسک مشابه استفاده کرد.
- **آموزش سریع‌تر** :انتقال یادگیری می‌تواند زمان آموزش مدل را کاهش دهد، زیرا مدل از وزن‌های از پیش آموزش دیده استفاده می‌کند.
- **کاهش هزینه‌های محاسباتی** :استفاده از مدل‌های از پیش آموزش دیده می‌تواند هزینه‌های محاسباتی را کاهش دهد.

مثال

فرض کنید یک مدل شبکه عصبی برای تشخیص اشیاء در تصاویر آموزش دیده است. این مدل می‌تواند برای یک تسک جدید مانند تشخیص انواع خاصی از حیوانات در تصاویر استفاده شود.

```
Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on k
nowledge and data engineering*, 22(10), 1345-1359.
```

تنظیم دقیق(Fine-Tuning)

تنظیم دقیق به فرایندی اشاره دارد که در آن، یک مدل از پیش آموزش دیده بر روی داده‌های جدید و برای یک تسک خاص دوباره آموزش داده می‌شود. در این فرایند، وزن‌های مدل از پیش آموزش دیده به عنوان نقطه شروع استفاده می‌شوند و سپس مدل بر روی داده‌های جدید تنظیم می‌شود.

کاربردها و شرایط استفاده

- **تسک‌های مشابه** :زمانی که تسک هدف بسیار مشابه با تسک اصلی است که مدل بر روی آن آموزش دیده، تنظیم دقیق می‌تواند بسیار مؤثر باشد.
- **بهبود عملکرد** :تنظیم دقیق می‌تواند به بهبود عملکرد مدل در تسک هدف کمک کند، زیرا مدل از اطلاعات و دانش قبلی خود استفاده می‌کند.
- **تطبیق با داده‌های خاص** :تنظیم دقیق به مدل کمک می‌کند تا بهتر با داده‌های خاص و ویژگی‌های منحصر به فرد تسک هدف سازگار شود.

مثال

فرض کنید یک مدل BERT بر روی یک مجموعه داده بزرگ از متن‌های عمومی آموزش دیده است. این مدل می‌تواند با تنظیم دقیق بر روی یک مجموعه داده خاص مانند نقدهای فیلم بهبود یابد.

```
Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classif
ication. *arXiv preprint arXiv:1801.06146*.
```

تفاوت‌ها

ویژگی	انتقال یادگیری (Transfer Learning)	تنظیم دقیق(Fine-Tuning)
تعریف	استفاده از یک مدل آموزش دیده برای یک تسک جدید	تنظیم و آموزش مجدد یک مدل از پیش آموزش دیده بر روی داده‌های جدید

ویژگی	انتقال یادگیری (Transfer Learning)	تنظیم دقیق(Fine-Tuning)
میزان تغییرات	معمولاً کمتر (استفاده از مدل به صورت آماده)	تغییرات بیشتر (آموزش مجدد مدل)
کاربرد	تسک‌های متفاوت ولی مرتبط	تسک‌های مشابه یا مرتبط بسیار نزدیک
زمان آموزش	معمولاً کمتر	معمولاً بیشتر
نیاز به داده	نیاز به داده کمتر	نیاز به داده بیشتر (برای تنظیم دقیق)

نتیجه‌گیری

انتقال یادگیری و **تنظیم دقیق** هر دو ابزارهای قدرتمندی برای بهبود عملکرد مدل‌ها با استفاده از دانش قبلی هستند. انتقال یادگیری بیشتر برای زمانی مناسب است که تسک هدف کاملاً جدید ولی مرتبط باشد و داده‌های آموزشی محدود باشند. از طرف دیگر، تنظیم دقیق برای زمانی مناسب است که تسک هدف بسیار مشابه با تسک اصلی باشد و نیاز به بهبود دقیق‌تر و تطبیق با داده‌های خاص داریم.

سوال ۴-

تأثیرات روش‌های مختلف Masking و تعیین میزان توکن‌های قابل Mask بر روی فرآیند آموزش و عملکرد مدل‌هایMLMs

Masked Language Models (MLMs) مانند BERT از روش‌های مختلف **masking** استفاده می‌کنند تا مدل‌ها بتوانند به صورت موثرتر زبان طبیعی را یاد بگیرند. در این مدل‌ها، بخشی از توکن‌ها در ورودی با یک توکن خاص مانند `[MASK]` جایگزین می‌شوند و مدل باید با استفاده از محتوای (context) باقی‌مانده، توکن‌های ماسک شده را پیش‌بینی کند. در اینجا به بررسی تاثیرات روش‌های مختلف masking و تعیین میزان توکن‌های قابل mask بر فرآیند آموزش و عملکرد مدل‌های MLMs می‌پردازیم.

روش‌های مختلفMasking

۱. **روش رندوم (Random Masking)**

در این روش، توکن‌ها به صورت تصادفی برای ماسک شدن انتخاب می‌شوند.

تأثیرات:

- **تنوع ورودی‌ها** :این روش باعث می‌شود که مدل با انواع مختلفی از بافت‌ها (contexts) مواجه شود، که می‌تواند به بهبود تعمیم‌دهی مدل کمک کند.
- **پوشش گسترده‌تر** :چون توکن‌ها به صورت تصادفی انتخاب می‌شوند، تقریباً تمامی توکن‌ها در طی فرآیند آموزش ممکن است ماسک شوند، که این امر به مدل کمک می‌کند تا اطلاعات بیشتری را از داده‌ها استخراج کند.

۲. **روش مبتنی بر Part of Speech (POS-Based Masking)**

در این روش، توکن‌ها بر اساس نوع دستوری (مانند فعل‌ها، اسم‌ها و صفت‌ها) برای ماسک شدن انتخاب می‌شوند. به عنوان مثال، می‌توان فقط فعل‌ها یا فقط اسم‌ها را ماسک کرد.

تأثیرات:

- **متمرکز بر نقش‌های دستوری** :این روش می‌تواند به مدل کمک کند تا درک عمیق‌تری از نقش‌های دستوری مختلف در جمله پیدا کند.
- **ارتقاء عملکرد در تسک‌های خاص** :مدل‌هایی که با استفاده از این روش آموزش دیده‌اند ممکن است در تسک‌های خاصی مانند تجزیه و تحلیل نحوی یا برچسب‌گذاری دستوری عملکرد بهتری داشته باشند.

تعیین میزان توکن‌های قابل Mask

میزان توکن‌هایی که در هر جمله ماسک می‌شوند نیز تأثیر قابل توجهی بر عملکرد مدل دارد. در ادامه به بررسی تأثیرات مقادیر مختلف این پارامتر می‌پردازیم.

۱. **درصد پایین توکن های Mask (مثلا ۱۰ درصد)**

- **حفظ اطلاعات بیشتر** :مقدار بیشتری از جمله در دسترس مدل باقی می‌ماند که می‌تواند به پیش‌بینی دقیق‌تر توکن‌های ماسک شده کمک کند.
- **آموزش کندتر** :چون فقط بخشی از توکن‌ها ماسک می‌شوند، مدل ممکن است نیاز به دوره‌های آموزشی طولانی‌تر داشته باشد تا بتواند به طور کامل زبان را یاد بگیرد.

۲. **درصد بالای توکن های Mask (مثلا ۵۰ درصد)**

- **افزایش چالش** :ماسک کردن تعداد بیشتری از توکن‌ها، مدل را مجبور می‌کند که با اطلاعات کمتر، توکن‌های ماسک شده را پیش‌بینی کند. این امر می‌تواند به بهبود تعمیم‌دهی مدل کمک کند.
- **ریسک کاهش دقت** :اگر تعداد توکن‌های ماسک شده بیش از حد زیاد باشد، مدل ممکن است نتواند به درستی از بافت باقی‌مانده استفاده کند که می‌تواند به کاهش دقت منجر شود.

نتیجه‌گیری

انتخاب روش masking و تعیین میزان توکن‌های ماسک شده تأثیرات قابل توجهی بر روی فرآیند آموزش و عملکرد مدل‌های MLMs دارد. روش‌های مختلف masking مانند روش رندوم و روش مبتنی بر POS هر کدام مزایا و معایب خاص خود را دارند و بسته به تسک مورد نظر، می‌توان از هر یک استفاده کرد. همچنین، تعیین میزان توکن‌های قابل mask باید به گونه‌ای باشد که تعادلی بین چالش‌برانگیز بودن و قابل یادگیری بودن ایجاد کند. انتخاب‌های هوشمندانه در این زمینه می‌تواند به بهبود قابل توجهی در عملکرد مدل‌های MLMs منجر شود.

منابع

- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa:

سوال ۵-

مقایسه عملکرد معماری‌های Seq2Seq, MLM وCLM

معماری‌های **Seq2Seq**, **MLM** و **CLM** هر کدام برای اهداف خاصی در پردازش زبان طبیعی (NLP) طراحی شده‌اند و هر یک مزایا و معایب خاص خود را دارند. در ادامه به مقایسه این سه معماری، مزایا و معایب هر یک و مثال‌هایی برای هر کدام می‌پردازیم.

۱. Seq2Seq (Sequence to Sequence)

Seq2Seq یک معماری است که یک دنباله ورودی را به یک دنباله خروجی تبدیل می‌کند. این معماری معمولاً شامل دو بخش اصلی است **Encoder** و **Decoder**

مزایا:

- **انعطاف‌پذیری بالا** :می‌تواند برای انواع مختلفی از تسک‌ها مانند ترجمه ماشینی، خلاصه‌سازی متن، و تولید پاسخ به سوالات استفاده شود.
- **پردازش متنی پیچیده** :توانایی پردازش و تولید دنباله‌های متنی پیچیده را دارد.

معایب:

- **نیاز به داده‌های زیاد** :برای عملکرد بهینه به مجموعه داده‌های بزرگ نیاز دارد.
- **زمان و منابع محاسباتی بالا** :آموزش این مدل‌ها به زمان و منابع محاسباتی زیادی نیاز دارد.

مثال‌ها:

- **ترجمه ماشینی** :مدل‌های ترجمه ماشینی مانند Google Translate از معماری Seq2Seq استفاده می‌کنند.
- **چت‌بات‌ها** :برخی از چت‌بات‌ها از معماری Seq2Seq برای تولید پاسخ‌های متنی استفاده می‌کنند.

```
Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with ne
ural networks. *arXiv preprint arXiv:1409.3215*.
```

۲. MLM (Masked Language Model)

MLM یک معماری است که بخشی از توکن‌های ورودی را به طور تصادفی ماسک می‌کند و مدل باید این توکن‌های ماسک شده را پیش‌بینی کند. این روش در مدل‌هایی مانند BERT استفاده می‌شود.

مزایا:

- **درک عمیق از بافت** :با پیش‌بینی توکن‌های ماسک شده، مدل می‌تواند درک عمیق‌تری از بافت متنی پیدا کند.
- **عملکرد خوب در تسک‌های درک زبان** : مدل‌های MLM در تسک‌هایی مانند طبقه‌بندی متن، پاسخ به سوالات و تجزیه و تحلیل احساسات عملکرد بسیار خوبی دارند.

معایب:

- **عدم توانایی در تولید متن** :مدل‌های MLM برای تولید متن مناسب نیستند و بیشتر برای درک زبان طراحی شده‌اند.
- **نیاز به داده‌های پیش‌پردازش شده** :پیش‌پردازش داده‌ها برای ماسک کردن توکن‌ها می‌تواند پیچیده باشد.

مثال‌ها:

- **BERT** : یک مدل معروف MLM که در بسیاری از تسک‌های NLP استفاده می‌شود.
- **RoBERTa** : یک نسخه بهینه‌شده از BERT که از همان معماری استفاده می‌کند.

```
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
```

۳. CLM (Causal Language Model)

CLM یا مدل‌های زبانی علی، دنباله‌های متنی را به صورت ترتیبی پیش‌بینی می‌کنند. این مدل‌ها از پیش‌بینی توکن بعدی در یک دنباله استفاده می‌کنند و معمولاً در مدل‌هایی مانند GPT استفاده می‌شوند.

مزایا:

- **تولید متن روان** : مدل‌های CLM در تولید متن روان و طبیعی عملکرد بسیار خوبی دارند.
- **کاربرد گسترده** : می‌توانند در تسک‌های تولید متن، تکمیل متن و پاسخ به سوالات استفاده شوند.

معایب:

- **محدودیت در درک بافت بلندمدت** : ممکن است در درک بافت بلندمدت نسبت به مدل‌های MLM ضعیف‌تر باشند.
- **نیاز به داده‌های ترتیب‌دار**: برای عملکرد بهینه به داده‌های ترتیب‌دار نیاز دارند.

مثال‌ها:

- **GPT-3:** یک مدل زبان بزرگ که از معماری CLM استفاده می‌کند و کاربردهای گسترده‌ای دارد.
- **OpenAI Codex:** مدل زبان برنامه‌نویسی که از معماری CLM بهره می‌برد.

```
Radford, A., Narasimhan, K., Salimans, T., & S
```

سوال ۶-

مدل‌های (Masked Language Models) **MLM** مانند BERT به صورت اصلی برای درک زبان و پیش‌بینی توکن‌های ماسک شده طراحی شده‌اند و نه برای تولید متن. با این حال، می‌توان با استفاده از برخی روش‌ها و تکنیک‌ها از مدل‌های MLM برای تولید دنباله‌ای از متن استفاده کرد. در ادامه به بررسی این روش‌ها می‌پردازیم.

روش‌های تولید متن با استفاده از مدل‌هایMLM

۱. روش Iterative Masking and Infilling

در این روش، یک دنباله ابتدایی از متن داریم و به طور مکرر بخشی از متن را ماسک کرده و مدل را برای پیش‌بینی توکن‌های ماسک شده استفاده می‌کنیم. این فرآیند تا زمانی که متن به طور کامل تولید شود، ادامه می‌یابد.

مراحل:

۱. **ابتدا یک دنباله ابتدایی (seed) داشته باشید** :این دنباله می‌تواند یک جمله یا بخشی از جمله باشد.
۲. **چند توکن را به طور تصادفی ماسک کنید** :برخی از توکن‌ها را با [MASK]جایگزین کنید.
۳. **پیش‌بینی توکن‌ها** :مدل MLM را برای پیش‌بینی توکن‌های ماسک شده استفاده کنید.
٤. **جایگزینی توکن‌های پیش‌بینی شده** : توکن‌های پیش‌بینی شده را در مکان‌های ماسک شده قرار دهید.

۵. **تکرار مراحل ۳ تا ٤** :این فرآیند را تکرار کنید تا دنباله متن به طور کامل تولید شود یا به طول مطلوب برسد.

مزایا:

- **کنترل بیشتر بر تولید متن** :می‌توانید به صورت تدریجی متن را تولید کنید و در هر مرحله پیش‌بینی مدل را بررسی و کنترل کنید.

معایب:

- **پیچیدگی بیشتر** :نیاز به تکرار مراحل و تنظیم دقیق تعداد توکن‌های ماسک شده در هر مرحله دارد.
- **زمان‌بر بودن** :این روش نسبت به مدل‌های CLM زمان بیشتری برای تولید متن نیاز دارد.

۲. روش ترکیبی (hybrid)

این روش شامل ترکیب یک مدل MLM با یک مدل تولید متن مانند GPT است. مدل MLM برای بهبود درک بافت و مدل تولید متن برای تولید دنباله‌های جدید استفاده می‌شود.

مراحل:

۱. **استفاده از مدل MLM برای درک بافت** :مدل MLM می‌تواند برای بهبود درک بافت و اصلاح دنباله‌های متنی کمک کند.

۲. **استفاده از مدل تولید متن (CLM):** مدل تولید متن مانند GPT برای تولید دنباله‌های جدید استفاده می‌شود.

مثال:

- **BERT-GPT:** ترکیب BERT برای درک بافت و GPT برای تولید متن.

مزایا:

- **بهترین‌های هر دو جهان** : بهره‌گیری از قدرت درک بافت مدل MLM و توانایی تولید متن مدل CLM

معایب:

- **پیچیدگی پیاده‌سازی** :نیاز به تنظیم دقیق و هماهنگی بین دو مدل مختلف دارد.
- **نیاز به منابع بیشتر** :این روش به منابع محاسباتی بیشتری نیاز دارد.

نتیجه‌گیری

مدل‌های **MLM**به طور اصلی برای تولید متن طراحی نشده‌اند، اما با استفاده از روش‌هایی مانند **Iterative Masking and Infilling**و **روش Hybrid** می‌توان از این مدل‌ها برای تولید دنباله‌های متنی استفاده کرد. این روش‌ها نیاز به تنظیم دقیق و پیاده‌سازی پیچیده‌تری دارند، اما می‌توانند نتایج قابل قبولی را ارائه دهند. با این حال، در مواردی که تولید متن به صورت مستقیم و سریع نیاز است، استفاده از مدل‌های **CLM** مانند GPT معمولاً انتخاب بهتری است.

سوال ۷-

Question about MLM strategy:

80% Masked with [MASK] Token:

The primary reason for replacing 80% of the masked tokens with the [MASK] token is to force the model to predict the missing words based on the context provided by the surrounding tokens. By doing so, the model learns to understand the relationships between words in a sentence and how they contribute to the overall meaning. This focus on contextual understanding is crucial for the model's performance in various downstream tasks, such as question answering, sentiment analysis, and named entity recognition.

During training, the model is presented with sentences where 80% of the tokens are masked. The objective is to predict the original words at these masked positions based on the remaining visible tokens. By focusing on this task, the model learns to capture the contextual information present in the sentence and make accurate predictions. This process helps the model develop a strong understanding of word relationships and enables it to generate more coherent and contextually appropriate sentences.

10% Replaced with Random Words:

Replacing 10% of the masked tokens with random words from the vocabulary introduces noise into the training data. This strategy helps the model become more robust and better equipped to handle unexpected or novel input during real-world applications.

During training, the model encounters sentences with 10% of the tokens replaced with random words. The objective remains the same: predict the original words at the masked positions based on the surrounding tokens. However, the presence of random words in the input adds an element of uncertainty and challenges the model's ability to make accurate predictions. By exposing the model to this noise, it learns to handle unexpected input and make more generalized predictions. This improves the model's robustness and its ability to adapt to new or unseen data.

10% Left Unchanged:

Leaving 10% of the masked tokens unchanged helps the model generalize better and avoid overfitting to the [MASK] token specifically.

During training, the model encounters sentences with 10% of the tokens left unchanged. The objective for these tokens is to predict the original words based on the remaining visible tokens, just like the masked tokens. However, the presence of unchanged tokens provides the model with additional information about the original words in the sentence. This information can help the model better understand the context and make more accurate predictions.

By leaving some masked tokens unchanged, the model is exposed to a more balanced and diverse set of training examples. This helps the model generalize better and avoid overfitting to the [MASK] token specifically. This generalization capability is crucial for

the model's performance in real-world applications, where it may encounter a wide range of input data that was not present during training.

In summary, the masking strategy employed in MLMs such as BERT is designed to enhance the model's ability to learn contextual information, improve its robustness, and generalize better. By replacing 80% of the masked tokens with the [MASK] token, the model is forced to focus on predicting missing words based on the surrounding context. Replacing 10% of the masked tokens with random words introduces noise into the training data and helps the model become more robust. Leaving 10% of the masked tokens unchanged provides the model with additional information about the original words in the sentence and helps it generalize better. This masking strategy plays a crucial role in the model's performance and its ability to handle a wide range of input data during real-world applications.

Question about how to make our MLM better:

To improve the performance of our Masked Language Model (MLM), we can take the following steps:

1. Pretraining on a large dataset: Training the MLM on a large dataset can significantly improve its performance. BERT, for example, was pretrained on the BooksCorpus (800M words) and English Wikipedia (2,500M words) datasets. Using a larger and more diverse dataset can help the MLM learn more about the language and its nuances.
2. Increase the model size: Increasing the model size, such as the number of layers, hidden units, or attention heads, can improve its performance. A larger model can capture more complex patterns and relationships in the data.
3. Increase the training time: Training the MLM for a longer time can help it converge to a better solution. Increasing the number of training epochs or the batch size can improve the model's performance.
4. Use a more sophisticated optimization strategy: Using a more sophisticated optimization strategy, such as learning rate warm-up, gradient clipping, or layer-wise learning rate decay, can improve the model's performance. These strategies can help the model converge faster and avoid getting stuck in local minima.
5. Fine-tuning on a downstream task: Fine-tuning the MLM on a downstream task, such as question answering or sentiment analysis, can improve its performance. Fine-tuning

allows the model to adapt to the specific requirements of the task and improve its accuracy.

6. Use a pretrained model: Using a pretrained model, such as BERT or RoBERTa, can significantly improve the performance of your MLM. Pretrained models have been trained on large datasets and can capture complex patterns and relationships in the language. Fine-tuning a pretrained model on a smaller dataset can help the model achieve good performance with less data.