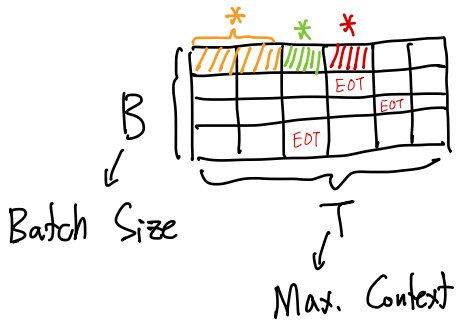
GPT Assistant Training Pipeline.

① Pretraining → ② Supervised Finetuning → ③ Reward Modeling → ④ Reinforcement Learning

1) Pretraining

- Computing time의 99% 차지 → Base model 만들기
- 많은 GPU, 많은 시간 학습 (수천개, 몇 일) ⟷ 나머지 단계: 소수 GPU, 몇 시간 ~ 며칠
- 여러가지 학습 data 수집 → 특정 비율에 따라 샘플링해서 학습
- 학습 전에 Tokenization : Raw text → token → Sequence of Integers
- Parameter 개수 : GPT-3 > LLaMA
  - but 학습 token 개수 : GPT-3 < LLaMA
  - → LLaMA가 성능 더 ↑ → parameter 개수로 model 성능 판단 X
- 실제 training :



1) context length 단위로 row로 쌓음.
2) <[EOT]>로 document 구분
3) *에서 *의 context를 transformer에 제공
   → transformer는 * 예측
   → transformer의 W 업데이트

- GPT1 : 감정분류를 위해 큰 Language model을 pretraining → Fine-tuning
  ↓
  GPT2 : Context 주고 Q : — ? 식으로 질문 후 A : .
- Base model ≠ Assistants
  - Base model ↓ LLaMA
  - Assistants ↓ GPT-4
  - → 질문에 답하는게 아니라 document 완성하는 model
    ex) 시를 쓰라고 지시하는것 보다 시의 첫문장 쓰는게 더 좋음.
- Base 모델을 Assistant 처럼 사용하기 위해 어떤 document 있는 것처럼 ...
  → 잘 안된다.

2) Supervised Finetuning

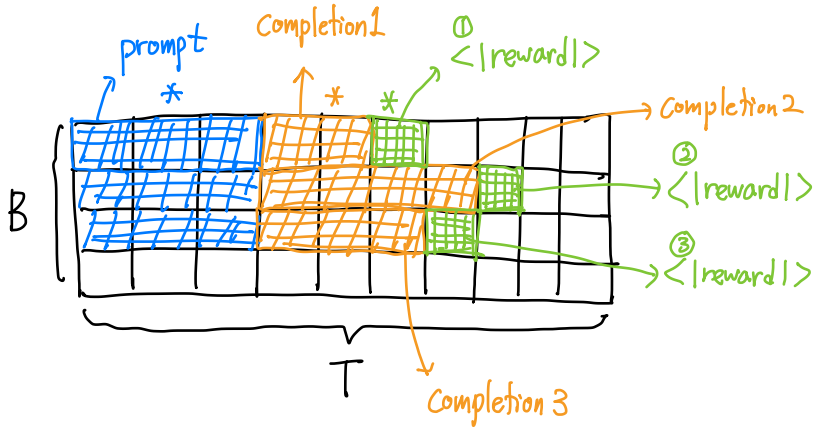
- 실제 GPT Assistant 만들기 (≠ document 예측하는 base model)
- 인간에게 prompt, 이상적인 response 요청
  → Size ↓, quality ↑인 dataset 학습

인간이 쓴 helpful, truthful, harmless한 내용

3) Reward Modeling

- 앞서 만든 SFT model 이용해서 동일한 prompt에 대해 3개의 결과 만든다.
  → 사람이 1~3등 순위 매김.

- 학습:



*, * 는 무시하고 * (reward token) 만 transformer에 supervised learning 시킴.

↓

transformer는 reward 예측 → 정수화

4) Reinforcement Learning

- Reward Model을 이용해서 *을 생성하는 policy를 바꿔나감.
  → RLHF: Reinforcement Learning with Human Feedback.

- RLHF 모델은 Base 모델 보다 살짝 Entropy 손실 있음.