

## Motivation & Summary

- LLM을 추천시스템의 lever로 사용하는 것은(leverage) 최근에 큰 주목을 받고 있고, 좋은 성능을 내기 위한 LLM adaptation에 fine-tuning이 중요한 역할을 하고 있음.
- 하지만 매우 많은 item이 새롭게 등장하고 있고(ex, TikTok), 이것을 LLM에 반영하기 위해서 빈번한 (frequent, 또는 많은) fine-tuning이 필요하고, 이는 많은 비용을 필요로 함.
- Fine-tuning의 비용을 줄이기 위해, LLM에서 좋은 성능을 기대할 수 있는(tailored) few-shot fine-tuning을 적용해볼 수 있음. 여기서 key point는 어떤 data pruning 기법을 이용하면, representative few-shot data를 얻을 수 있는가임. (representative samples은 새로 등장하는 item과 user 행동을 LLM이 이해할 수 있게 하는 데이터임)
  - 모델 전체 성능에 영향을 주는(influence) sample을 정확하게 식별할 수 있어야 함
    - \* Influence를 identify 할 수 있는 metric이 필요함
    - \* 각 sample에 대해 모두 계산해야 해서 비용이 많이 들기 때문에, 비용을 줄이는 방법(ex, estimate once)등이 필요함
  - data pruning process에 들어가는 비용을 효율적으로 낮출 수 있어야 함
    - \* LLM에 직접 pruning process를 적용하는 것은 비용이 상대적으로 크므로, LLM을 대신 할(surrogate) 것이 필요할 수 있음
    - \* LLM을 대신한다면, LLM과 surrogate model의 간극(gap)을 고려해야 할 수 있음
- 다양한 접근 방법이 존재할 수 있음
  - Random : 효율적이지만 정확하지 않음
  - Coreset selection
    - \* Heuristic methods : optimum을 구하기 어렵고 empirical risk에 미치는 영향을 추정하기 힘들
    - \* Optimization-based : large-scale 데이터에서 bi-level optimization 문제를 풀기 어려움
  - DEALRec
    - \* Influence score : 각 training sample을 제거했을 때 empirical loss에 대한 영향력(influence)을 추정한 값(estimate)
    - \* Effort score : LLM parameter에 대한 sample loss의 gradient norm. 즉 특정 sample을 LLM에 fitting하기 위해서 들어가는 비용

## Task formulation

### 1. Fine-tuning an LLM for sequential recommendation tasks

- user sequence :  $s = (x, y)$ , where  $x = [i_1, \dots, i_{|x|}]$  is the user's historical interactions in chronological order, and  $y$  is the next interacted item of the user, where  $\{i_1, \dots, i_{|x|}, y\} \subset \mathcal{I}$
- training set :  $\mathcal{D} = \{s_u | u \in \mathcal{U}\}$
- Optimization of the learnable parameters ( $\phi \in \Phi$ ) of an LLM :

$$\min_{\phi \in \Phi} \left\{ \mathcal{L}_{\phi}^{LLM} = - \sum_{t=1}^{|y|} \log P_{\phi}(y_t | y_{<t}, x) \right\}$$

, where  $y_t$ 는  $y$ 의  $t$ -th token을 나타내고, and  $y_{<t}$ 는  $y_t$  앞의 token sequence를 의미한다.

## 2. Data pruning

- Training sample  $\mathcal{D} = \{s_u | u \in \mathcal{U}\}$ 의 부분집합  $\mathcal{S} \subset \mathcal{D}$ 에 대해,  $\mathcal{S}$ 를 이용한 few-shot fine-tuning 된 LLM이 testing set에서 좋은 성능을 낼 수 있도록 하는것이 목표

## Methodology

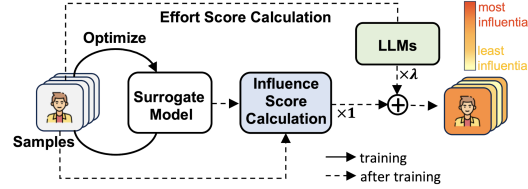


Figure 2: Overview of DEALRec. DEALRec first trains a surrogate model on the full training samples. Subsequently, it calculates the influence score, which is then regularized by the effort score, to identify influential samples.

### 1. Influence score

- (Understanding Black-box Predictions via Influence Functions 논문을 참고)
- Pruned dataset  $\mathcal{S}$ 를 이용하여 좋은 성능을 내기 위해서는, empirical risk에 대한 influence를 평가할 수 있어야 함. 이 성능은, 학습데이터의 sample을 각각 제거했을 때 empirical risk를 의미.
- 각 sample을 제거했을때의 influence를 모두 측정하는것은 비현실적. 계산을  $|\mathcal{D}|$ 번 해야함.
- Parameter change에 대한 influence를 이용하여, empirical risk에 대한 influence를 추정하고자 함
  - Parameter change에 대한 influence : sample에 대해  $\epsilon$ 만큼 upweight를 시켰을 때, empirical risk의  $\epsilon$ 에 대한 미분값을 활용 (Hessian and positive definite)
  - Parameter change에 대한 influence와 chain rule을 이용
- Influence score를 효율적으로 계산하기 위해서, stochastic-based Hessian-Vector Products를 활용

### 2. Gap regularization

- Influence score 계산 비용을 줄이기 위해 surrogate model을 이용하고, surrogate model과 LLM 사이의 gap을 줄인다.
- Surrogate model : small-size의 traditional recommender model
- Effort score : LLM에 더 중요한 sample을 추출하기 위한 score.
  - $\delta_s = \|\nabla_{\phi} \mathcal{L}^{LLM}(s)\|_2$ .
  - user sequence를 LLM에 fitting 시키기 위한 learning effort (or cost)라 생각하면 됨.
- Overall score : influence score + effort score

### 3. Few-shot fine-tuning

- Few-shot sample selection은 stratified sampling 기법을 활용
- Overall scores에 따라 전체 sample을  $K$ 개의 그룹으로 나누고, 반복적으로  $n_s$  sample을  $K$ 번 추출.  $n_s$ 는 budget을 의미하며 평균 샘플 수로 구함. 계속 업데이트 됨 (초기값 :  $\left\lfloor \frac{r|D|}{K} \right\rfloor$ )
- 추출된 few-shot sample을 이용하여 LLM의 학습가능한 parameter를 optimization

$$\hat{\phi} = \arg \min_{\phi \in \Phi} \frac{1}{|\mathcal{S}|} \sum_{s_i \in \mathcal{S}} \mathcal{L}_{\phi}^{\text{LLM}}(s_i)$$

## Experiments

- Models
  - BIGRec : Llama-7B & Lora (PEFT) & item title을 user sequence에 활용
  - TIGER : transformer-based architecture & full fine tuning & item feature에서 별도의 token을 학습에 활용
- Few-shot fine tuning과 full fine tuning을 비교했을 때, 전자의 성능이 높았음. 이것이 의미하는 것은, 모든 user의 sequences가 모델 학습에 informative하지 않으며 오히려 방해가 될 수 있다는 것을 보여줌 (False negative interaction)