[Study 3. 박하늘] Survey on Knowledge Distillation for Large Language Models: Methods, Evaluation, and Application

- Title: Survey on Knowledge Distillation for Large Language Models: Methods, Evaluation, and Application
- Authors: Yang, C., Zhu, Y., Lu, W., Wang, Y., Chen, Q., Gao, C., ... & Chen, Y.
- Publication: *ACM Transactions on Intelligent Systems and Technology*

Abstract

- LLM의 뛰어난 성능에도 불구하고, 거대한 규모와 높은 연산 비용은 실제 서비스 환경, 특히 **자원이 제한된 환경**에서의 배포에 큰 제약이 된다.
- 정확도를 최대한 유지하면서도 언어 모델을 압축하려는 시도가 많았음

- **Knowledge Distillation**은 성능을 크게 떨어뜨리지 않으면서 추론 속도를 향상시키는 효과적인 방법으로 부각

Introduction

Large Language Model

- LLM은 다양한 생성 과제에서 텍스트 생성 품질을 크게 향상시켜 인공지능 분야에서 핵심적이고 뜨거운 주제가 되었다.
- 기존 모델들에 비해 일반화 능력, 다단계 추론, 지시 따르기 등 고급 능력을 보여주었음.
- LLM의 성공 자체는 **대량의 학습 데이터와 모델 파라미터 수** (GPT-3 1750억 파라미터)에 기인
- 이로 인해 막대한 추론 비용, 메모리 요구로 실 서비스 배포에 제약이 있다.
- ex) GPT-3는 float16 기준 350GB 모델 저장 공간이 필요하며, 추론 시 최소 80GB 메모리 5개 A100 GPU를 요구

Knowledge Distillation

- 연산 자원 요구를 줄이는 필요성이 커짐에 따라 Knowledge Distillation(KD)이 유망한 기법으로 떠올랐다.
- KD는 대형 딥러닝 모델의 지식을 소형 모델로 옮겨 연산 비용을 줄이고 추론 속도를 높이면서 성능 손실을 최소화한다.
- 훈련 중 소형 모델은 원본 데이터 레이블뿐 아니라 대형 모델의 동작을 모방하도록 학습된다.

Knowledge Distillation in Large Language Models

본 논문에서는 크게 2가지로 나누어 KD를 설명한다.

- White-Box KD: 학습 시 teacher 모델의 접근 가능한 내부 정보를 활용한다. (open model)
- Black-Box KD: teacher의 출력 값만을 사용한다. (closed model)

White-Box Knowledge Distillation

Logits-Based KD

마지막 출력 분포를 학습한다.

$$\mathcal{L}_{\text{logits}} = KL\left(p^{\mathbf{t}} \| p^{\mathbf{s}}\right) = \sum_{j=1}^{C} p_j^{\mathbf{t}} \log\left(\frac{p_j^{\mathbf{t}}}{p_j^{\mathbf{s}}}\right),$$

- teacher와 student가 같은 입력에 대해 가능한 클래스들에 부여하는 확률 분포를 최대한 비슷하게 만들도록 KL divergence를 최소화하는 loss.

Hint-based KD

logit 기반 지식 중류에서는 지식 추출 능력에 한계가 있음. Hint-based KD에서는 결과 뿐만 아니라 중간 레이어 표현을 학습한다.

$$\mathcal{L}_{\text{hint}} = \mathcal{H}\left(F^{\mathbf{s}}, F^{\mathbf{t}}\right) = \left\|F^{\mathbf{t}} - \phi\left(F^{\mathbf{s}}\right)\right\|^2,$$

- F: 중간 레이어 feature map
- ϕ: student feature의 차원을 teacher feature와 맞춰 주는 변환 함수(예: 1x1 conv, 선형 projection 등)
- H: metric function (예: MSE, cosine similarity 등)

한계

- 레이어 매핑 설계가 매우 까다로움
- 모델 구조를 **깊이 이해해야 함**
- GPU 메모리 사용량 큼
- **학습 비용·시간 줄이는 방향이 중요**
- 실용성 측면에서 부담 큼

Black-Box Knowledge Distillation

White-Box 증류 기법은 모두 teacher 모델의 **내부 정보**에 접근해야 한다. 반면 많은 최신 모델은 이런 내부 정보를 공개하지 않는 경우가 많다. 이런 상황에서 teacher의 **출력값**만으로 지식을 옮기는 방식을 **black-box knowledge distillation**이라고 한다.

In-Context Learning

- Teacher LLM이 추론한 출력 예시들을 보고 모방하도록 설계

$$\mathrm{LLM}(I, \underbrace{f(x_1, y_1), \ldots, f(x_k, y_k)}_{\text{demonstrations}}, f(\underbrace{x_{k+1}}_{\text{input}}, \underbrace{\quad}_{\text{answer}})) \rightarrow \hat{y}_{k+1},$$

```
# I
Task: Determine the sentiment of the sentence.

# f(x_1, y_1)
Example 1:
Sentence: "I love this movie."
Sentiment: Positive

# f(x_2, y_2)
Example 2:
Sentence: "This was a terrible experience."
Sentiment: Negative
```

```
# f(x_k+1, )
Sentence: "The service was okay, nothing special."
Sentiment:
```

Chain-of-Thought

복잡한 추론 문제를 잘 풀기 위해 최종 답만 제시하지 않고 중간 추론 과정을 함께 프롬프트에 포함하는 기법

$$\text{LLM}(I, \underbrace{f(x_1, r_1, y_1), \ldots, f(x_k, r_k, y_k)}_{\text{demonstrations}}, f(\underbrace{x_{k+1}}_{\text{input}}, \underbrace{\quad}_{\text{rational}}, \underbrace{\quad}_{\text{answer}})) \rightarrow \hat{r}_{k+1}, \hat{y}_{k+1}.$$

- student 모델은 정답뿐 아니라 teacher 모델이 생성한 추론 과정까지 함께 모방하도록 학습된다.

```
# I
You are a helpful reasoning assistant. Think step by step and then give the final answer.

# f(x_1, r_1, y_1)
Q: If Alice has 3 apples and buys 2 more, then eats 1, how many apples does she have?
Reasoning: Alice starts with 3 apples. She buys 2 more, so 3 + 2 = 5. Then she eats 1, so 5 - 1 = 4.
Answer: 4

# f(x_2, r_2, y_2)
Q: A train travels 120 km in 2 hours. What is its average speed in km per hour?
Reasoning: Average speed is distance divided by time. The train travels 120 km in 2 hours, so 120 / 2 = 60.
Answer: 60
```

```
# f(x_k+1, r_k+1, y_k+1)
Q: Tom has 5 red balls and 7 blue balls. How many balls does he have in tot
al?
Reasoning:
Answer:
```

Instruction Following

- teacher LLM에게 여러 **task-specific instruction + 정답**을 생성하게 하고,
- 그걸 모아서 instruction 데이터셋을 만든 뒤 student 모델을 그 위에서 파인튜닝하면, student가 teacher의 instruction-following 지식을 distill한다

```
# 일반 KD
문제: 고양이 사진이다 → 정답: cat
Student: "아 cat 이렇게 푸는 거구나"

# Insturction Following KD
Instruction: "다음 이미지를 보고 어떤 동물인지 말하세요."
Input: (고양이 이미지)
Output: "고양이"
```

- 어떨 때 쓰는가
  - 모델이 질문을 잘 못 알아들을 때
    - 한 단어로 답해, json으로 출력해 등
  - 하나의 모델로 요약, 분류 등의 task를 instruction만 바꿔서 하고 싶을 때

한계

- 데이터 생성 비용 비쌈
- 메모리 적게 듬

VL2Lite: Task-Specific Knowledge Distillation from Large Vision-Language Models to Lightweight Networks