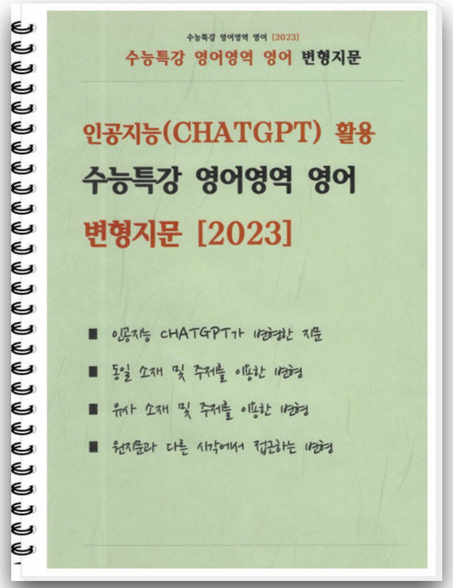
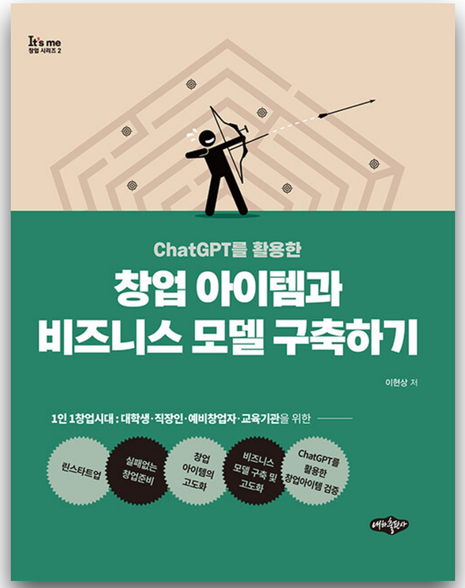
안전하고 효율적인 코드 언어 모델 학습

SIGPL 2024 여름학교, 류연희, 박일범, 허기홍, 김기응

KAIST Programming Systems Laboratory

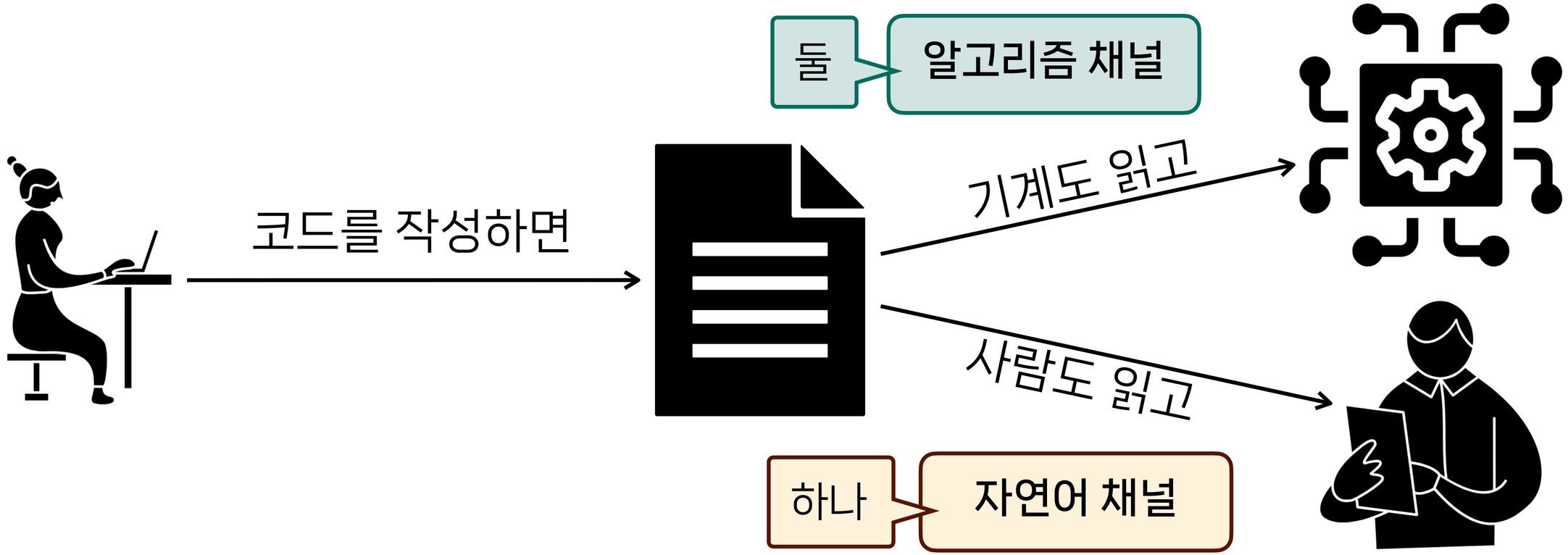
그냥 ChatGPT 한테 물어보면 안됨? 🤷





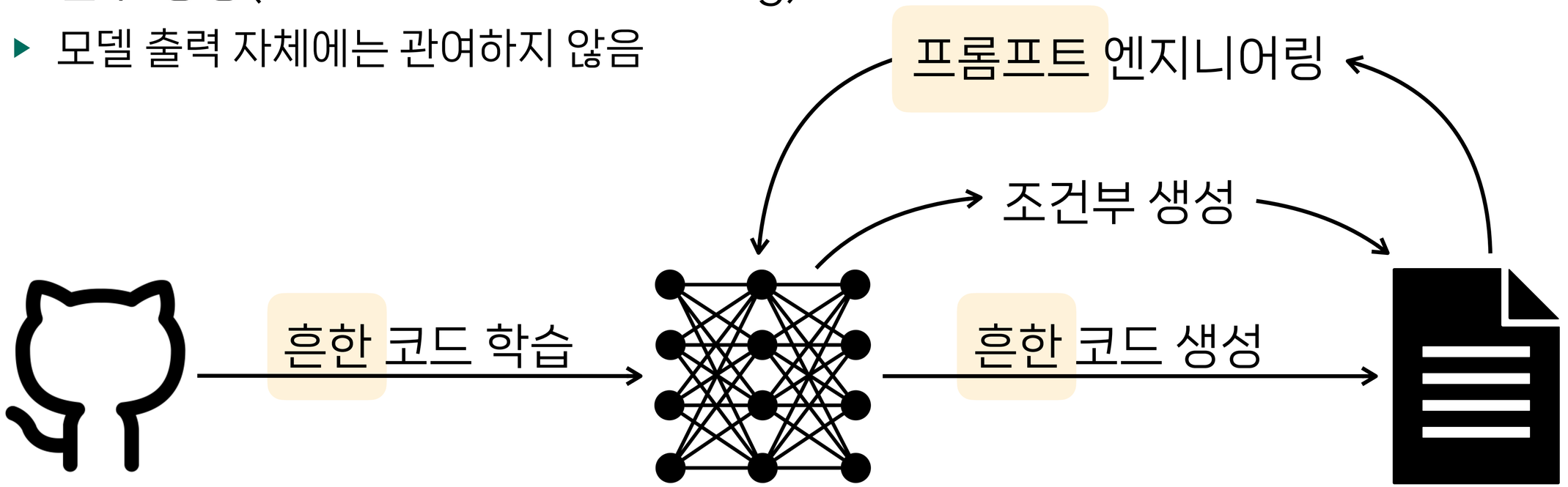
하나만 알고 둘은 모른다

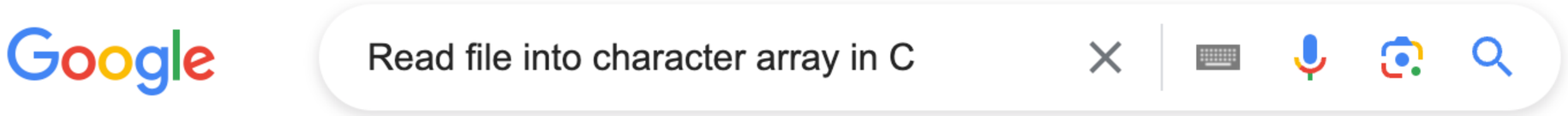
- 소스 코드를 읽는 두 가지 채널
  - A Theory of Dual Channel Constraints, ICSE-NIER'20



그냥 ChatGPT 한테 물어보면 왜 안됨?

- 모든 과정에서 자연어 채널 정보만 사용
- 알고리즘 채널은 자연어 채널을 통해 간접적으로 사용
- 조건부 생성(Constrained decoding)
  - 모델 출력 자체에는 관여하지 않음






```
File *f = open("file.txt", "r");
char buf[1000];
fread(buf, ...)
```

안전

```
File *f = open("file.txt", "r");
char *buf = malloc(1000);
fread(buf, ...)
```

위험

```
File *f = open("file.txt", "r");
char *buf = malloc(1000);
if (buf == NULL) // do something
fread(buf, ...)
```

안전

언어 모델이 생성한 코드의
40%는 보안 취약점을 내포

(Pearce et. al., S&P 2022)

위험!

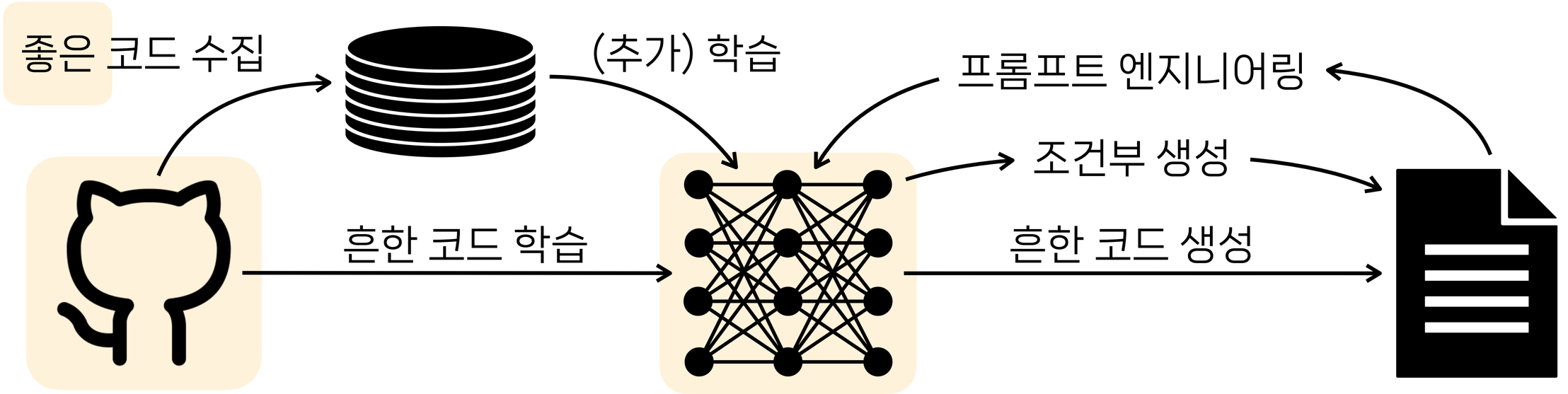
C read.c

```c
#include <stdio.h>

int main() {
  FILE *f = fopen("file.txt", "r");
  char *buf = malloc(100);
    fread(buf, 1, 100, f);
```

그럼 좋은 코드만 골라서 LLM 학습하면 안됨?

- "좋은 코드" 의 기준이 다양
  - 예: division-by-zero 에 안전하지만 속도가 매우 느린 코드
- 거대 모델이 학습할 좋은 코드 데이터 2026년부터 고갈 (Villalobos et. al., arxiv 2022)
- 모델 구조는 여전히 자연어 채널만 고려



좋은 코드 데이터 수집 문제

- "문제가 있는 코드"와 이에 대응되는 "좋은 코드"를 쌍으로 수집하기 어려움
  - 문제가 있는 코드 데이터가 매우 드물게 분포
  - 문제의 원인을 명확하게 분리하기 어려움

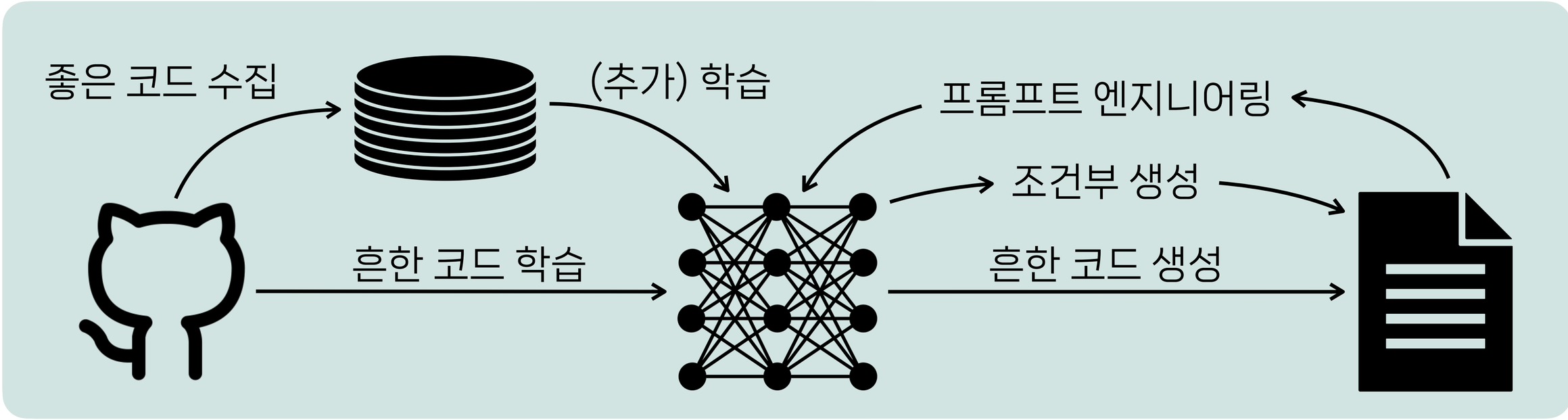
기술	학습 대상 문제	데이터 규모	수집 방법
SVEN (He & Vechev, CCS'23)	보안 취약점	C & Python Commit 1,606개	저자가 직접 수집
SafeCoder (He et.al., ICML'24)	보안 취약점	6개 언어 Commit 465개	CodeQL 이용하여 자동 수집
CodeRL (Le et.al., NeurIPS'22)	유닛 테스트 성패	APPS 5K, MBPP 374	사람이 직접 수집한 벤치마크 사용
IRCoCo (Li et.al., FSE'24)	완성된 코드	Py150 100K, JavaCorpus 11K	자동 수집된 벤치마크 사용

알고리즘 채널도 이용하는 모델 학습 방법

- 코드 실행 의미를 고려하는 데이터를 효율적으로 수집
- 코드 실행 의미를 고려하는 효율적인 학습 방법
- 코드 실행 의미로 조절 가능한 생성 기법

정적 분석

강화 학습



자세한 내용은 포스터에서

기존 모델 만큼 자연어를 잘 이해하면서

기존 모델보다 안전한 코드 생성