

생각하는 개발자

조현준

거대 언어 모델(Large Language Model, LLM)은 잘 구조화된 프로그래밍 언어에서 특히 강점을 보이며, 개발 문화에 점점 더 큰 영향을 미치고 있다. 그러나 LLM은 정확성의 한계로 인해 단순한 작업에는 유용하지만, 높은 신뢰성이 요구되는 프로그래밍에서는 위험을 동반한다. LLM이 생성한 코드의 신뢰도를 높이기 위해 모델의 행동을 이해하기 위한 연구, 도구로서 활용 범위를 제한하는 현실적인 접근이 필요하다. 결론적으로, 기술의 발전에도 불구하고 창의적 사고와 도구 활용 능력을 가진 개발자는 여전히 중요한 역할을 할 것이다.

GPT3의 등장 이후, 수많은 연구자들이 분야를 막론하고 거대 언어 모델(Large Language Model)을 자신들의 연구에 활용하고 있다. 이러한 추세는 특히나 컴퓨팅 분야에서 두드러지는데, 언어 모델의 통계적 특성을 고려하면 자연스러운 현상이다. 자연어 전체와 비교했을 때, 프로그래밍 언어는 잘 구조화되어 있고 그래서 예측이 수월하다. 덕분에 최근 2~3년 동안 깃헙 Copilot과 같은 프로그래밍 도구들이 자연스레 개발 문화에 녹아들었다. 그와 동시에 한 가지 의문이 떠오른다. “10년 후에도 개발자로 먹고 살 수 있을까?” 이 질문에 대한 정답은 아무도 모를 것이다. 그나마 합리적인 결론을 도출하기 위해서는 언어 모델의 장단점, 그리고 한계에 대해 생각해볼 필요가 있다.

거대 언어 모델은 무엇을 잘하고, 무엇을 못할까. 언어 모델은 매번 그럴듯한 문장을 만든다. 하지만 매번 정확한 문장을 만드는 것은 아니다. 이 그럴듯함이 거대 언어 모델의 장점이자 단점이다. 덧붙여서 이 또한 통계에 기반한 모델의 특성일 것이다. 학습에 활용된 방대한 말뭉치(Corpus)를 토대로 다음에 올 확률이 높은 단어, 즉 가장 그럴듯한 단어를 택하는 것이 언어 모델의 본질이다. 그럼 이 언어 모델을 우리네 삶에 어떻게 활용할 수 있을까?

정답이 없는 작업에는 강점이 있다. 소설을 이어 쓰거나, 자연스러운 대화를 이어나가는 것으로 충분하다면 탁월한 선택이다. 또 정확도가 다소 떨어지더라도 사용자가 직접 검증하고 조정할 수 있는 상황이라면 쓸만 할 것이다. 일례로 웹사이트에서 언어 모델에 기반한 챗봇 서비스를 제공하는 경우가 늘어나고 있다. 또 우리 수업의 천하제일 예술대회에서와 같이 비전문가들이 생각하는 바를 빠르게 그림이나 음악, 영상으로 바꿔 볼 수 있다는 점 역시 언어 모델의 장점이다.

하지만 정확도가 중요한 경우, 언어 모델의 장점은 이내 단점이 되어 버린다. 프로그래밍이 대표적인 예시다. 언어 모델이 생성한 코드가 그럴듯하다는 이유로 무분별하게 갖다 쓴다면, 이후 어떤 문제가 생길지 알 수 없다. 우리가 수행하는 과제만 해도 테스트가 잘 갖춰져 있기 때문에, 작은 오류에도 민감하게 반응한다. 하물며 수만명의 사용자들이 접속하는 서비스는 어떨 것이며, 자율주행이나 의료 체계와 같은 생명과 직결되는 프로그램이라면 어떻겠는가. 오류는 허용되지 않는다. 최악의 경우에는 언어 모델이 생성한 코드를 고치는 비용이 우리가 직접 구현하는 비용보다 커질 수 있다.

그럼 어떻게 하면 이 문제를 극복할 수 있을까? 이상적인 답변은 거대 언어 모델의 행동을 이해하는 것이다. 우리가 프로그래밍을 할 때에는, 매순간 의사결정을 내린다. 한 개 함수에는 얼마만큼의 코드를 담을지, 여러 객체들간의 상호작용은 어떻게 구현할 것인지 등이다. 언어모델이 코드와 더불어 자신이 해당 코드를 내어놓게 된 과정을 설명할 수 있다면, 혹은 우리가 모델 내부 뉴런의 활성도를 바탕으로 이 같은 정보를 추출할 수 있다면 조금 더 믿고 활용할 수 있을 것이다. 보다 현실적인 타협점은 애당초 문제가 생기지 않는 선에서, 언어 모델은 어디까지나 도구로써 이용하는 것이다. 냉소적이지만 확실한 해결책이다. 본격적인 작업에 앞서 빠르게 시제품을 만들어보거나, 정형화된 시각화 작업 등을 할 때에는 분명히 시간을 절약할 수 있다.

다시 “개발자로 먹고 살 수 있을까?”라는 질문으로 돌아가보자. 우리는 언어 모델이 없던 때에도 늘 대체 불가능한 인력이 되기 위해 고군분투했다. 새로운 기술을 익히고 더 나은 프로그램을 작성하기 위해 고민한다. 언어모델이 급부상하고 있는 것은 사실이나, 그보다도 뛰어난 개발자들이 시시각각 등장함을 기억하자. 우리가 진정으로 길러야할 능력은 생각하는 힘이고, 인공지능 도구를 도구로써 활용하는 능력이다.