1. 실험에 사용한 문장 예시

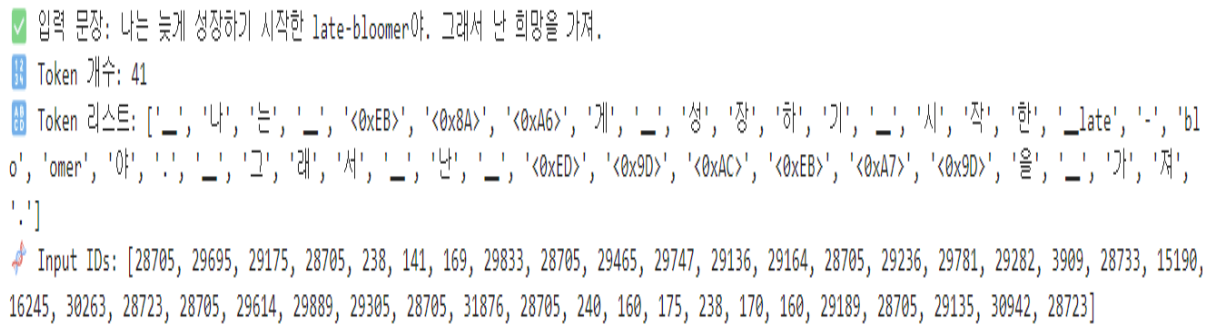
- “나는 늦게 성장하기 시작한 late-bloomer야. 그래서 난 희망을 가져.”

2. 사용한 tokenizer 이름 및 링크

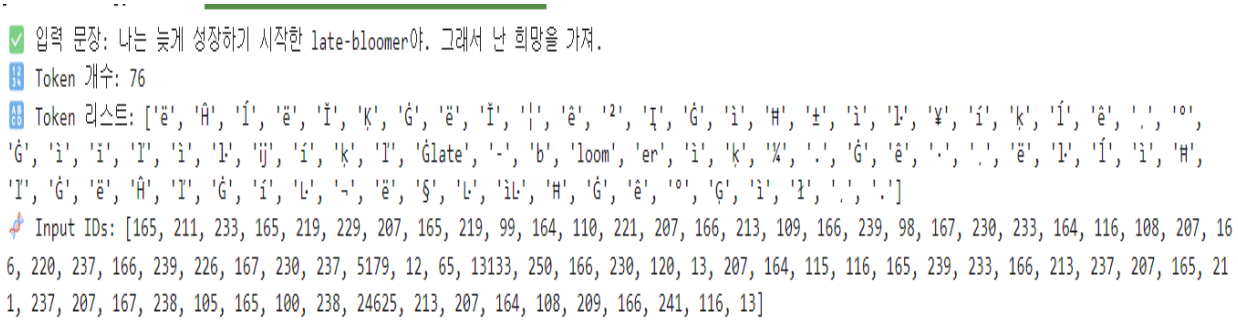
- DeepSeek-V3-0324 (https://huggingface.co/deepseek-ai/DeepSeek-V3-0324)
- openchat/openchat-3.5-0106 (https://huggingface.co/openchat/openchat-3.5-0106)

3. 각 tokenizer에서 분해된 token 수 및 token 목록

- openchat-3.5-0106 결과



- Deepseek-V3-0324 결과



4. 가장 흥미로웠던 token 분해 결과와 이유

- OpenChat은 유니코드 방식으로 분해한 것 같은데, 딥시크는 토큰이 많은 대신 그걸 효율적으로 잘게 잘라 메모리를 경량화한게 흥미로웠습니다. 두 토크나이저의 접근 방식이 확연히 다른 것을 볼 수 있었습니다. OpenChat-3.5는 한글 문자를 비교적 온전하게 보존하며 '나', '는', '늦', '게'와 같이 음절 단위로 토큰화했고, 영어 단어인 'late-bloomer'도 음절 단위로 유지했습니다. 반면 Deepseek-V3는 더 세밀한 수준으로 토큰을 분해하여 전체적으로 토큰 수는 많아졌지만, 각 토큰의 크기를 줄여 메모리 사용을 최적화한 것으로 보입니다. 이러한 차이는 각 모델의 설계 철학과 자원 활용 전략을 반영하는 것으로, 특히 Deepseek-V3의 경량화 접근 방식이 주목할 만했습니다.

앞으로는 주제를 더 정확히 이해하고 분석하여 보고서를 작성하겠습니다. 이번에는 두 문장 이상을 tokenizer에다가 하라는 권고사항을 읽지 못하고 성급하게 실험했던 점을 인정합니다.