# [발표 스크립트] LLM의 핵심 첫걸음: 토크나이저의 모든 것

## 01. 도입 및 중요성 (Why it matters)

[스크립트]

안녕하세요.

오늘은 지난 시간에 알아보았던 LLM의 성능을 결정짓는

가장 첫 번째 관문이자 핵심 기술인

'토크나이저'에 대해 알아보겠습니다.

흔히 우리는 모델의 파라미터 수나 아키텍처에 집중하지만,

사실 모델이 세상을 바라보는 '눈'은 바로 토크나이저 라고 합니다

토크나이저는 우리가 사용하는 텍스트를 모델이 처리할 수 있는

숫자 시퀀스로 변환하는 장치인데요

여기 예시를 보시면, "Hello, world!"라는 문장이 토큰으로 분할되고 최종적으로 ID 값으로 변환되는 것을 볼 수 있습니다. 만약 이 첫 단계에서 "tokenizing"이라는 단어를 의미 없는 "t", "okeni", "zing"으로 쪼갠다면, 모델은 단어의 본래 의미를 파악하는 데 훨씬 더 많은 고생을 하게 됩니다.

즉, 효율적인 토큰화가 모델의 지능과 직결되는 겁니다

같은 의미의 문장이라도 언어의 특성에 따라 토큰화된 결과의 길이는 크게 달라질 수 있습니다.

특히 영어 중심의 다국어 모델에서는 한국어가 비효율적으로 처리되는 경우가 많습니다. GPT-4 토크나이저를 예로 들면, 같은 의미의 문장이 언어별로 얼마나 다른 토큰 수를 갖는지 알 수 있습니다. \*

**영어:** I ate an apple (4 토큰) \*

**한국어:** 나는 사과를 먹었다 (10 토큰) \*

**일본어:** 私はリンゴを食べました (12 토큰)

이렇게 토큰 수가 늘어나는 것은 두 가지 측면에서 문제가 됩니다. 1.

**응답 속도 저하:** 모델이 처리해야 할 토큰이 많아져 LLM의 응답이 느려집니다. 2.

**API 비용 증가:** 대부분의 LLM API는 토큰 길이에 비례하여 과금됩니다.

이런 문제를 해결하기 위해 SKT는 한국어 특성을 깊이 고려한 최적화된 토크나이저를 개발했습니다.

간단한 인사말인 "안녕하세요" 과거 OpenAI의 토크나이저는 이 다섯 글자를 14개의 토큰으로 분할했지만, SKT 에이닷 토크나이저는 단 1개의 토큰으로 처리했습니다.

이후 GPT 업데이트를 통해 5개 토큰으로 개선되었지만, 여전히 한국어 특화

토크나이저와는 상당한 차이를 보입니다.

## 02. 토큰화의 종류와 발전 (What it is)

[스크립트]

그렇다면 단어를 어떻게 쪼개는 것이 가장 좋을까요?

과거에는 세 가지 방식이 고민되었습니다.

첫째는 **단어 기반**입니다.

공백을 기준으로 쪼개면 쉽지만, 어휘 사전이 너무 거대해지고

사전에 없는 단어인 'OOV(Out-of-Vocabulary)' 문제가 발생합니다.

신조어나 오타가 나오면 모델은 당황하게 되죠.

둘째는 **문자 기반**입니다.

모든 글자를 토큰으로 쓰면 OOV 문제는 사라지지만, 문장이 너무 길어지고 개별 글자에는

의미가 담기지 않아 학습 효율이 매우 낮습니다.

그래서 탄생한 것이 현재의 표준인 **서브워드(Subword)** 방식입니다.

자주 쓰이는 단어는 통째로, 드문 단어는 의미 있는 하위 단위로 쪼개는 전략입니다.

이를 통해 어휘 사전의 크기는 압축하면서도, 모든 단어를 유연하게 표현할 수 있게 되었습니다.

## 03. 핵심 알고리즘: BPE와 WordPiece (How it works)

[스크립트]

이제 기술적으로 들어가 보겠습니다. 현재 가장 널리 쓰이는 두 가지 알고리즘은 BPE와 WordPiece입니다.

\*\*BPE(Byte Pair Encoding)\*\*는 데이터 압축 기술에서 왔습니다.

가장 자주 함께 나타나는 문자 쌍을 반복적으로 병합하여 토큰을 만듭니다.

GPT 계열 모델의 근간이 되는 기술이죠.

반면, **WordPiece**는 단순히 빈도만 보는 것이 아니라 통계적인 '우도(Likelihood)'를 계산합니다.

관찰된 데이터를 바탕으로 특정 통계 모델의 모수(parameter)가 얼마나 그럴듯한지를 나타내는 척도입니다.

빈도와 우도라는 두 토큰을 합쳤을 때 전체 학습 데이터의 확률을 얼마나 높이는가?"를 점수로 매겨 병합합니다. 얼마나 그럴듯 한가 라는거죠

여러분이 BERT 모델 결과에서 흔히 보는 ## 표시가 바로 WordPiece 알고리즘이 적용되었다는 증거입니다.

이런 알고리즘들을 구글의 **SentencePiece**나 허깅페이스의 **Tokenizers** 라이브러리가 빠르고 정확하게 구현해내고 있습니다.

## 04. 특수 도메인 응용: 화학 언어 모델 (Chemical LMs)

[스크립트]

토크나이저가 일반 언어를 넘어 '화학'과 같은 전문 분야에서 더욱 빛납니다.

예닮 조교님께 자문을 얻은 자료인데요

분자 구조를 텍스트로 표현한 SMILES나 InChI를 모델에 학습시킬 때

전용 토크나이저가 필요합니다.

특히 **InChI 기반 화학 모델**이 굉장합니다.

화학식, 연결성, 입체화학 등 계층 구조별로 '/ '를 기준으로 다중 토큰화**,**

**멀티 토크나이제이션**을 진행합니다.

**분자의 로컬한 특징**과 **구조적 의존성**을 동시에 학습할 수 있습니다.

텍스트 처리 기술이 신약 개발이나 재료 과학의 혁신을 이끄는 도구가 된 것이죠.

## 05. 결론 및 시사점 (Summary)

[스크립트]

오늘 내용을 정리하겠습니다.

첫째, 토크나이저는 모델의 지능뿐만 아니라 처리 속도와 비용에도

직접적인 영향을 미칩니다.

둘째, 특히 한국어 사용자 입장에서 영어 중심의 토크나이저는

매우 비효율적일 수 있으므로,

한국어 특화 토크나이저 설계가 반드시 병행되어야 합니다.

셋째, 앞으로의 AI는 텍스트를 넘어 화학, 생물, 코드 등

다양한 데이터 도메인에 최적화된 '전용 토크나이저'를 통해 더욱

전문화될 것이라 봅니다.

많은 연구가 진행되고 있기도 하고요.

다음장에 예상 질답을 적어 놓았습니다

혹시 다른 질문 있으시면 ~ 답변해 드리겠습니다.