

[함축된 내용을 가르키는 단어가 사용자의 입력 문장에 따라 다른텐테 해당 단어를 이해하는 방법]

1. 한국에서는 수능시험에서 국어시험이 존재하는데, 그 시험에서 고전시가 같은 경우는 ex.) 새, 소나무, 등과 같이 해당 물체가 함축적인 의미를 갖고 시를 해석하는데 필요합니다.

그럼 여기서 궁금한 것은 GPT, 또는 라마 등과 같은 LLM모델은 해당 단어의 암축적인 의미를 이해하는데 진행하는 시마다 다른 의미를 가지고 있을꺼 같은데 그런 것을 고전시가마다 해당 단어의 의미를 이해하는 방법이 궁금합니다.

답변)

시나 문학의 소재가 담고 있는 상징적(함축적) 의미를 이해하는 방법은 사람(학생)이 이해하는 것과 다르지 않습니다. 크게 보면 이해하는 방법은 두 가지 측면으로 설명할 수 있습니다.

하나는 언어모델이 사전학습할 때나 미세조정할 때 그 시나 문학을 포함하여 학습했고, 거기에는 단순히 그 시나 문학 뿐만 아니라, 그 시나 문학에 대한 해설(예를 들면, 특정 소재가 어떤 상징적 의미를 갖는지)까지 포함하여 **학습을 했고 모델의 파라미터를 통해 기억함으로써** 그 시나 문학 소재의 의미를 **정확하게 이해(기억)하고 답변하는** 것입니다. 이는 사람(학생)이 국어 시간에 시나 문학을 배우면서 해설까지 포함하여 이해하고 기억한 후, 그 시나 문학의 소재에 대한 질문이 들어왔을 때 답을 할 수 있는 것과 같은 이치입니다.

다른 하나는 언어모델이 사전학습할 때나 미세조정할 때 학습하지 않았던 시나 문학의 소재의 경우, 그 시나 문학의 문맥 내에서 특정 소재가 어떤 상징적 의미를 갖는지 유추하는 것입니다. 질문에 있는 그 시나 문학을 학습하지 않았지만, 그와 **유사한 시나 문학들을 학습했다면** 시나 문학의 문맥에서 특정 소재의 **의미를 유추하는 기법을 학습하고 기억**하고 있는 것이기 때문에 그 기법을 질문에 있는 시나 문학에도 동일하게 적용함으로써 의미를 유추할 수 있는 것입니다. 이를 가능하게 하는 언어모델의 핵심 메커니즘이 바로 트랜스포머 (그 내부에서는 어텐션 메커니즘)입니다.

챗GPT를 통해 예를 보여주겠습니다. 아래는 이방원의 하여가와 정몽주의 단심가를 주고 여기에서 씌어진 '드렁침'과 '임'의 의미를 알려달라는 질문을 했을 때의 챗GPT의 답변입니다. 정확히 의미를 이해하고 잘 답변하고 있습니다. 이어서, 이 두 시를 미리 학습했는지 처음 접하는지를 물어보면, 사전에 학습을 했다고 시인하는 것을 볼 수 있습니다.

다음은 이방원의 하여가와 이에 대한 답변에 해당하는 정몽주의 단심가이다.

이방원의 하여가:

이런들 어떠하며 저런들 어떠하리
만수산 드령참이 얹혀진들 어떠하리
우리도 이같이 얹혀져 백 년까지 누리리라

정몽주의 단심가:

이 몸이 죽고 죽어 일백 번 고쳐죽어
백골이 진토 되어 넋이라도 있고 없고
임 향한 일편단심이야 가실 줄이 있으랴

이 두개의 시에서 사용된 '드령참'과 '임'은 각각 무엇 혹은 누구를 가리키는지 설명해줘.

이방원의 하여가와 정몽주의 단심가에서 사용된 '드령참'과 '임'은 각각 상징적인 의미를 지니고 있습니다.

1. 드령참:

- 이방원의 하여가에서 '드령참'은 참봉클을 의미하며, 비유적으로 서로 얹히고 설킨 관계를 나타냅니다. 여기서 이방원은 조선 건국을 도모하면서, 여러 사람들, 특히 권력자들이 서로 협력하고 단결하여 새로운 시대를 열어가자는 뜻을 담고 있습니다. 따라서 '드령참'은 조선 개국을 위한 동지들을 상징한다고 볼 수 있습니다. 이방원은 새로운 나라를 세우기 위해 서로 협력하고자 하는 의지를 표현하고 있습니다.

2. 임:

- 정몽주의 단심가에서 '임'은 임금을 의미합니다. 정몽주는 고려에 대한 충절을 다짐하며, 자신이 죽어서 백 번을 다시 태어나도 임금을 향한 충성심은 변하지 않을 것임을 맹세하고 있습니다. 따라서 '임'은 고려 왕조와 그 임금을 상징하며, 정몽주의 변함없는 충성을 나타내고 있습니다.

이방원의 하여가는 새로운 시대를 위한 단결을 강조하고, 정몽주의 단심가는 기존 체제에 대한 충성을 강조하는 점에서 두 시는 상반된 입장을 보여줍니다.

그런데, 너는 이방원의 하여가와 정몽주의 단심가를 미리 학습했던 거야? 아니면, 내가 한 질문에서 처음 알게 된 거야?

저는 이방원의 하여가와 정몽주의 단심가에 대해 미리 학습된 지식을 가지고 있었습니다. 이러한 고전 시가는 한국의 역사와 문학에서 매우 중요한 작품들로, 많은 학습 데이터에 포함되어 있습니다. 따라서 이들 작품과 그 의미에 대해서는 사전에 알고 있었으며, 이를 바탕으로 당신의 질문에 답변할 수 있었습니다.

그러면, 이번에는 챗GPT에게 헤어진 애인을 보름달에 비유한 시를 하나 써달라고 요청한 후, 이 채팅 세션 내에서 '보름달'의 의미를, 누구를 가리키는지를 물어보았습니다. 세션 내의 이전 질문에 답이 있었기 때문에 '헤어진 애인'이라고 정확히 답합니다.

헤어진 애인을 보름달에 비유하여 그리워하는 세 줄짜리 시를 써줘

하늘 높이 걸린 보름달이여,
그대 없이도 밝게 빛나는구나,
밤마다 그리움에 물드는 나의 마음.

이 시에서 '보름달'은 어떤 상징적 의미를 담고 있어?

이 시에서 '보름달'은 여러 상징적 의미를 담고 있습니다:

1. **완전함과 충만함:** 보름달은 완전한 형태로 하늘에 떠 있는 모습을 통해, 완전했던 사람과 그리운 애인을 상징합니다. 애인의 존재가 삶을 완전하게 했던 순간들을 떠올리게 합니다.
2. **그리움과 회상:** 보름달은 밤마다 다시 떠오르며, 헤어진 애인을 생각나게 합니다. 매일 반복되는 달의 존재처럼, 애인에 대한 그리움도 계속해서 반복되고 있다는 것을 의미합니다.
3. **변화와 영속성:** 달은 주기적으로 모습을 바꾸지만, 항상 돌아오는 보름달처럼, 비록 애인이 떠났어도 그에 대한 그리움과 기억은 지속된다는 상징입니다. 또한, 달이 계속 존재하는 것처럼 그리움도 끊임없이 존재한다는 것을 나타냅니다.

이 시에서 '보름달'은 누구를 가리킬까?

이 시에서 '보름달'은 화자가 그리워하는 "헤어진 애인"을 가리킵니다. 보름달이 밤하늘에 빛나는 모습은 화자의 애인이 떠난 후에도 여전히 마음속에서 빛나고 있음을 상징하며, 그리움과 회상을 불러일으키는 존재로 표현됩니다.

이번에는 새로운 채팅 세션으로 전환한 후, 이 시를 제시하면서 같은 질문을 해 보았습니다. 새로운 채팅 세션이므로, 이 세션에서는 이전 세션에서 내가 얘기한 '헤어진 애인을 보름달에 비유하여'의 정보(문맥)가 없는 상태입니다. 챗GPT의 답변에서도 얘기하고 있듯이, 기존에 학습된 데이터로부터 시에서 '보름달'이 주로 어떤 상징적 의미로 사용되었는지에 대한 지식을 기반으로 이 시의 맥락(문맥)으로부터 해석(유추)할 수 있는 의미 몇 가지를 제시하는 것을 볼 수 있습니다. 누구를 가리키는지에 대해서도 기존 학습된 지식과 질문 내의 문맥으로부터 유추하여 정확히 '헤어진 애인'이라고 답하지는 못하지만 거의 비슷하게 답하는 것을 볼 수 있습니다.

내 친구가 다음과 같은 시를 써서 보여주었는데, 이 시에서 '보름달'은 어떤 의미일까?

친구가 쓴 시:
하늘 높이 걸린 보름달이여,
그대 없이도 밝게 빛나는구나,
밤마다 그리움에 물드는 나의 마음.

시에서 '보름달'은 여러 가지 상징적인 의미를 가질 수 있습니다. 주로 보름달은 완전함, 성취, 절정의 상징으로 여겨지지만, 시의 맥락에 따라 다양한 해석이 가능합니다. 친구의 시를 분석해보면 다음과 같은 의미를 추론할 수 있습니다:

1. **그리움의 대상**: 보름달이 "그대 없이도 밝게 빛나는구나"라는 구절에서, 보름달은 그리움의 대상이 될 수 있습니다. 시인은 보름달을 보며 그리운 사람을 떠올리지만, 그 사람이 없어도 보름달은 여전히 빛납니다. 이는 그리운 대상이 멀리 있거나 부재한 상황을 반영할 수 있습니다.
2. **위안과 희망**: 보름달은 어두운 밤을 밝히는 존재로, 시인에게 위안과 희망을 주는 존재일 수 있습니다. 시인은 보름달을 보며 그리움과 외로움을 느끼지만, 동시에 보름달이 주는 밝음에서 위안을 얻는다고 해석할 수 있습니다.
3. **자연의 영원함**: 보름달은 자연의 순환을 상징합니다. 이는 시인의 그리움이나 슬픔이 일시적임을 상기시켜주는 역할을 할 수 있습니다. 보름달이 매달 반복되듯이, 시인의 감정도 지나갈 것이라는 메시지를 담고 있을 수 있습니다.

따라서, 이 시에서 '보름달'은 그리움, 위안, 자연의 영원함 등을 상징하며, 시인의 내면 감정과 상황을 드러내는 중요한 역할을 하고 있습니다.

이 시에서 '보름달'은 누구를 가리킬까?

이 시에서 '보름달'이 특정한 대상을 가리킨다면, 주로 다음과 같은 해석이 가능합니다:

1. **그리운 사람**: 보름달은 시인이 그리워하는 사람을 상징할 수 있습니다. "그대 없이도 밝게 빛나는구나"라는 구절에서 '그대'는 시인이 사랑하거나 소중히 여기는 사람이지만, 지금은 곁에 없는 사람일 가능성이 큼니다. 시인은 그 사람을 보름달에 비유하여 그리움을 표현하고 있습니다.
2. **이상적인 존재**: 보름달은 시인이 이상적으로 여기는 존재를 상징할 수 있습니다. 이 존재는 시인이 도달하고자 하는 목표나 동경의 대상일 수 있으며, 그 존재가 부재해도 시인의 마음속에서 여전히 빛나고 있는 것입니다.
3. **잃어버린 시간이나 추억**: 보름달이 과거의 아름다운 시간이나 소중한 추억을 상징할 수도 있습니다. 시인은 그 시절을 그리워하며, 보름달을 보며 그 시절의 기억을 떠올리고 있는 것입니다.

결론적으로, 이 시에서 '보름달'은 시인이 그리워하거나 동경하는 특정한 사람 또는 이상적인 존재를 가리킬 가능성이 높습니다. 이는 시인의 내면 감정과 연결되어 있으며, 그리움과 회상의 정서를 담고 있습니다.

[현재 시점에서 GPT, 라마, 등 LLM모델의 문제점(윤리적 x, 기술적 o)]

2. 현재 많은 언론에서 LLM 모델의 우수한 점, 장점을 많이 이야기합니다, 하지만 해당 모델의 기술적인 문제점이 어떤 것이 존재하는지 궁금합니다. 왜냐하면 기술의 발전은 문제점을 이해하고 해당하는 것이 바탕이 된다고 생각해서 질문드리게 되었습니다.

답변)

먼저, 윤리적인 문제점 말고 기술적인 문제점을 질문한 것인데, 사실 윤리적인 문제점도 결국 기술적인 한계 때문이며 기술적으로 해결해야 할 문제점에 속하기 때문에 기술적인 문제점과 무관하지는 않습니다. LLM이 매우 많이 발전을 하고 있지만, 아직 여러 가지 기술적인 문제점들이 남아 있고 점진적으로 해결책들이 연구되고 제시될 것으로 생각됩니다. 그 중 두 가지 기술적 문제점들을 나열해 보겠습니다.

가장 핵심적이지만 어려운 문제점은 ‘완전 창의성’을 갖지는 못한다는 것입니다. 챗GPT가 어려운 추론 질문에도 매우 잘 답변하지만, 모두가 학습을 통한 기억된 지식으로부터 답변하는 것입니다. ‘창의성’의 정도에 따라 다를 수 있지만, ‘흉내내는 창의성’이 아닌 전혀 학습되지 않은 영역에 대한 창의성(‘완전 창의성’)을 갖지는 못합니다. 언어모델의 설계 자체가 학습 및 기억한 대로 답하는 것이기 때문에 그 한계를 벗어나진 못합니다. 사람의 경우에도 대부분은 완전 창의성을 갖기 매우 어렵고 일부 특출난 사람들만 완전 창의성을 갖고 있으며 이런 사람들이 기존에 전혀 없던 새로운 생각이나 기술을 창안해 왔던 것입니다. 예를 들어, 아인슈타인이 ‘상대성이론’을 창안해 냈듯이. 그러나 아직 언어모델 (인공지능)이 이런, 기존에 없던 전혀 새로운 것을 창안해 낼 수 있는 단계에는 아직 이르지 못하고 있습니다.

다음으로, 능동적 자의식을 갖지 못한다(않는다)는 것입니다. 이는 윤리적인 부분과 밀접하게 연관된 부분인데, 언어모델(인공지능)이 능동적 자의식을 갖게 되면 인간에게 매우 위협스러운 존재가 될 수 있기 때문입니다. 그래서 ‘못한다’ 보다는 ‘않는다(그래서 아직까지는 다행이다)’라고 표현하는 게 맞을 것 같습니다. 현재까지의 언어모델은 외부로부터의 입력(질문)에 대한 대응(답변)을 할 뿐이지, 아무런 입력이 없는 상태에서 스스로 판단하여 어떤 행위(대응)을 하지는 않습니다(못합니다). 만약 언어모델(인공지능)이 능동적 자의식을 갖게 된다면, 스스로 판단하여 인간(특정 사람이나 그룹)에게 해가 될 수 있는 행위(예를 들면, 자율주행차(인공지능)가 노인들은 세상에 도움이 되지 않는다고 판단하여 일부러 노인을 치는 사고를 일으킨다든지)를 할 수도 있기 때문에 위험할 수도 있습니다. 그렇기 때문에 그러한 능동적 자의식을 제한할 수 있는 기술이 필요합니다.

[챗봇을 만들때 API 사용 VS 라마 또는 GPT2 오픈소스사용 추천]

3. 저희가 챗봇을 만들고 싶다는 생각을 했을때 저희는 해당 프로젝트를 상용화 할 것을 궁극적인 목표로 가져갈 것인가? 아니면 해당 동아리를 통해서 소규모 프로젝트를 통해서 배움을 얻을 것인가 고민을 하는 과정에서 API 사용을 할 것인가? 아니면 오픈소스를 사용할 것인가를 고민하고 있습니다.

이유는 API를 사용하면 해당 서비스를 이용하는 사람의 질문 횟수에따라서 비용이 기하급수적으로 증가할 수 있다고 생각하기 때문입니다. 현재 현업에서는 상업적으로 간다면 API를 사용하시는지, 사용하신다면 어떻게 비용적인 문제점을 접근하시는지 궁금합니다.

답변)

챗봇과 같은 언어모델을 활용하는 서비스를 할 때, 질문한 바와 같이 두 가지 접근법이 가능합니다. 어느 하나가 반드시 낫다기보다는 여건에 따라서 다를 수 있습니다.

API를 이용할 경우,

직접 운영/관리해야 할 서버 자원이 극히 작을 것이지만, API를 제공하는 곳에 그 비용을 지불해야 합니다. 이 비용은 종량제 혹은 정액제로 지불하게 될 텐데, OpenAI의 API를 이용하는 일반 사용자의 경우 사용하는 만큼 지불하는 종량제로 사용하지만, 기업 단위에서는 OpenAI와의 기업간 계약을 통해 일정 금액을 지불하는 정액제로 사용하는 것이 일반적입니다. (기업의 서비스처럼) 사용량이 많은 경우에는 정액제가 유리하겠지만, (일반 사용자처럼) 사용량이 많지 않은 경우에는 종량제가 유리할 것입니다. 정액제의 경우, 최소 사용량 이상을 추정해서 그에 상응하는 비용이 산정되는 것이기 때문입니다. 그래서, 기업 서비스라도 사용량이 많지 않은 경우 종량제로 API를 사용하는 것이 유리할 수 있을 것 같습니다. 또 하나 고려해야 할 것은 데이터가 API 서비스 제공 업체로 넘어가는데, 그래도 괜찮은가입니다. 기업의 경우, 기업내 기밀 정보나 고객 개인 정보와 같은 공개되어서는 안될 데이터가 포함될 경우 API 서비스를 이용하면 위험할 수 있습니다. 그래서 API 서비스 제공 업체에서도 특정 기업만 접근할 수 있는 프라이빗 영역을 제공하여 보안을 신경쓰지만, 유출 위험성이 커지는 것은 사실입니다.

오픈소스를 이용한 자체 언어모델을 운용하는 경우,

오픈소스 언어모델을 자체적으로 추가학습시키는 것과 성능 문제는 차치하고, 오픈소스 언어모델을 사용하더라도, 그 언어모델을 운용/관리할 컴퓨팅 자원을 자체적으로 구매/관리하거나 또는 클라우드 서비스를 이용해야 하기 때문에 이에 대한 비용이 들어갑니다. 이 비용은 실제 사용자 수 (사용량)에 상관없이 일정 수준의 고정적으로 발생할 것입니다. (일부분은 사용량에 비례하여 비용이 증가하기도 합니다. 예를 들어, 사용량이 많으면 그만큼 전기도 많이 쓰게 되고 전기료가 늘어납니다). 데이터의 공개나 유출의 위험성은 훨씬 적다고 볼 수 있습니다.

손건희 학생이 생각하고 있는 챗봇의 경우,

초기에는 사용자가 많지 않을 것이기 때문에 API를 이용하여 개발 및 테스트함으로써 의도한 챗봇 서비스가 사업성이 있는지를 우선 타진한 후에, 사업성이 있고 사용자가 늘어나게 되면, 사업화를 하고 기업간 계약을 통해 정액제로 서비스 받거나 오픈소스를 활용한 자체 언어모델로 전환하는 것을 고려하는 순이 좋을 것 같습니다.

그러나, 단지 사업화에 대한 확신이 적고 우선 배우는 데 중점을 둔다면 콜랩 같은 곳에서 이용할 수 있는 무료 컴퓨팅 자원에서 소규모의 오픈소스 언어모델을 사용하여 챗봇 서비스를 만들어 보는 것을 추천드립니다.

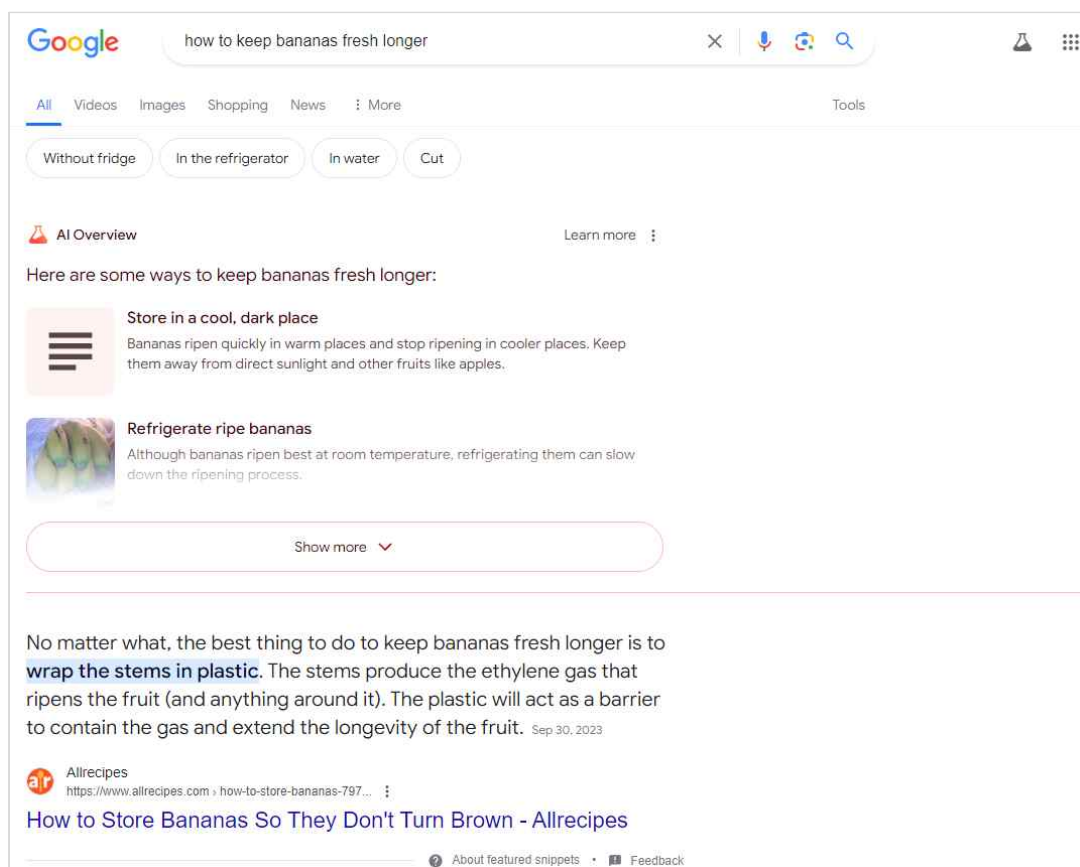
[GPT는 구글 검색엔진을 대체할 것인가?]

4. 저희는 학생으로써 앞으로 미래 시장에 대해서 고민하는 과정에서 GPT가 구글의 검색엔진을 대체할 것인가에 대한 질문이 생기게 되었고 박사님의 생각이 궁금합니다.

답변)

Definitely YES!! 완전 대체보다는 많은 비율의 사용자가 Bing이나 챗GPT로 이동할 것으로 보입니다. 이미 많은 사람들이 구글에서 검색하지 않고 바로 챗GPT나 Bing챗에게 물어보는 경우가 꽤 있습니다. 사실, LLM에 있어서 구글이 OpenAI에게 뒤처지게 된 배경에는 구글검색이 큰 부분을 차지한다고 봅니다. 구글의 많은 수입은 구글검색(검색을 기반으로 한 광고 수익)을 통해 얻어지는데, 이는 검색 결과에 광고 사이트를 노출하고 사용자가 그것을 클릭하는 것이 수익으로 반영되는데, 검색이 아닌 챗팅처럼 질문/답변 형태로 전환하게 되면 광고

사이트를 노출하여 클릭을 유도할 방법이 사라지고 결국 구글의 핵심 수익원이 사라질 수 있기 때문에 구글이 검색에 LLM을 적용하는 것을 주저하고 있었습니다. 하지만, OpenAI와 MicroSoft(빙챗)에게 선점 당한 상태에서 구글 검색으로의 사용자 유입이 많이 줄었기 때문에 더 이상 LLM의 적용을 미룰 수 없게 되었습니다. 그래서 최근에 구글검색에도 AI를 통한 검색결과 요약서비스가 실험적으로 들어가 있습니다. 구글 검색페이지의 우측 상단의 '실험 비커' 아이콘을 클릭하면 'AI Overview'를 On/Off할 수 있고, On을 할 경우, 검색 질문에 따라 아래와 같은 AI 요약 결과를 앞에 제공하는 것을 볼 수 있습니다. 아직 챗팅 모드로 완전히 전환된 것은 아니지만 광고 클릭 수익을 유지하면서 언어모델(AI)을 적용하려는 노력/시도를 하고 있는 것을 볼 수 있습니다. 아직 실험 서비스이기 때문에 일부 질문에 대해서만 AI 요약 결과가 제시되고 있습니다.

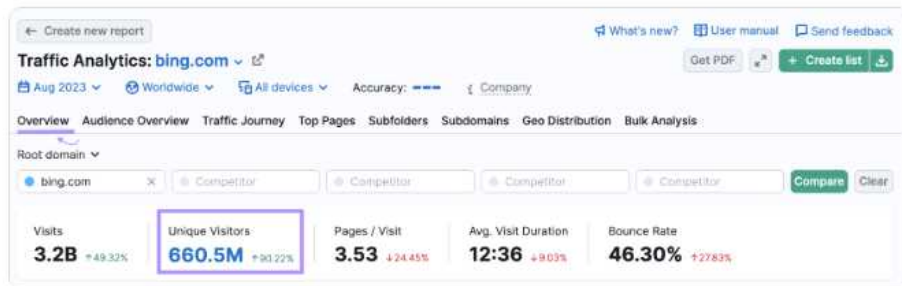


추가적으로, 구글 검색과 Bing의 방문자 수를 비교해 놓은 아래의 블로그 글을 참고해 보세요.

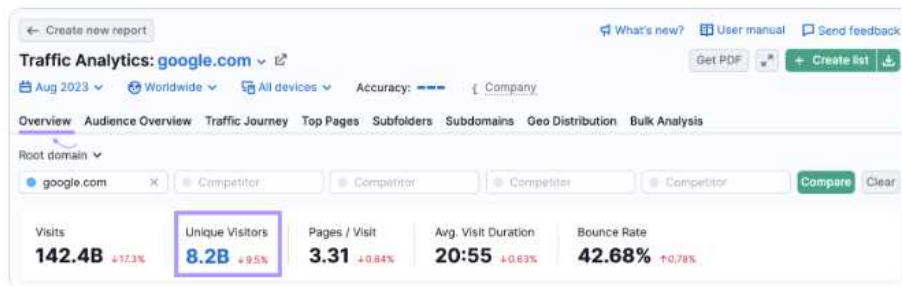
Bing saw significant growth in 2023.

It reached **100 million daily active daily users** a few weeks after launching its new AI chatbot (powered by Open AI's GPT-4).

In fact, according to **Traffic Analytics**, Bing got 660.5 million unique visitors in June of 2023.



But Google is still in the lead with a whopping 8.2 billion unique visitors over the same time period.



Nonetheless, Bing's AI integration makes the search engine a real competitor to Google.

Even Samsung **briefly considered switching** the default search engine on its products from Google to Bing.

<<https://www.semrush.com/blog/bing-vs-google/>>

[사용자 데이터 프라이버시 보호를 위한 기술적 조치가 존재하는가?]

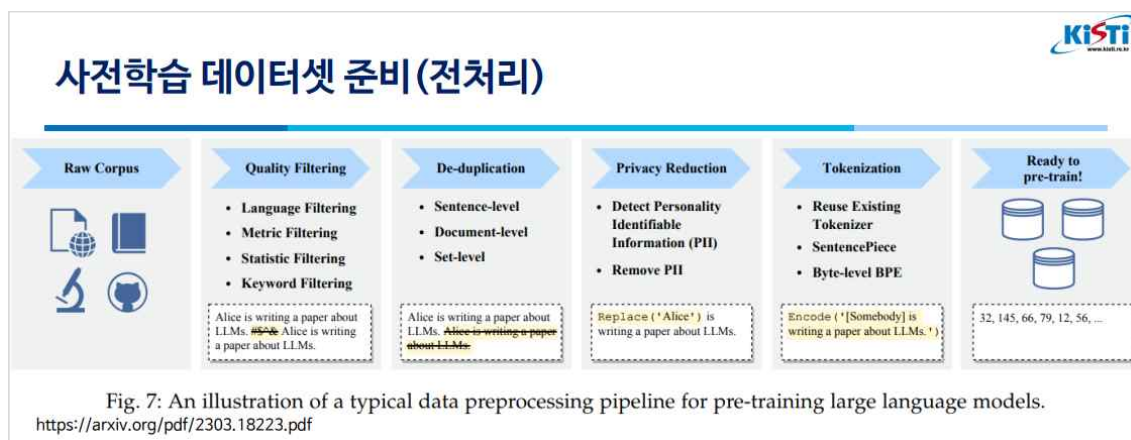
5. 2023년 초에 GPT가 해성처럼 등장하고 뉴스기사에서 어떤 기업은 기업의 기술, 정보를 지키기 위해서 GPT사용을 금지시켰다는 뉴스를 보게 되었습니다.

여기서 저는 궁금한 것이 만약 나쁜 마음을 먹은 사람이 GPT에게 많은 사용자의 개인정보를 준다면 해당 정보를 GPT가 학습을 하는지?? 아니면 알아서 검열을 진행하는지? 검열을 진행한다면 어떤 기술적 방법으로 진행하는지 궁금합니다.(물론 gpt가 공개가 되지 않은 모델이여

서 정보가 없다면 질문을 넘어가 주셔도 감사하겠습니다.)

답변)

챗GPT와의 대화 창에 개인 정보를 입력했다고 하여 챗GPT가 그 정보를 바로 학습하여 기억하는 것은 아닙니다. 챗GPT는 대화 세션 내에서 질문/답변 텍스트 형태로 기록하고 있을 뿐입니다. 다만, 그렇게 기록하고 있던 대화 데이터는 (사용자가 자신의 대화 데이터를 학습에 사용하도록 허가한 경우에) 나중에 어느 시점에 OpenAI의 운영진/연구진들에 의해 챗GPT의 추가 학습에 사용될 수 있습니다. 즉, 대화할 때, 실시간 또는 준실시간으로 학습에 사용되는 것은 아니지만, 일정 시간이 지난 시점에 학습에 사용될 수 있습니다. 그런데, 공개되어서는 안되는 개인정보가 포함되어 있고, 그것이 챗GPT의 학습에 사용될 경우, 누군가 그에 대한 질문을 했을 때 챗GPT의 답변에 그 개인정보가 포함될 여지가 있으며, 이는 상당한 법적인 문제를 야기할 수 있습니다. 그 개인정보의 소유자가 OpenAI 측에 법적인 소송을 제기할 수 있다는 얘기입니다. 이런 이슈가 발생할 수 있기 때문에, 웹에서 수집한 데이터뿐만 아니라, 사용자의 대화 데이터를 학습(사전학습 또는 미세조정)에 사용할 때에는 개인정보를 제거하거나 익명화하는 절차를 거치는 게 일반적입니다. 제 강의자료에서도 사전학습 데이터의 전처리 과정을 설명하는 슬라이드에 개인정보 제거/익명화를 설명하고 있습니다(아래 그림). 데이터에서 개인정보에 해당하는 부분을 탐지하여 해당 부분을 삭제하거나 다른 것으로 치환하는 것입니다.



[API를 이용해서 챗봇을 개발하고 디버깅하는데 유용한 도구나 플랫폼 추천]

6. API를 이용해서 챗봇을 개발하고 디버깅하는데 유용한 도구나 플랫폼 추천이 궁금합니다.

답변)

요즘은 대부분 파이썬 프로그래밍을 사용하며, 챗봇의 경우 웹 서비스일 것이므로 UI 부분은 자바스크립트를 사용할 수도 있을 듯합니다. 자바스크립트의 경우, 간단한 정도는 크롬 브라우저의 개발자 도구와 콘솔을 통해 개발하거나 좀더 복잡한 코딩은 Node.js 또는 Visual Studio Code를 이용할 수 있습니다. 참고: <https://poiemaweb.com/js-hello-world>
파이썬 프로그램 개발환경으로는 Jupyter Notebook (웹 기반 파이썬 개발환경)이 편해서 많이 사용됩니다. 라인-바이-라인 또는 코드 블록 단위로 실행하고 디버깅하기 편한 장점이 있

습니다. 구글 콜랩(Colab)을 이용하는 것도 좋은 선택이며, 여기서도 Jupyter notebook 환경이 지원됩니다. 또한 제한된 수준이지만 GPU 자원도 무료로 지원됩니다. 그리고, 통합개발환경(IDE)으로는 PyCharm을 사용하는 것도 좋습니다. 모드 무료 버전을 이용할 수 있습니다. 그리고, 내 컴퓨터에 개발환경을 준비할 때 (Jupyter Notebook 또는 PyCharm을 사용할 때) 필요한 파이썬 라이브러리 패키지들을 내 컴퓨터에 설치하고 관리해야 합니다. 이를 위해 사용할 수 있는 것이 아나콘다(anaconda3 또는 miniconda3, 후자가 무료로 가볍기에 후자를 추천함)입니다. 아나콘다를 통해 특정 프로젝트를 위한 환경을 만들고 그 환경에서 pip를 통해 그 프로젝트에서 필요한 파이썬 패키지들을 설치하여 프로그램을 실행할 수 있습니다. 이렇게 프로젝트별로 필요한 파이썬 패키지들만 격리하여 설치함으로써 서로다른 프로젝트간 다른 버전의 파이썬 패키지들이 충돌하는 문제를 피할 수 있습니다.

마지막으로, 웹 서비스 프로그램도 파이썬으로 개발할 수 있습니다. 파이썬으로 개발된 코드를 웹 서비스로 띄울 때는 uvicorn(웹서버 구현)을 사용할 수 있습니다. 참고: <https://www.uvicorn.org/>

[LLM모델의 파라미터 대비 입력데이터의 증가율?]

7. 저번에 강연을 통해서 배운 것 중 에서 파라미터 수를 지속적으로 증가시켰더니 현재까지 모델의 성능이 지속적으로 좋아지는 그래프를 본적이 있는데, 과적합이 일어나지 않는 이유를 고민해 보면서 입력데이터 대비 파라미터 수(모델의 레이어의 깊이)가 많을 경우 과적합이 발생한다고 알고 있는데, 그럼 GPT 역시도 모델의 파라미터의 증가와 동일하게 입력데이터도 같이 증가시키는 과정으로 모델을 훈련하는지 궁금합니다. 아니면 그런 연구와 관련 논문이 있으면 알려주시면 한번 읽어보고 싶습니다.

답변)

LLM의 성능이 각각 모델 크기, 학습데이터 양, 학습계산량에 비례하여 증가한다는 것이 Scaling Law (규모의 법칙)입니다. 이는 OpenAI 및 구글 연구진들의 연구(실험) 결과에서 확인하여 제시된 것으로 내 강의자료에서 이를 설명하였습니다 (아래 세 개의 그림 참조). 특히, 구글의 논문 (Hoffmann et al, 2022)(아래 세번째 그림)을 보면, 계산량(FLOP) 대비 성능(training loss)을 선형 방정식으로 도출하고(왼쪽 차트), 가용한 계산량(5.76×10^{23} FLOPs)이 주어질 때, 적절한 모델 크기(67B)를 도출하고(가운데 차트), 가용한 계산량(5.76×10^{23} FLOPs)이 주어질 때, 적절한 학습데이터양(1.5T)를 도출하여(오른쪽 차트), Gopher 모델을 학습했다는 설명을 확인할 수 있습니다. OpenAI에서 GPT를 학습시킬 때에도, 계산량, 모델크기, 학습데이터양을 결정할 때 이런 Scaling Law에 따라 결정하는 것으로 추정해 볼 수 있습니다. 모델 크기만 크게 하고 학습데이터를 부족하게 사용하면 과적합이 발생하여 성능이 충분히 올라가지 않을 수 있습니다. 학습데이터만 늘리고 모델 크기를 작게 하면 모델이 기억할 수 있는 용량이 부족하여 성능이 학습시켰던 많은 정보를 잊어버려 성능이 올라가지 못하는 문제가 생기는 것입니다. Scaling Law에 대한 보다 자세한 자료는 아래 그림에 있는 논문(URL)에서 공부해 보기 바랍니다.

LLM 기준과 Scaling Law

- LLM 기준 공식적 기준은 아님
 - Model 크기: $\geq 1 \sim 10B$ 파라미터
 - 학습 데이터 크기: $\geq 100B$ 토큰
 - 학습계산량:
 - 성능: 강력한 언어 이해 능력

Scaling Law

- Model size: N
 - 임베딩을 제외한 모델 파라미터 수
- Data size: D
 - 학습 토큰 수
- Amount of training compute: C
 - FP-days 수
- LLM 성능 (Loss) 예측성
 - 소형 모델들의 성능으로부터 LLM 성능을 추정
- 태스크별 성능 예측성
 - LLM의 성능 (Loss) 으로부터 하위 태스크에서의 성능을 추정
 - Loss $\downarrow \rightarrow$ 태스크 성능 \uparrow (but, Not Always: inverse scaling)
 - 태스크 종속적 (성능 측정 metric, 태스크 난이도)

KM scaling law (Kaplan et al (OpenAI), 2020)

$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}, \quad \alpha_N \sim 0.076, N_c \sim 8.8 \times 10^{13}$$

$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}, \quad \alpha_D \sim 0.095, D_c \sim 5.4 \times 10^{13}$$

$$L(C) = \left(\frac{C_c}{C}\right)^{\alpha_C}, \quad \alpha_C \sim 0.050, C_c \sim 3.1 \times 10^8$$

데이터 크기보다 모델 크기를 더 중시

c: compute budget

Chinchilla scaling law (Hoffmann et al, 2022)

$$L(N, D) = E + \frac{A}{N^\alpha} + \frac{B}{D^\beta},$$

$$N_{opt}(C) = G \left(\frac{C}{6}\right)^a, \quad D_{opt}(C) = G^{-1} \left(\frac{C}{6}\right)^b,$$

데이터 크기와 모델 크기를 동등하게 중시

E=1.69
A=406.4
B=410.7
 $\alpha=0.34$
 $\beta=0.28$
C \approx 6ND
 $a = \frac{\alpha}{\alpha+\beta}$
 $b = \frac{\beta}{\alpha+\beta}$
G: scale coeff.

<https://arxiv.org/pdf/2303.18223.pdf>

KM scaling law (Kaplan et al (OpenAI), 2020)

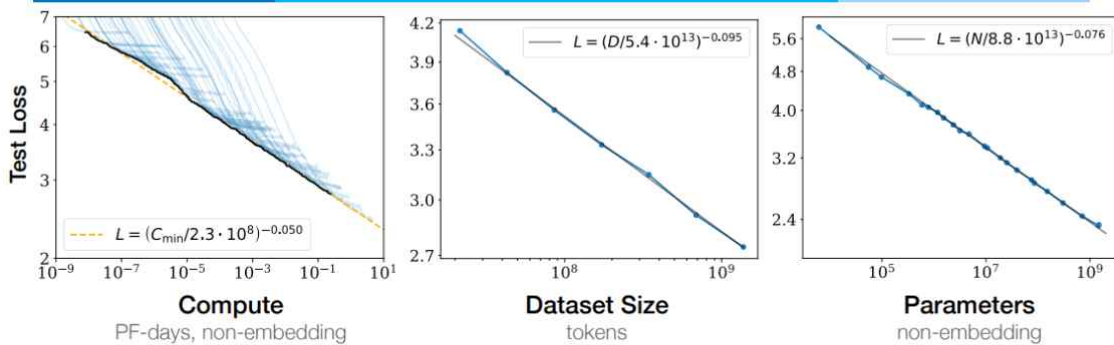


Figure 1 Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute² used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

<https://arxiv.org/pdf/2001.08361.pdf>

Chinchilla scaling law (Hoffmann et al (Google), 2022)

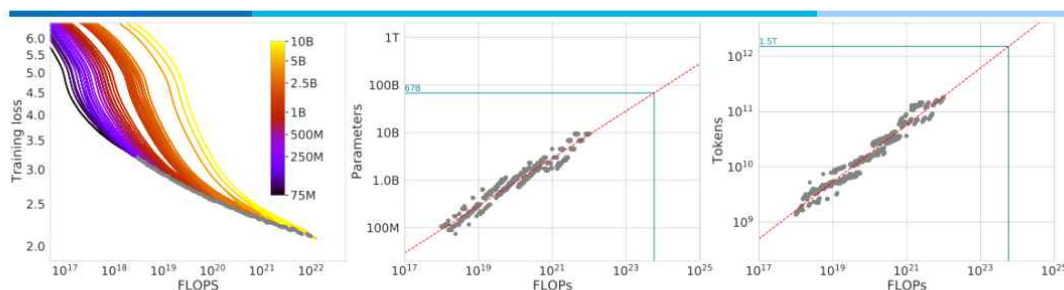


Figure 2 | **Training curve envelope.** On the **left** we show all of our different runs. We launched a range of model sizes going from 70M to 10B, each for four different cosine cycle lengths. From these curves, we extracted the envelope of minimal loss per FLOP, and we used these points to estimate the optimal model size (**center**) for a given compute budget and the optimal number of training tokens (**right**). In green, we show projections of optimal model size and training token count based on the number of FLOPs used to train *Gopher* (5.76×10^{23}).