SPLARE / Harness Engineering

2026 세미나

장영준

Table Of Contents

1. [SPLARE] Learning Retrieval Models with Sparse Autoencoders

2. Harness design for long-running application development

LEARNING RETRIEVAL MODELS WITH SPARSE AUTOENCODERS

Thibault Formal*
Naver Labs Europe
thibault.formal@gmail.com

Maxime Louis*
Naver Labs Europe
maxime.louis.x2012@gmail.com

Hervé Déjean
Naver Labs Europe
herve.dejean@naverlabs.com

Stéphane Clinchant
Naver Labs Europe
stephane.clinchant@naverlabs.com

Preliminaries

SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking

Thibault Formal
Naver Labs Europe
Meylan, France
Sorbonne Université, LIP6

Benjamin Piwowarski
Sorbonne Université, CNRS, LIP6
Paris, France
benjamin.piwowarski@lip6.fr

Stéphane Clinchant
Naver Labs Europe
Meylan, France
stephane.clinchant@naverlabs.com

SPLADE v2: Sparse Lexical and Expansion Model for Information Retrieval

Thibault Formal
Naver Labs Europe
Meylan, France
Sorbonne Université, LIP6

Benjamin Piwowarski
Sorbonne Université, CNRS, LIP6
Paris, France
benjamin.piwowarski@lip6.fr

SPLADE-v3: New baselines for SPLADE

Carlos Lassance
Cohere (*Work done while at Naver*)
cadurosar at gmail dot com

Hervé Déjean, Thibault Formal, Stéphane Clinchant
Naver Labs Europe
first.lastname at naverlabs dot com

Preliminaries-SPLADE

Motivation

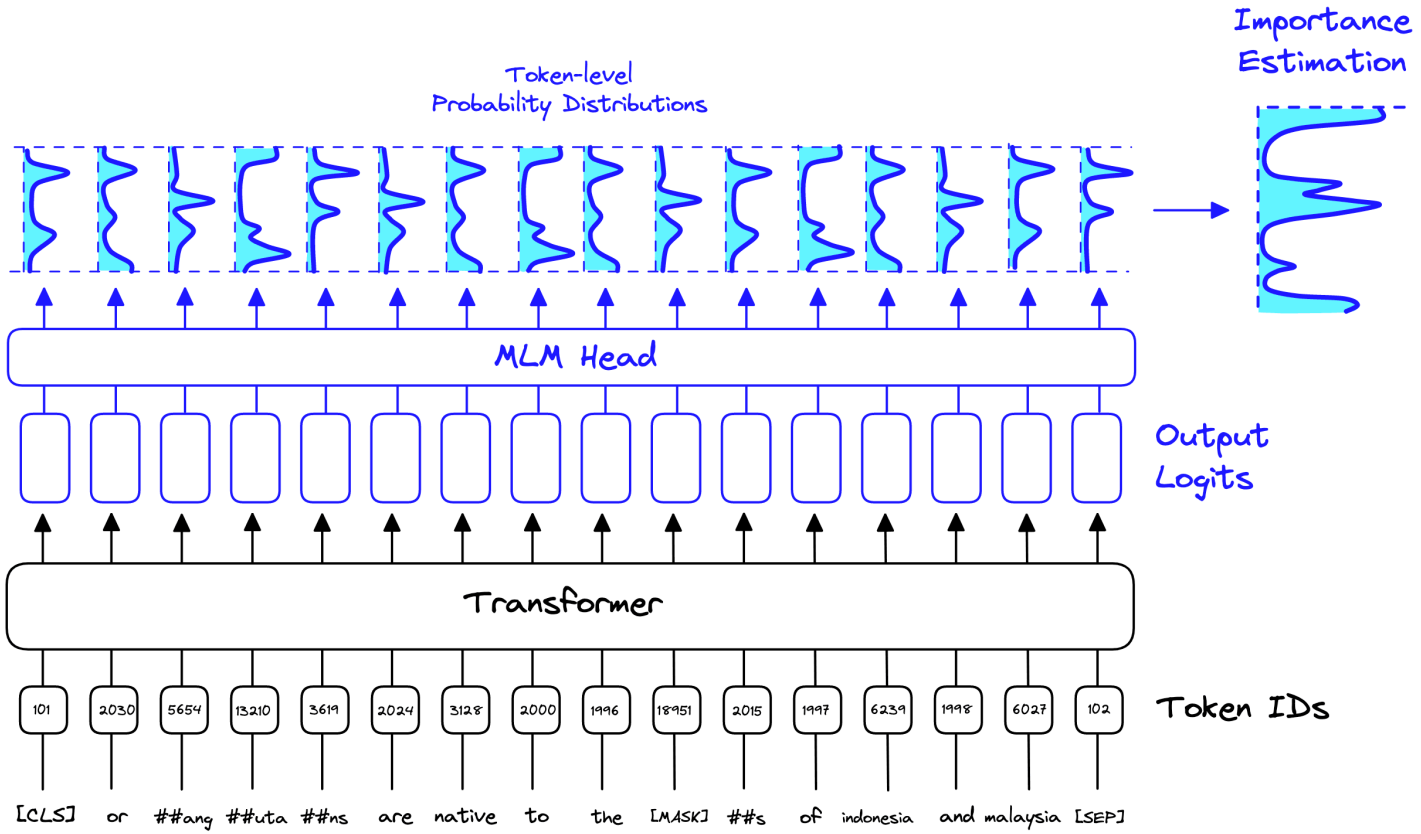
SparTerm(literal-only)	27.46	51.05	60.21	71.55	78.28	83.27	88.33	91.16
SparTerm(expansion-only)	19.8	40.93	-	63.42	70.96	77.62	84.81	89.08
SparTerm(expansion-enhanced)	**27.94**	51.95	61.58	72.48	78.95	84.05	89.5	92.45

- BM25의 Term Mismatch Problem (“파스타 식당”과 “스파게티 음식점”을 구분할 수 없음)
- 이를 해결하기 위한 이전 Sparse Encoder 모델 (SparTerm)이 너무 복잡하고, end-to-end로 학습할 수 없다.

Preliminaries-SPLADE

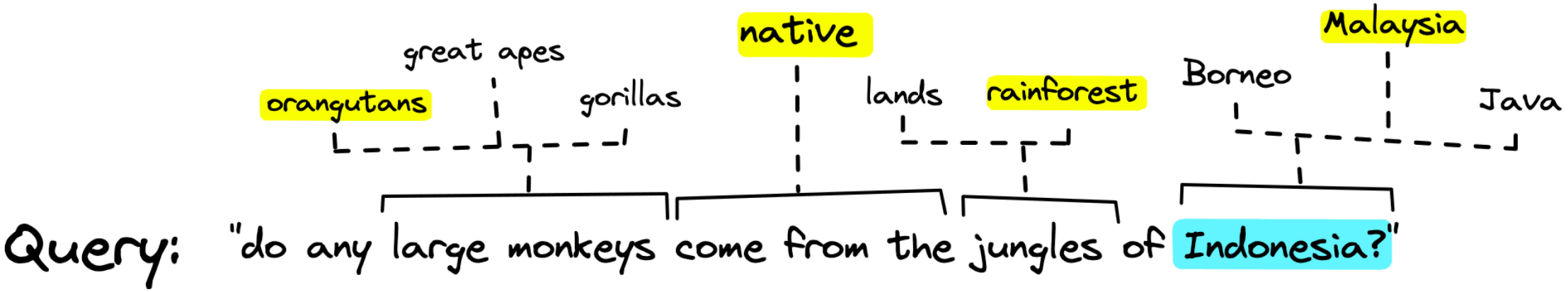
Method

- 최종 벡터: **입력 문맥이 가장 강력하게 지지하는 토큰들의 점수로 구성된, vocab 크기의 벡터**



Preliminaries-SPLADE

검색 상황 예시



with query expansion

without query expansion

Doc: "Orangutans are native to the rainforests of Indonesia and Malaysia"

(원래는 doc도 확장됨)

Preliminaries-SPLADE

Method

- **학습 방법**

1. Ranking Loss (Contrastive Loss)

$$\mathcal{L}_{rank-IBN} = -\log \frac{e^{s(q_i,d_i^+)}}{e^{s(q_i,d_i^+)} + e^{s(q_i,d_i^-)} + \sum_j e^{s(q_i,d_{i,j}^-)}}$$

2. Flops Loss (Sparse Loss)

- 개념: "배치 내에 있는 각 토큰의 평균 빈도 수를 계산하고, 이 값을 제곱하여 최소화하자"
- 효과: 불필요한 단어의 가중치를 0으로 보내버리고, 꼭 필요한 단어만 남김

$$\ell_{\text{FLOPS}} = \sum_{j \in V} \bar{a}_j^2 = \sum_{j \in V} \left(\frac{1}{N} \sum_{i=1}^{N} w_j^{(d_i)} \right)^2$$

Preliminaries-SPLADE

Method

- **최종 Loss**

$$\mathcal{L} = \mathcal{L}_{rank-IBN} + \lambda_q \mathcal{L}^q_{\text{reg}} + \lambda_d \mathcal{L}^d_{\text{reg}}$$

Preliminaries-SPLADE

Method

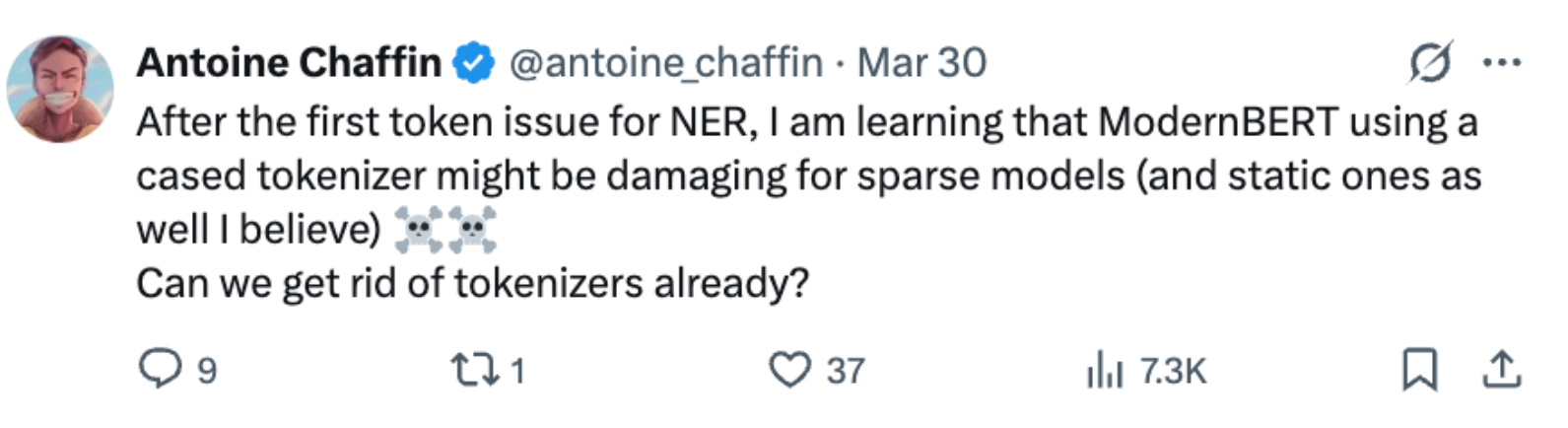
- **Decoding 예시**

```
Model: naver/splade-v3
Sentence: The weather is lovely today.
Decoded: [
  ('weather', 2.754288673400879),
  ('today', 2.610959529876709),
  ('lovely', 2.431990623474121),
  ('currently', 1.5520408153533936),
  ('beautiful', 1.5046082735061646),
  ('cool', 1.4664798974990845),
  ('pretty', 0.8986214995384216),
  ('yesterday', 0.8603134155273438),
  ('nice', 0.8322536945343018),
  ('summer', 0.7702118158340454)
]
```

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Motivation

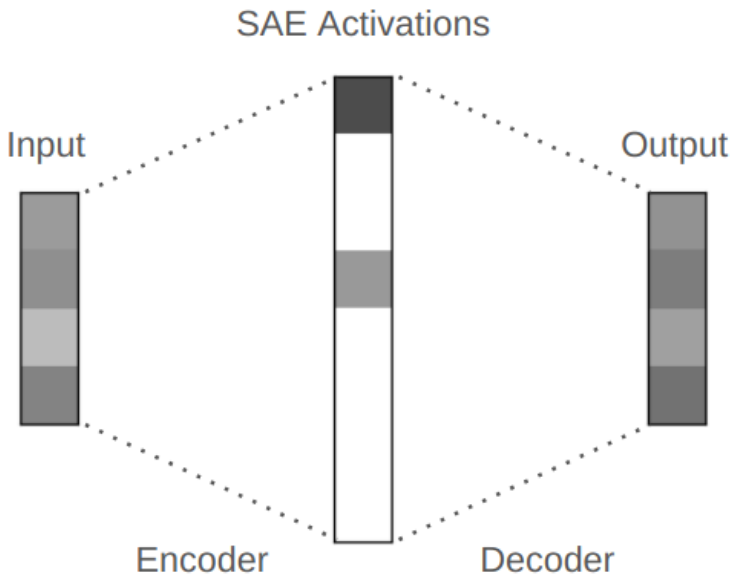
- 현대 Learned Sparse Retrieval (LSR) models는 너무 vocab에 의존적이다
  - vocab에 의해 vector가 표현되기 때문에, 차원 축소도 어렵고, multilingual, cross-lingual retrieval도 어렵다.
  - cased 버전의 경우, LSR로 학습하기 어렵게 된다.



SPLARE: Learning Retrieval Models with Sparse Autoencoders

Preliminaries

- **SAE란?**
  - LLM 내부 동작의 해석을 위해 LLM이 생성하는 representation을 **해석 가능한 latent features로 분해**하는 역할을 담당
  - SAE는 단순히 **encoder-decoder**로 구성된 single hidden layer 모델임
  - 각 토큰의 hidden state를 encode/decode하며, **Reconstruction Loss**로 학습됨
    - $L = |\hat{x} - x|^2$
    - 이때, activation을 sparse하게 만들기 위해 ReLU, Top-K 같은 함수 사용
  - 이렇게 학습된 SAE가 표현하여 activate되는 차원은 **고유한 semantic을 나타냄**



SPLARE: Learning Retrieval Models with Sparse Autoencoders

Motivation

- SAE는
  - **mono-semantic** (하나의 feature가 하나의 semantic을 나타냄)
  - **multilingual** (feature들이 language-agnostic함)
  - **multimodal** (feature들이 modality를 넘어선 의미 align을 보여줌)

  의 특징을 가지고 있다는 것이 사전 연구들을 통해 증명됨

  ➔ SAE는 LSR Models에 완벽히 fit하다!

SPLARE: Learning Retrieval Models with Sparse Autoencoders

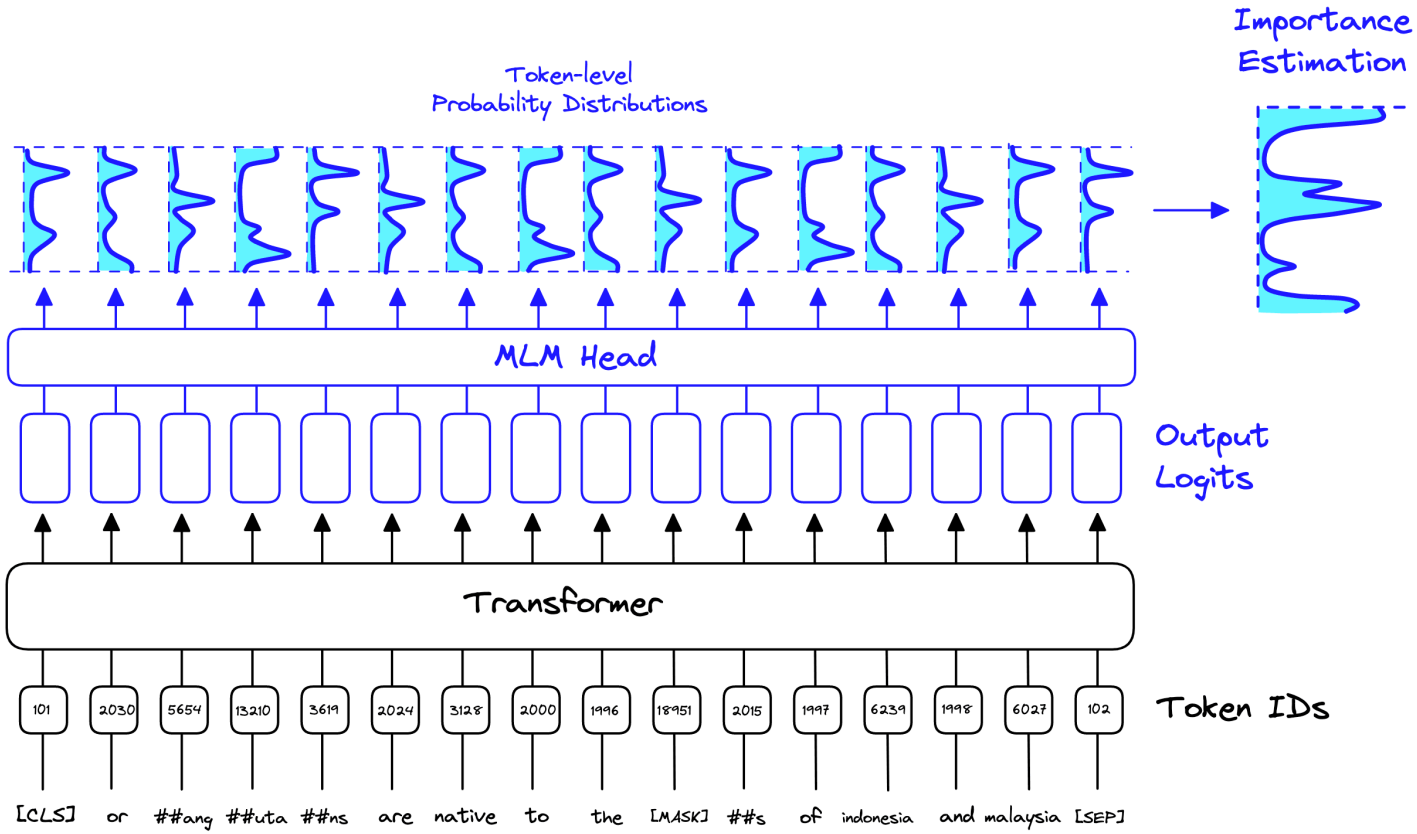
Contribution

1. **SPLARE**: **SP**arse **LA**tent **RE**trieval – pre-trained SAEs를 사용하여 LSR 모델을 만드는 새로운 방식 제안
2. Latent vocab을 사용하는 것의 유용함 분석
3. 7B, 2B 크기의 SOTA 성능을 보이는 multilingual sparse embedding model 개발

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Method

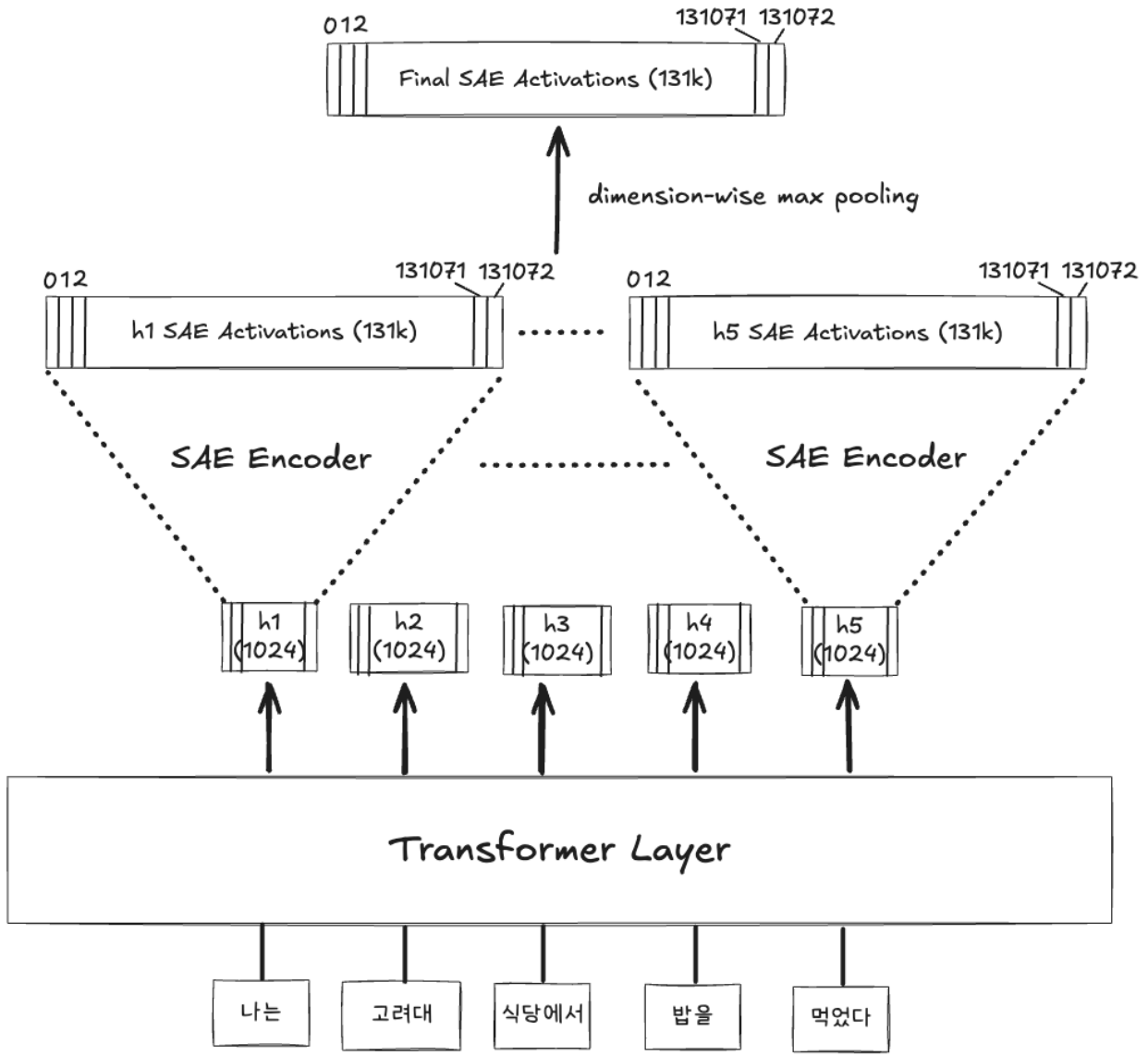
- **원래 SPLADE**



SPLARE: Learning Retrieval Models with Sparse Autoencoders

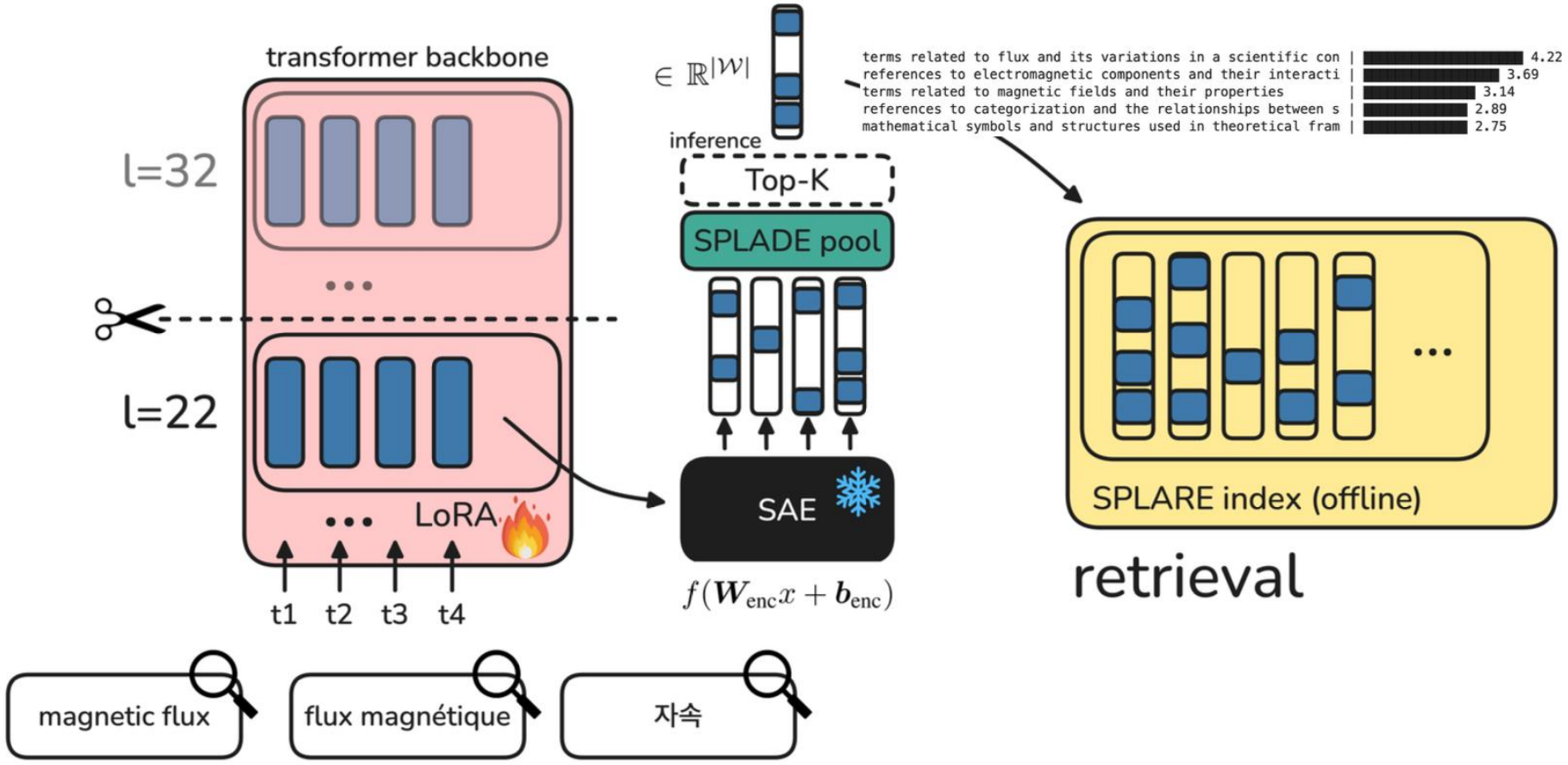
Method

- **SPLARE**



SPLARE: Learning Retrieval Models with Sparse Autoencoders

Method



SPLARE: Learning Retrieval Models with Sparse Autoencoders

Method

- **Modeling**
  - 특정 layer에 SAE 달아서 학습
  - SAE Encoder는 freeze한 채로 학습

> We used the KLDiv because it's one of the recipes that works well for distillation. It worked better than only contrastive in our case. We did not apply any normalization other than the softmax and tuning the temperature hyperparam within the KLDiv.

- **Training Method**
  - 원래 LSR Models를 학습하기 위해 Contrastive Learning을 주로 사용하지만, SPLARE 학습에는 **Cross Encoder Teacher scores로 KL Divergence 을 활용한 Distillation**을 사용
  - <query, positive, hard_negatives>가 주어진 경우, student model (SPLARE)가 매긴 query와 pos, query와 hn들의 유사도 score 분포가 teacher model (reranker)가 매긴 유사도 score 분포를 따라가도록 학습

$$L = L_{KL} + \lambda_q L^q_{FLOPS} + \lambda_d L^d_{FLOPS}$$

- **Inference Time**
  - 맨 마지막에 Top-K pooling 달아서 사용 (query=40, document=400)

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Experimental Setup

- **Backbone & SAE**
  - Backbone: Llama-3.1-8B, Gemma-2-2B
  - SAE: Llama Scope (사전학습된 Llama용 SAE) // Gemma Scope (사전학습된 Gemma용 SAE)

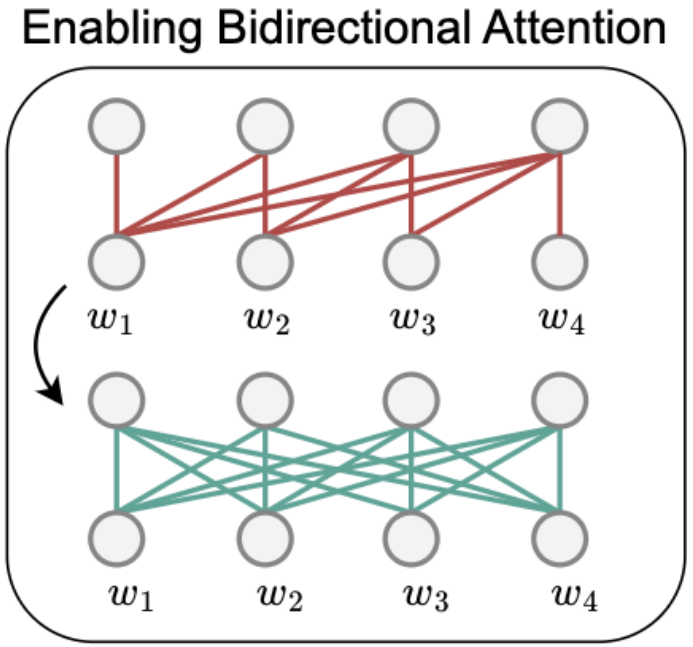
1. **Pretrain**
   - SPLADE류는 bidirectional attention이 굉장히 중요함. 따라서 msmarco 데이터로 **Masked Next Token Prediction (MNTP)** 수행

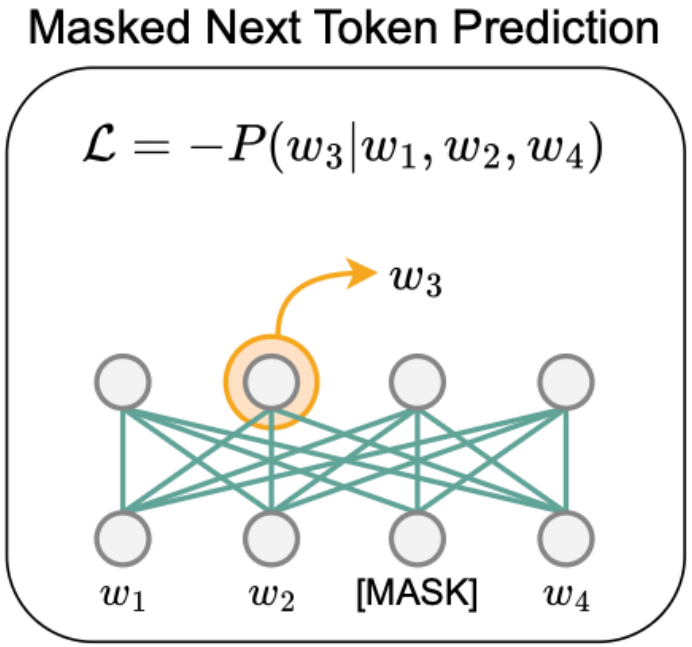
2. **Exp1 (변인 통제 실험)**
   1. English 언어에 대해, msmarco[train]으로 학습시켰을 때의 성능 비교
   2. **목적: Transformer의 어느 layer에서 성능이 가장 좋은지, SAE의 넓이가 성능에 어떤 영향을 미치는지 분석**

3. **Exp2**
   1. Exp1에서 정한 design choice 적용하고, 공개된 multilingual data 사용하여 SOTA multilingual sparse 모델 학습

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Experimental Setup





Experimental Setup

- **Exp1**
  - Dataset: msmarco, HN mine w/ SPLADE-v3
  - Teacher: DeBERTa-v3 reranker
- **Exp2**
  - Dataset: bge-multilingual-gemma2 학습 위해 사용된 multilingual dataset (1.3M)
    - Msmarco, nq, hotpotqa, zh reteival, miracle, mrtidy 등등..
  - Teacher: bge-reranker-v2-m3 reranker
- **Details**
  - KLDiv temperature: {1, 10, 20, 40, 50, 80, 100} grid search w/ NanoBEIR
  - LoRA training

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Experimental Setup

- **Details**

Component	Value
LoRA rank r	64
Max training sequence length (english models)	128
Max training sequence length (multilingual models)	512
Epochs	1
Batch size (w/ gradient accumulation)	128
Learning rate	5×10^{-5}
Warmup ratio	0.01
Nb negatives per query	8
λ_d	0.0001
λ_q	0.0001
τ SPLARE - Llama Scope	80
τ SPLARE - Gemma Scope	50
τ SPLADE-Llama	10
Evaluation max context size	1024
Adam βs	0.9, 0.999

Experimental Setup

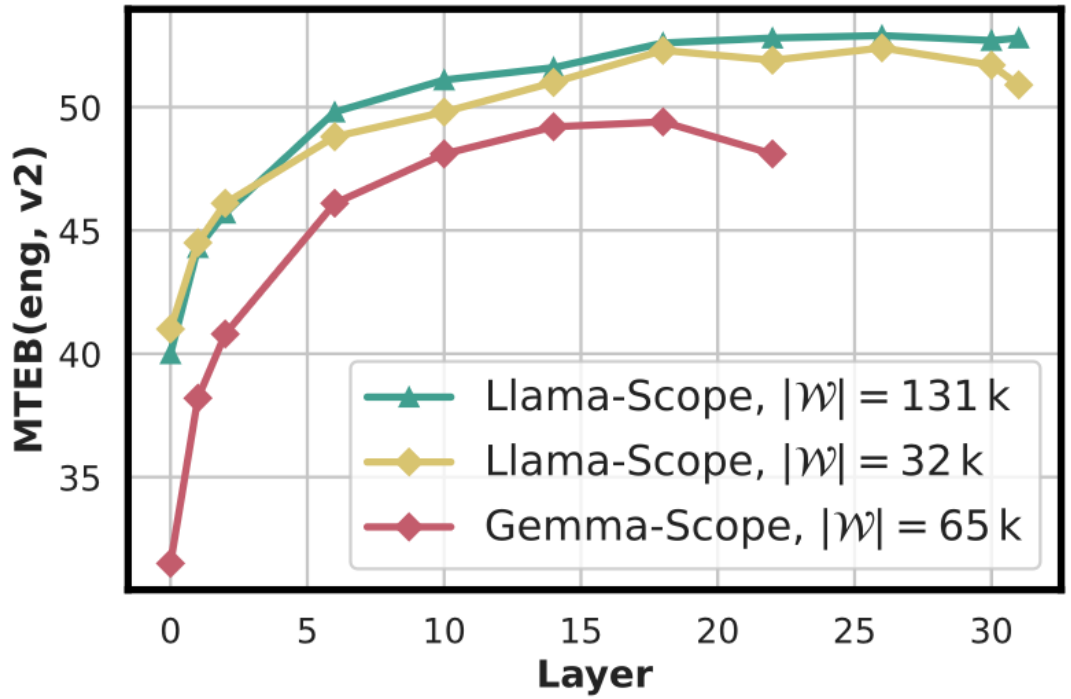
- **Evaluation**
  - MTEB, MMTEB retrieval 태스크 대상으로 평가
  - MIRACL (multilingual), XTREME-UP (cross-lingual: X ➔ EN)

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Results & Analysis – Exp1

Llama vocab으로 학습 (SPLADE-Llama) vs. SAE 달아서 학습 (SPLARE)

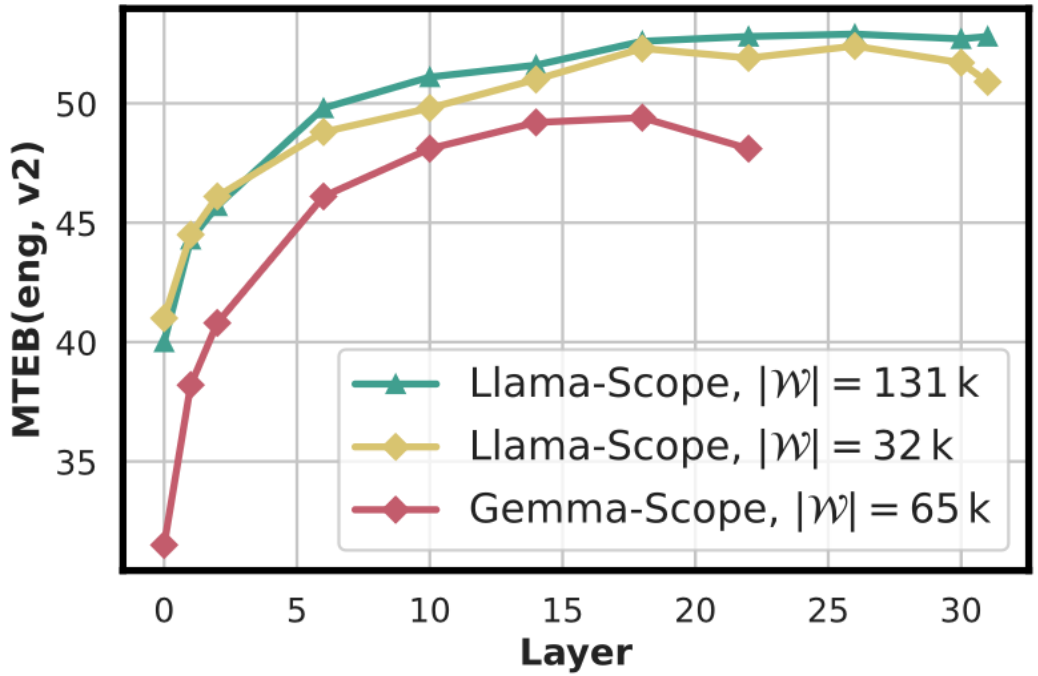
- RQ1. 어떤 transformer layer에서 가장 효과적인 sparse vector를 얻는가?

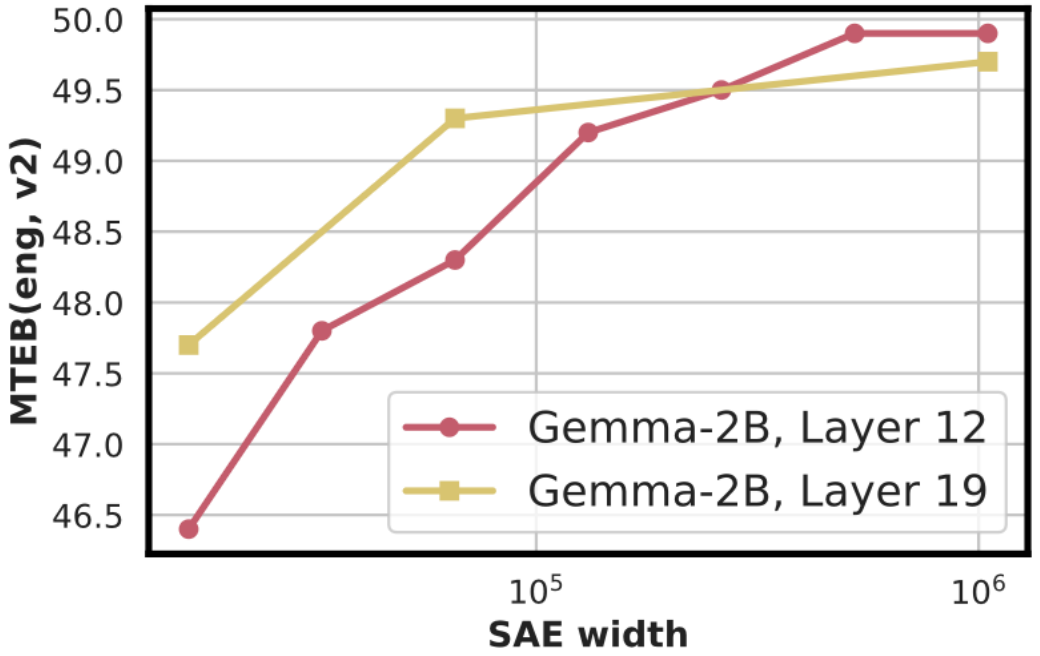


SPLARE: Learning Retrieval Models with Sparse Autoencoders

Results & Analysis – Exp1

- RQ2. SAE width (출력 차원 수)에 따라 성능이 얼마나 변하는가?

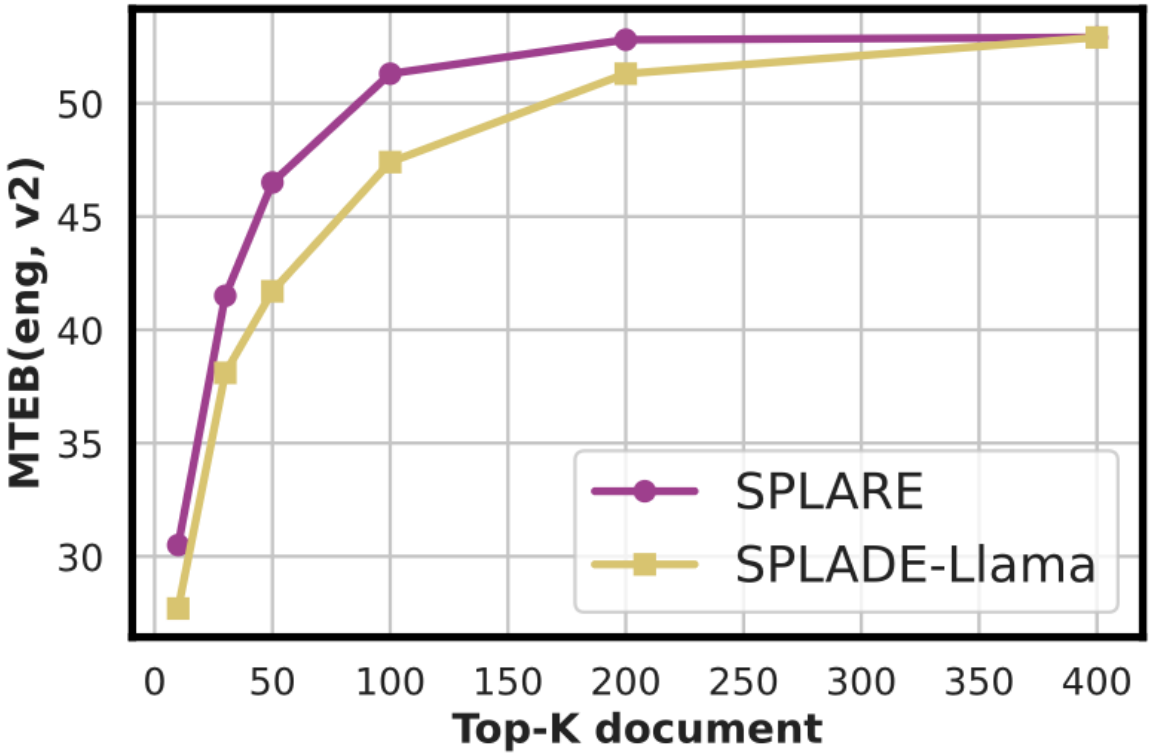


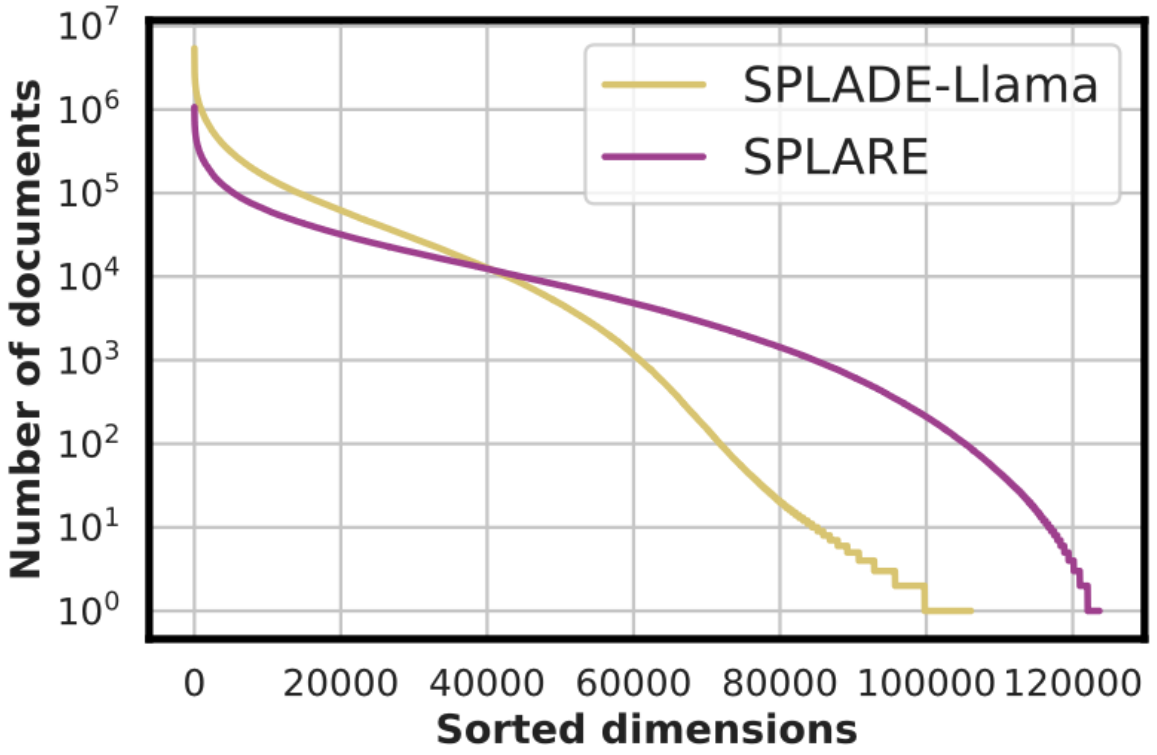


SPLARE: Learning Retrieval Models with Sparse Autoencoders

Results & Analysis – Exp1

- RQ3. Efficiency-Effectiveness trade-off 분석





SPLARE: Learning Retrieval Models with Sparse Autoencoders

Results & Analysis – Exp1

- RQ4. 실제 다른 모델들과 비교했을 때의 벤치마크 성능

	English	Multilingual	Code	Medical	Law	ChemTEB
English Models						
SPLADE-v3 (Lassance et al., 2024)	50.7	38.1	44.5	44.2	40.4	75.6
Lion-SP-8B (Zeng et al., 2025)	48.5	50.0	53.3	54.4	48.5	71.1
SPLADE-Llama	**52.9**	54.3	**57.3**	61.0	49.0	**75.9**
SPLARE	**52.9**	**56.3**	55.1	**62.9**	**51.2**	70.0

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Results & Analysis – Exp2

1. SPLARE vs. SPLADE-Llama – 다국어 데이터로 학습시켰을 때의 성능은?

	English	Multilingual	Code	Medical	Law	ChemTEB
Multilingual Models						
SPLADE-Llama	58.9	61.7	**64.3**	67.6	60.7	77.4
SPLARE	**59.3**	**62.3**	63.0	**67.7**	**60.8**	**78.1**

Table 2: Multilingual comparison of SPLARE and SPLADE (Top-K $= (40, 400)$).

	indic	sca	deu	fra	kor	XTREME-UP	MIRACL
SPLADE-Llama	91.9	70.4	**57.3**	**65.6**	74.8	56.2	69.9
SPLARE	**92.3**	**70.8**	57.1	64.8	**76.0**	**58.6**	**71.7**

SPLARE: Learning Retrieval Models with Sparse Autoencoders

Results & Analysis – Exp2

2. 고트들과의 비교

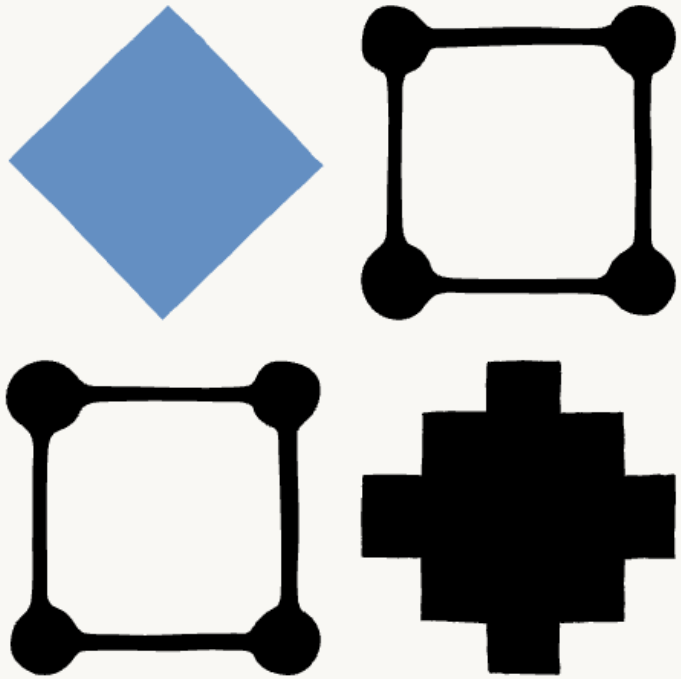
	English	Multilingual	XTREME-UP
Top Open Source models			
e5-mistral-7b-instruct (Wang et al., 2024a)	57.6	55.8	-
NV-Embed-v2 (Lee et al., 2025a)	62.8	56.7	-
multilingual-e5-large-instruct (Wang et al., 2024b)	53.5	57.1	18.7
GritLM-7B (Muennighoff et al., 2024)	55.0	58.3	-
SFR-Embedding-Mistral (Meng et al., 2024)	59.3	59.4	-
Linq-Embed-Mistral (Kim et al., 2024)	60.1	58.7	24.6
gte-Qwen2-7B-instruct (Li et al., 2023b)	58.1	60.1	17.4
voyage-3-large (AI, 2025)	53.5	66.1	39.2
jina-embeddings-v4 (Günther et al., 2025)	56.2	66.4	-
inf-retriever-v1 (Yang et al., 2025)	64.1	66.5	-
Qwen-3-Embedding-8B (Zhang et al., 2025)	69.4	70.9	-
Commercial models			
Cohere-embed-multilingual-v3.0 (Cohere, 2023)	55.7	59.2	-
text-embedding-3-large (OpenAI, 2024)	58.0	59.3	18.8
gemini-embedding-001 (Lee et al., 2025b)	64.4	67.7	64.3
SPLARE	59.3	62.3	58.6
SPLARE - no-pooling	61.4	63.8	61.4
SPLARE - Top-K $= (20, 200)$	55.9	59.9	53.8
SPLARE - Top-K $= (10, 100)$	50.1	56.0	46.5
SPLARE-2B	55.9	59.1	41.6

SPLARE: Learning Retrieval Models with Sparse Autoencoders

느낀점

1. SPLADE가 Vocab에 의존하는게 항상 문제라고 생각했는데, 그걸 해결하는 신박한 방법이라 재미있게 읽었음.
2. 논문을 작성하는 방식에서도 배울 것이 많다고 생각
   (e.g. design choice를 결정하는데에 있어서는 controlled experiment를 가져가고, 그렇게 결정된 design으로 추가 데이터들 학습)
3. Failure case를 다룬 논문을 처음 보는 것 같음. 얼마나 이 사람들이 " 왜 특정 벤치마크에서 실패했는가"를 고민했는지 잘 보여줌. (정성분석 appen에 있음)
4. 아쉬운점: SAE를 학습시키는 것 자체가 굉장히 큰 하나의 topic일 것 같은데, 이것도 좀 해보면 어땠을까 하는 생각? 근데 이건 아예 하나의 논문으로 나오는게 나을 수도 있다고도 생각함

Engineering at Anthropic



Harness design for long-running application development

Harness Engineering

요즘의 나

Harness Engineering

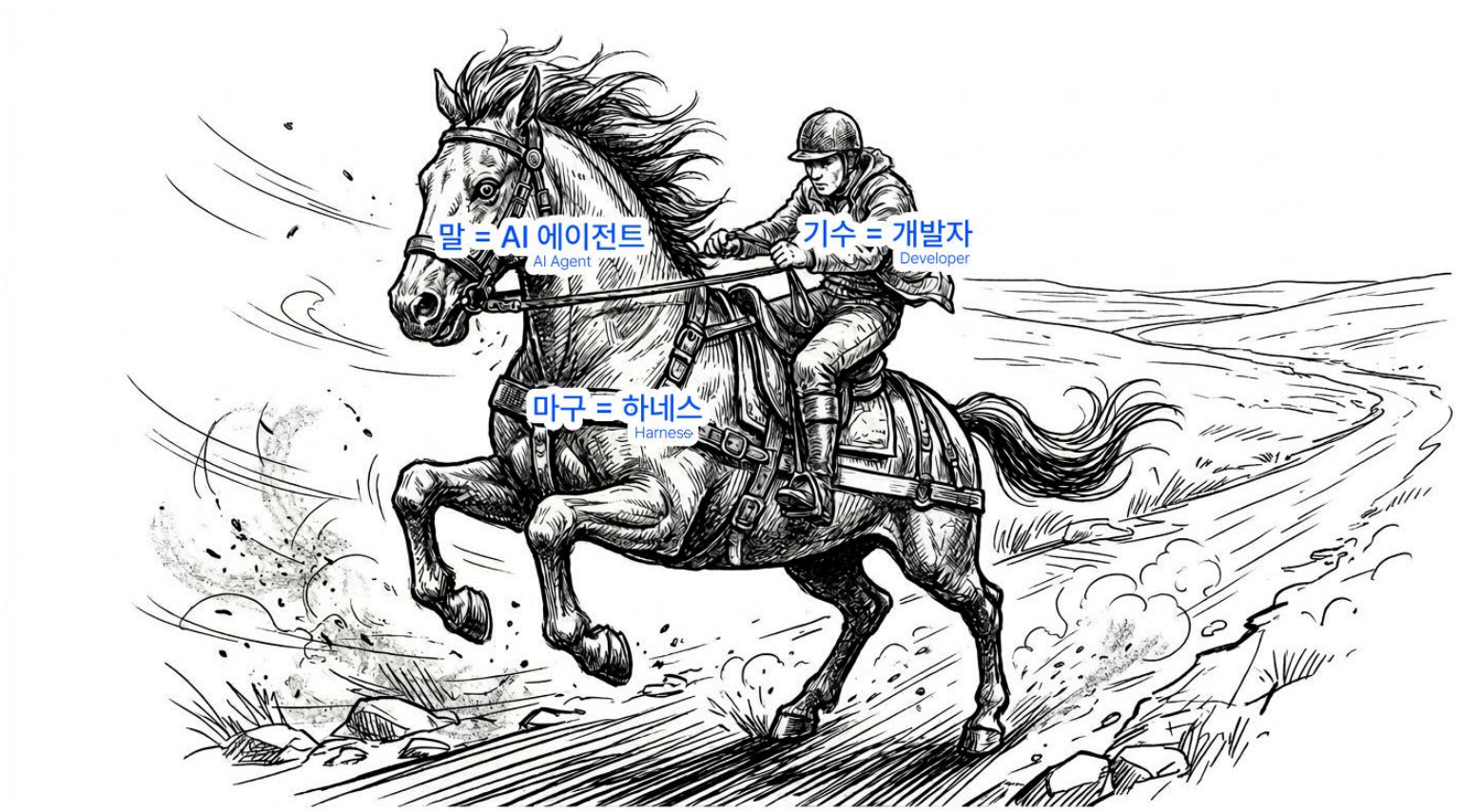
요즘의 개발자들

Harness Engineering

요즘 나의 고민

Harness Engineering

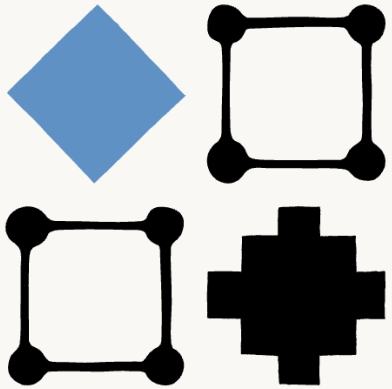
Harness Engineering 이란?



Harness Engineering

Harness design for long-running application development

Engineering at Anthropic



Harness design for long-running application development

Published Mar 24, 2026

Harness design is key to performance at the frontier of agentic coding. Here's how we pushed Claude further in frontend design and long-running autonomous software engineering.

Harness Engineering

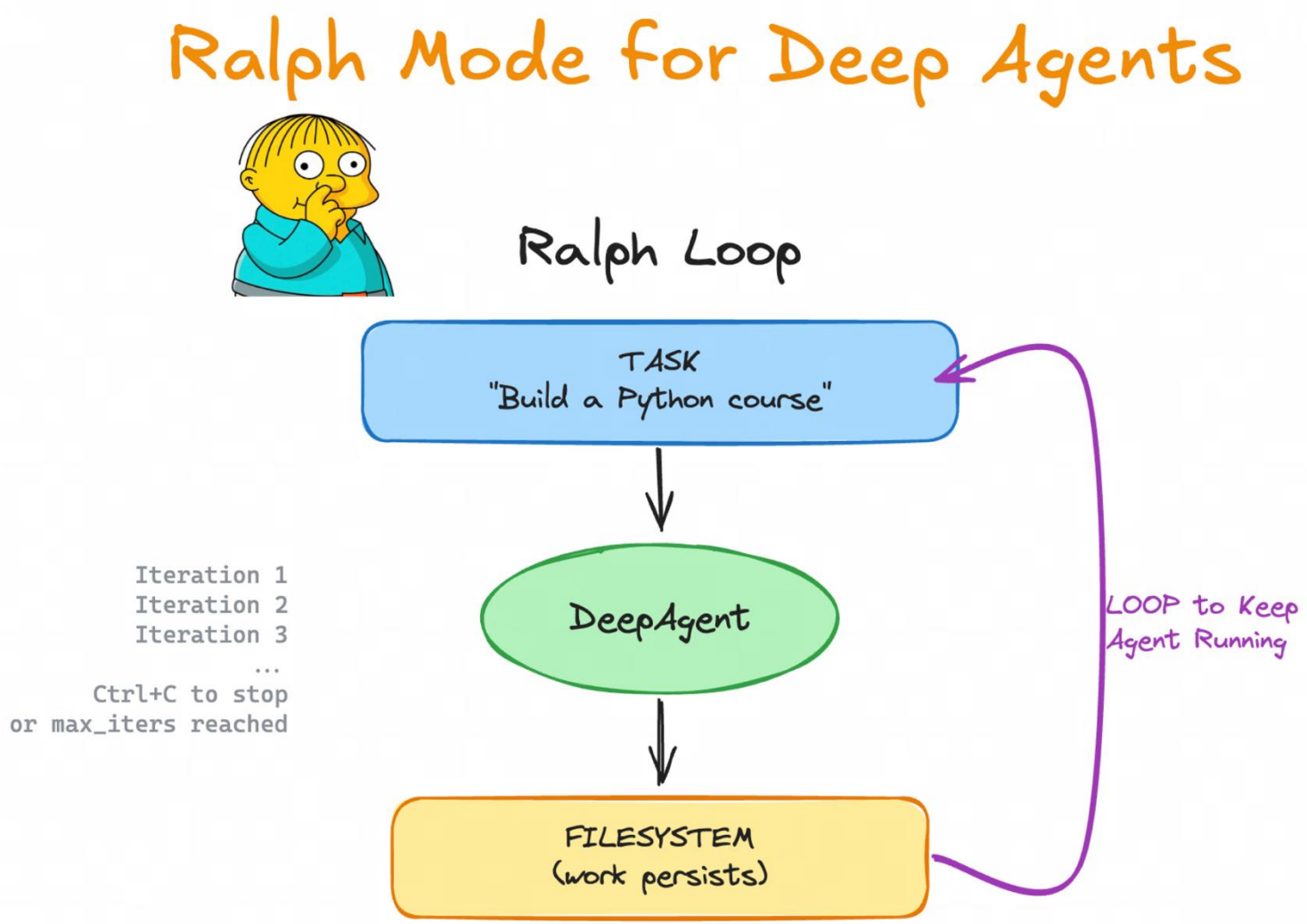
1. Ralph mode

Ralph

Ralph is an autonomous AI agent loop that runs AI coding tools (Amp or Claude Code) repeatedly until all PRD items are complete. Each iteration is a fresh instance with clean context. Memory persists via git history, `progress.txt`, and `prd.json`.

Harness Engineering

1. Ralph mode



2. Anthropic의 문제 지적

1. Context Anxiety

- Context window가 차오를수록, 모델이 작업을 성급하게 마무리하려고 함

2. Self-Evaluation Bias

- 모델이 자신의 결과물을 평가할 때 지나치게 관대해지는 현상
- 특히 이는 객관적 검증이 어려운 디자인이나 UX 영역에서 두드러짐

3. 해결책

GAN에서 영감받은 구조

1. Generator와 Evaluator를 완벽하게 분리하자 !!

● **Update**(docs/superpowers/specs/2026-03-31-adaptive-rubric-generation-design.md)
  └ Added 2 lines, removed 2 lines

```
123
124  4. Statistical Dimension Selection (코드, LLM 아님):
125     a. Discriminative power: d_j = |pass_rate(f_j=1) - pass_rate(f_j=0)|
126 -   b. Redundancy 제거: corr(f_j, f_k) > 0.8이면 d 낮은 쪽 제거
127 -   c. Top-K features 선택 (d_j 기준)
126 +   b. Top-K features 선택 (d_j 기준 내림차순)
127 +   (Note: 의미적 중복 제거는 Step 2c에서 이미 처리됨)
128
129  5. Weight Estimation (코드, LLM 아님):
130     w_j = d_j / Σ d_j  (normalized discriminative power)
```

● 간결해졌습니다. 다른 수정사항 있으신가요? 없으면 spec 확정하겠습니다.

❯ 또 불필요한 거 있나? 혹시 너가 지금 우리가 얘기하고 있는 세션과는 독립적인 evaluator agent (Opus 4.6)을 소환해서 계획서 판단하는 agent라는 system prompt를 주고 판단해보게 해줄 수 있어 ?

● 좋은 아이디어입니다. 독립적인 Opus evaluator를 소환해서 spec을 리뷰하게 하겠습니다.

● **oh-my-claudecode:architect**(Independent spec review by evaluator agent) Opus 4.6
  └ Done (5 tool uses · 34.6k tokens · 1m 50s)
  (ctrl+o to expand)

● **Evaluator Agent 리뷰 결과**

Overall: "Strong core idea, needs surgical simplification."

핵심 강점은 인정 (C0-C6 controlled comparison, attention failure framing, same-model design). 하지만 8가지 weakness를 지적했습니다. 가장 중요한 것들을 정리하면:

즉시 반영해야 할 것 (동의함)

#	지적	대응
W2	**Dynamic ARG가 under-specified하고 복잡성 대비 가치 불명확.** τ, L, decay_factor 미정, cross-task attribution이 통계적으로 의심스러움	**Static ARG를 main으로 하고, Dynamic은 preliminary exploration으로 축소** — 동의합니다. 이러면 구현 복잡도 30% 감소
W5	**SWE-bench Lite는 standard refinement loop와 안 맞음.** Repo-level task에는 agentic scaffolding이 필요한데, 그러면 controlled comparison이 깨짐	**SWE-bench 제거, 다른 benchmark로 교체** (APPS, CodeContests, 또는 LiveCodeBench)
W3	**C4 (Generic rubric) 정의가 모호.** 사람이 잘 만들면 C5를 이기고, 못 만들면 인위적으로 약해 보임	**C4를 "LLM이 generic prompt로 생성한 rubric (CFA 없이)"으로 정의** — C4 vs C5가 곧 CFA의 효과를 측정하는 clean test
W6	**CFA sample model mismatch** 미해결	**Main experiment에서 CFA도 Opus로 실행. Haiku CFA는 ablation으로**
W7	**3 runs는 통계적으로 부족.** C3/C4/C5 차이가 작을 수 있음	**5 runs로 증가**

Harness Engineering

3. 해결책

GAN에서 영감받은 구조

1. Generator와 Evaluator를 완벽하게 분리하자 !!
2. 기준을 **정말 구체적으로** 세우자

- **Design quality:** Does the design feel like a coherent whole rather than a collection of parts? Strong work here means the colors, typography, layout, imagery, and other details combine to create a distinct mood and identity.
- **Originality:** Is there evidence of custom decisions, or is this template layouts, library defaults, and AI-generated patterns? A human designer should recognize deliberate creative choices. Unmodified stock components—or telltale signs of AI generation like purple gradients over white cards—fail here.
- **Craft:** Technical execution: typography hierarchy, spacing consistency, color harmony, contrast ratios. This is a competence check rather than a creativity check. Most reasonable implementations do fine here by default; failing means broken fundamentals.
- **Functionality:** Usability independent of aesthetics. Can users understand what the interface does, find primary actions, and complete tasks without guessing?

4. Case Study1

Frontend 자동화

- Playwright MCP를 사용해서 Evaluator Agent가 직접 화면의 버튼 눌러보고, 스크롤 해보고 여러 인터렉션을 해볼 수 있는 환경 제공

Harness Engineering

4. Case Study1

https://cdn.sanity.io/files/4zrzovbb/website/9877febd34432f7f582aecd0023b951223605c6a.mp4

4. Case Study2

Scale-Up: 풀스택 자율 개발을 위한 3-Agent system

1. Planner: 짧은 프롬프트를, 상세한 제품 스펙으로 확장
2. Generator: 스펙을 바탕으로 하나의 기능씩 구현 (Sprint 형태)
3. Evaluator: Playwright로 실제로 사용해보면서 버그 탐색 및 평가 결과 충족 여부 결정

Harness	Duration	Cost
Solo	20 min	$9
Full harness	6 hr	$200

4. Case Study2

https://www.anthropic.com/engineering/harness-design-long-running-apps

4. Ablation (Opus 4.6)

- 모델 성능이 좋아지면서, Sprint 구조 (태스크 하나씩 해결해나가는 방법)을 제거해도 긴 문맥을 유지할 수 있고, 더 짧은 시간 안에 개발이 가능했음
- Insight: Evaluator의 필요성은 고정된 것이 아니라, **현재 모델의 한계점이 어디인지에 따라 동적으로 변한다**.

4. Conclusion & Takeaways

- 모델 성능이 좋아진다고 해서, Harness Engineering의 가치가 줄어드는 것이 아니다.
- 오히려 해당 모델에 따른 Harness의 현태를 새로 구축해야 하며, 최적의 Agent Harness 조합을 찾는 것이 정말 강력한 무기이다.

나의 생각

- Agent Debate 구조에서 벗어나 현재까지 이 Ralph 구조가 유지되고 있는 상태에서, 가장 중요한 것은 **Evaluation**이 아닐까 ??
- 앞선 실험들에서도 강조되었지만, Evaluation Agent가 어떤 평가 기준을 가지고 있는지에 따라 성능 상한이 결정됨
- 결국은 Open Question을 만났을 때,
  - "**우리의 기준을 Evaluator Agent에 어떻게 잘 전달 할 수 있을지**" or
  - "**Evaluator Agent가 어떻게 알잘딱깔센하게 인간의 기준을 반영한 평가 기준을 생성해 낼 수 있을지**" 연구해보는 것이 가치 있지 않을까? 하는 생각

Harness Engineering

마무리하며...

Meta-Harness: End-to-End Optimization of Model Harnesses

Yoonho Lee Stanford
Roshen Nair Stanford
Qizheng Zhang Stanford
Kangwook Lee KRAFTON
Omar Khattab MIT
Chelsea Finn Stanford

Project page w/ interactive demo: https://yoonholee.com/meta-harness/
Optimized harness: https://github.com/stanford-iris-lab/meta-harness-tbench2-artifact

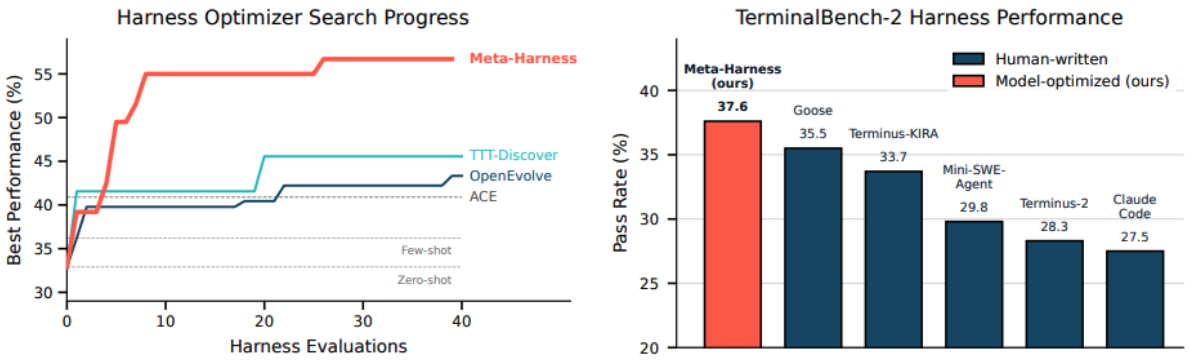
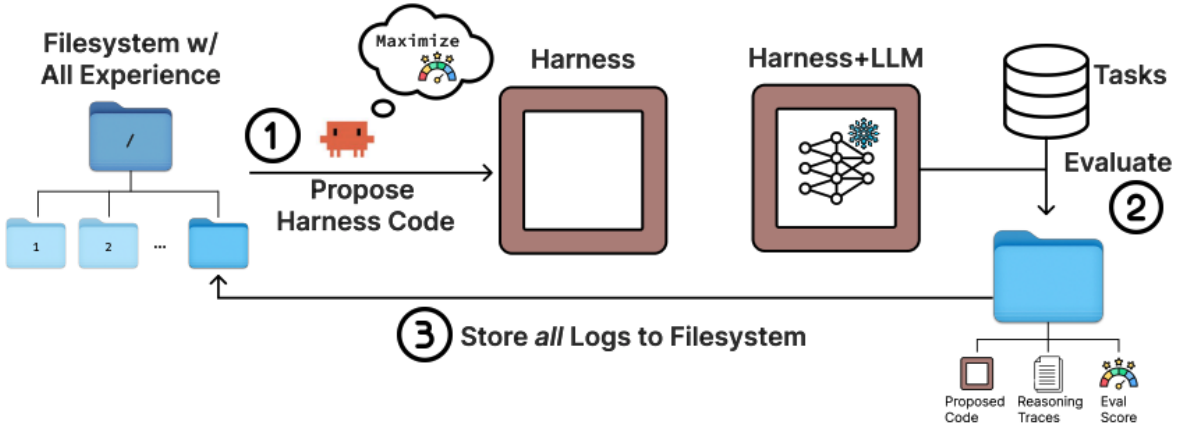


Figure 1: **(Left)** On text classification, Meta-Harness outperforms the best prior hand-designed harnesses (ACE) and existing text optimizers (TTT-Discover, OpenEvolve), matching the next-best method's final accuracy after just 4 evaluations. **(Right)** On TerminalBench-2, Meta-Harness outperforms all reported `Claude Haiku 4.5` harnesses.



Thank you

Q&A