SPLADE / Inference-Free SPLADE

2025 동계세미나

장영준

Table Of Contents

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

SPLADE

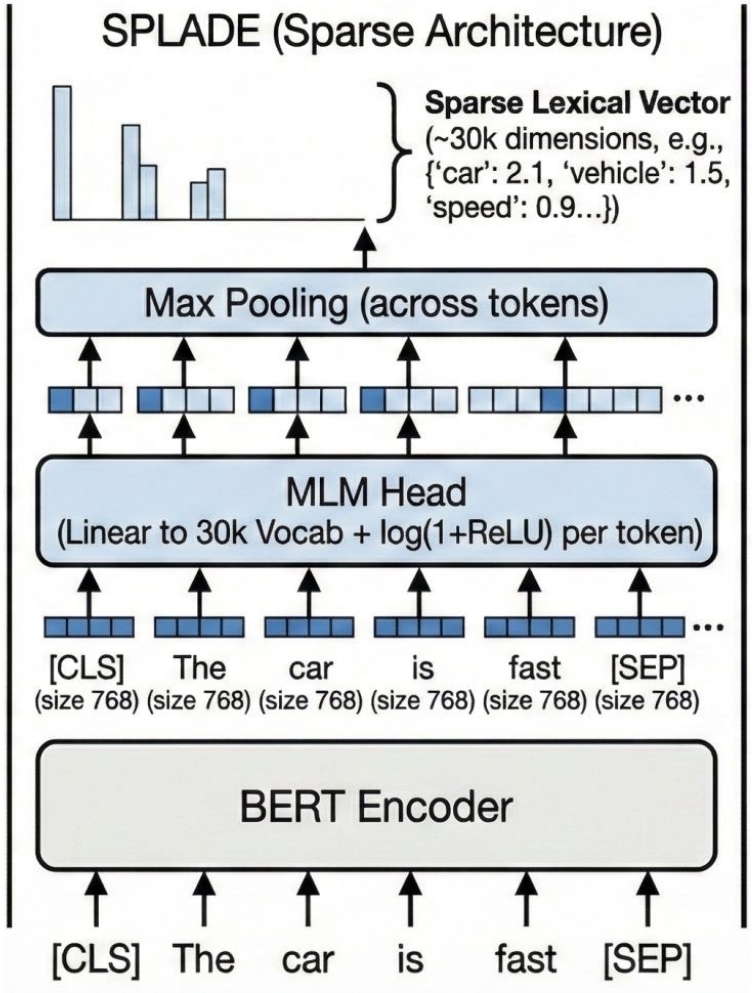
Problem?

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.

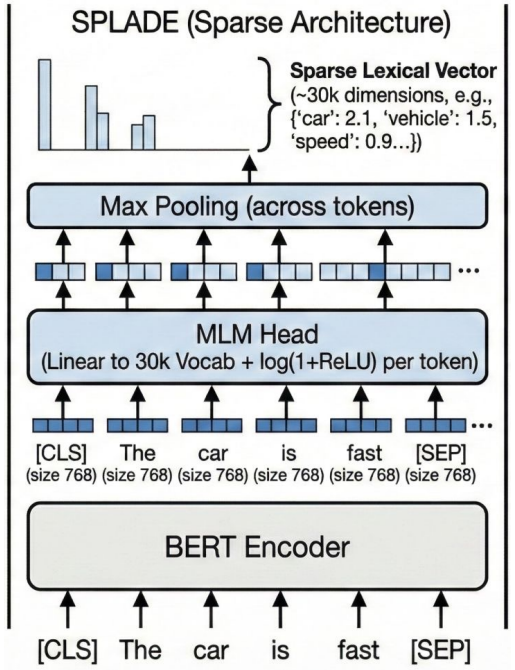
SPLADE

Method



SPLADE

Method



- 백본 모델: MLM model (BERT)
- SPLADE는 전적으로 백본의 MLM 능력에 의존함
- 구체적으로는, BERT에서 출력되는 토큰들의 hidden state를 MLM head에 통과시킴

$$w_{ij} = \text{transform}(h_i)^T E_j + b_j \quad j \in \{1, ..., |V|\}$$

- Wij = "문맥에서 i token이 j token에게 주는 **importance score**"
  - E.g. "The bank is near the river"

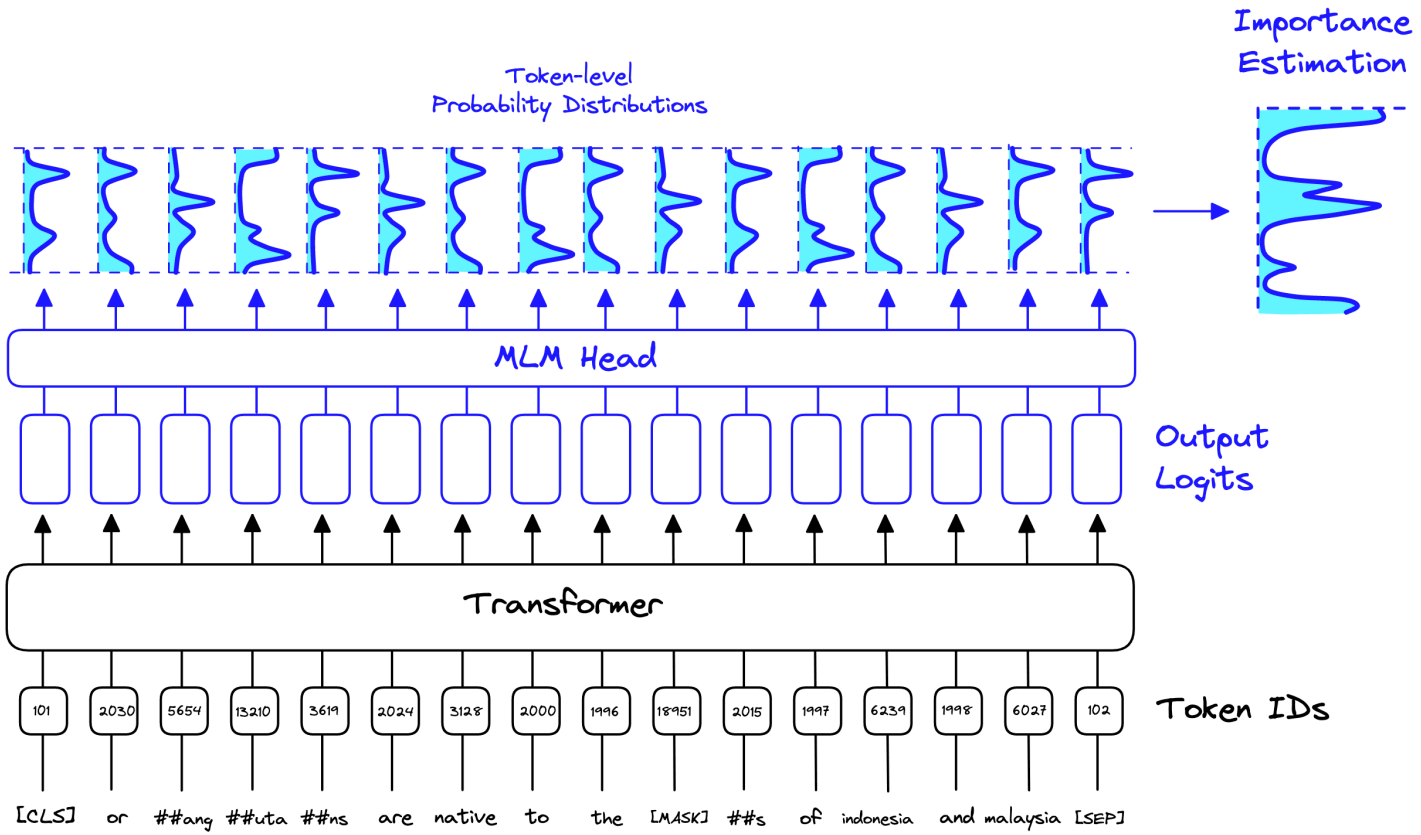
    ➔ "bank" 토큰은 "water", "flow" 같은 token들에 높은 score, "money", "deposit"와 같은 token들에 대해서는 상대적으로 낮은 score 매김
- 이후 이 importance score에 ReLU를 적용하여 음수 값 제거 후, Max Pooling

$$w_j = \max_{i \in t} \log\left(1 + \text{ReLU}(w_{ij})\right)$$

SPLADE

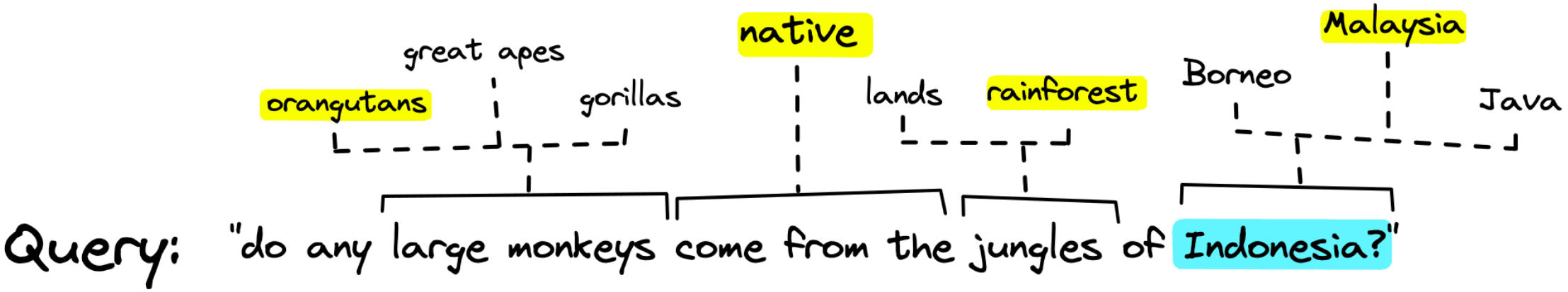
Method

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



SPLADE

검색 상황 예시



with query expansion

without query expansion

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

(원래는 doc도 확장됨)

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

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

SPLADE

Method

- **최종 Loss**

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

Training details & Experimental Setup

- Dataset: MSMARCO[train] for train // MSMARCO[dev], TREC-2019[eval] for eval
- Training Details
  - Backbone: BERT-base
  - Batch_size = 124, 150k step동안 학습

SPLADE

Result

model	MS MARCO dev		TREC DL 2019		FLOPS
	MRR@10	R@1000	NDCG@10	R@1000	
Dense retrieval					
Siamese (ours)	0.312	0.941	0.637	0.711	-
ANCE [25]	0.330	0.959	0.648	-	-
TCT-ColBERT [15]	0.335	0.964	0.670	0.720	-
Sparse retrieval					
BM25	0.184	0.853	0.506	0.745	0.13
DeepCT [4]	0.243	0.913	0.551	0.756	-
doc2query-T5 [18]	0.277	0.947	0.642	0.827	0.81
ST lexical-only [1]	0.275	0.912	-	-	-
ST expansion [1]	0.279	0.925	-	-	-
Our methods					
ST lexical-only	0.290	0.923	0.595	0.774	1.84
ST exp-ℓ_1	0.314	0.959	0.668	0.800	4.62
ST exp-ℓ_{FLOPS}	0.312	0.954	0.671	0.813	2.83
SPLADE-ℓ_1	0.322	0.954	0.667	0.792	0.88
SPLADE-ℓ_{FLOPS}	0.322	0.955	0.665	0.813	0.73

SPLADE-v2

Result

- SPLADE-v2에서는 max pooling 추가

model	MS MARCO dev		TREC DL 2019	
	MRR@10	R@1000	NDCG@10	R@1000
Dense retrieval				
Siamese (ours)	0.312	0.941	0.637	0.711
ANCE [29]	0.330	0.959	0.648	-
TCT-ColBERT [16]	0.359	0.970	0.719	0.760
TAS-B [11]	0.347	0.978	0.717	0.843
RocketQA [24]	0.370	0.979	-	-
Sparse retrieval				
BM25	0.184	0.853	0.506	0.745
DeepCT [4]	0.243	0.913	0.551	0.756
doc2query-T5 [20]	0.277	0.947	0.642	0.827
SparTerm [1]	0.279	0.925	-	-
COIL-tok [9]	0.341	0.949	0.660	-
DeepImpact [18]	0.326	0.948	0.695	-
SPLADE [8]	0.322	0.955	0.665	0.813
Our methods				
SPLADE-max	0.340	0.965	0.684	0.851
SPLADE-doc	0.322	0.946	0.667	0.747
DistilSPLADE-max	0.368	0.979	0.729	0.865

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)
]
```

Towards Competitive Search Relevance For Inference-Free Learned Sparse Retrievers

Zhichao Geng
zhichaog@amazon.com
Amazon
Shanghai, China

Yiwen Wang
wangyiwe@amazon.com
Amazon
Shanghai, China

Inference-Free SPLADE

Problem?

- 굳이 query에 자원을 소모해야 하나?
- **어차피 Vocab Mismatch 해결이 목적이라면, Document만 확장해서 색인해두면 되는 거 아닌가 ?? Query는 굳이 무거운 모델에 통과시킬 필요 없이 들어온 단어 그대로만 써도 충분할 것 같은데...**

Inference-Free SPLADE

Problem?

- SPLADE-v2 paper에서는 이러한 생각을 바탕으로 SPLADE-doc 제안

  ➔ **Document만 SPLADE 모델로 확장하고, query는 별도의 인코더 통과 없이 단어의 유무만 따지는 방식**

$$s(q,d) = \sum_{j \in q} w_j^d$$

- **Query**: 단순히 토크나이징된 단어들에 해당하는 위치만 1인 벡터 사용
- **Document**: SPLADE 모델을 통과하여 확장된 Sparse Vector 사용

Inference-Free SPLADE

Problem?

model	MS MARCO dev		TREC DL 2019	
	MRR@10	R@1000	NDCG@10	R@1000
Our methods				
SPLADE-max	0.340	0.965	0.684	0.851
SPLADE-doc	0.322	0.946	0.667	0.747

- Inference-Free SPLADE의 저자들: 성능 차이가 크지는 않지만...그래도 꽤 유효하다 ?!

  ➔ 이러한 성능 갭을 해결해보자!!

- Problem Statement:

  기존의 FLOPS Regularizer가 모든 토큰을 평등하게 penalize하는게 문제다

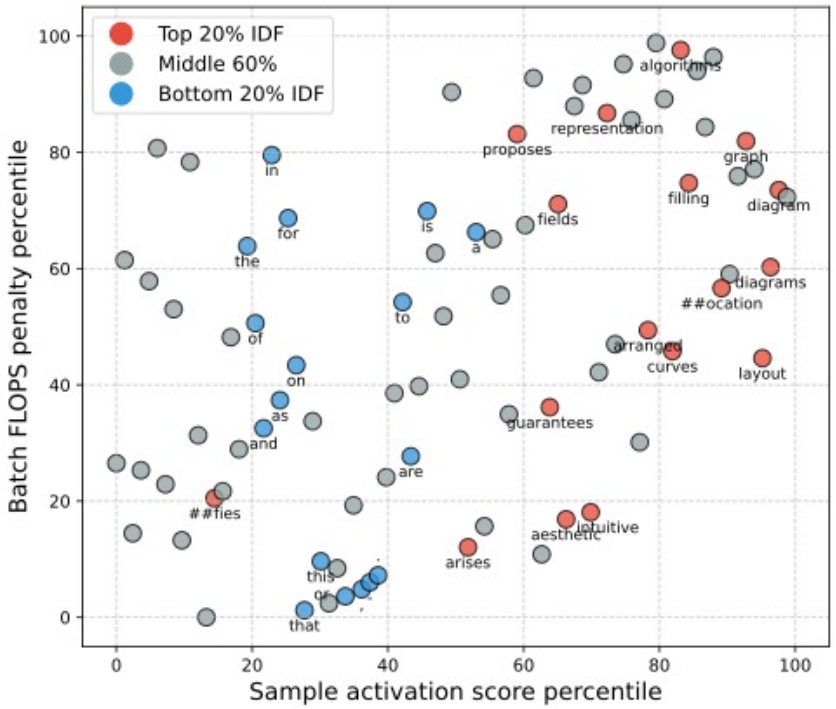
(그림 참고)

- “this”, “that” 같은 무의미한 토큰은 자주 등장함에도 페널티를 적게 받고,
- “algorithms”, “graph” 같은 중요한 토큰은 드물게 등장한다는 이유로

  오히려 더 강하게 penalize 당하고 있었다.

➔ 현재의 FLOPS penalty 체계가 semantic을 반영하고 있지 않다고 말하며,

IDF (Inverse Document Frequency)를 반영한 penalty를 제안



Inference-Free SPLADE

Method

1. 먼저 vocab에 있는 모든 토큰들로 train data에 대한 IDF 값을 계산함.
   이 계산으로 vocab size |V| 크기의, idf 값들을 원소로 가지는 벡터 idf(t)를 얻음.

2. Idf(t)를 이용하여 유사도 계산 함수를 다음과 같이 변경

$$s(q, d_i) = \sum_{t \in \mathcal{V}} \text{idf}(t) \cdot q_t \cdot d_{i,t},$$

3. 최종 Loss

$$\mathcal{L} = \mathcal{L}_{\text{rank-idf}} + \lambda \cdot \mathcal{L}_{\text{FLOPS}}.$$

4. 미분 시

$$\frac{\partial \mathcal{L}}{\partial d_{i,t}} = \frac{\partial \mathcal{L}_{\text{rank-idf}}}{\partial d_{i,t}} + \lambda \cdot \frac{\partial \mathcal{L}_{\text{FLOPS}}}{\partial d_{i,t}}. \qquad \frac{\partial \mathcal{L}_{\text{rank-idf}}}{\partial d_{i,t}} \propto \text{idf}(t) \cdot q_t \cdot (\text{softmax}_{\text{stu}} - \text{softmax}_{\text{tea}}).$$

➔ Idf가 크면 ranking loss로부터 받는 gradient가 커지고, flops loss로부터 받는 gradient가 작아지므로 토큰이 보존될 수 있음

Inference-Free SPLADE

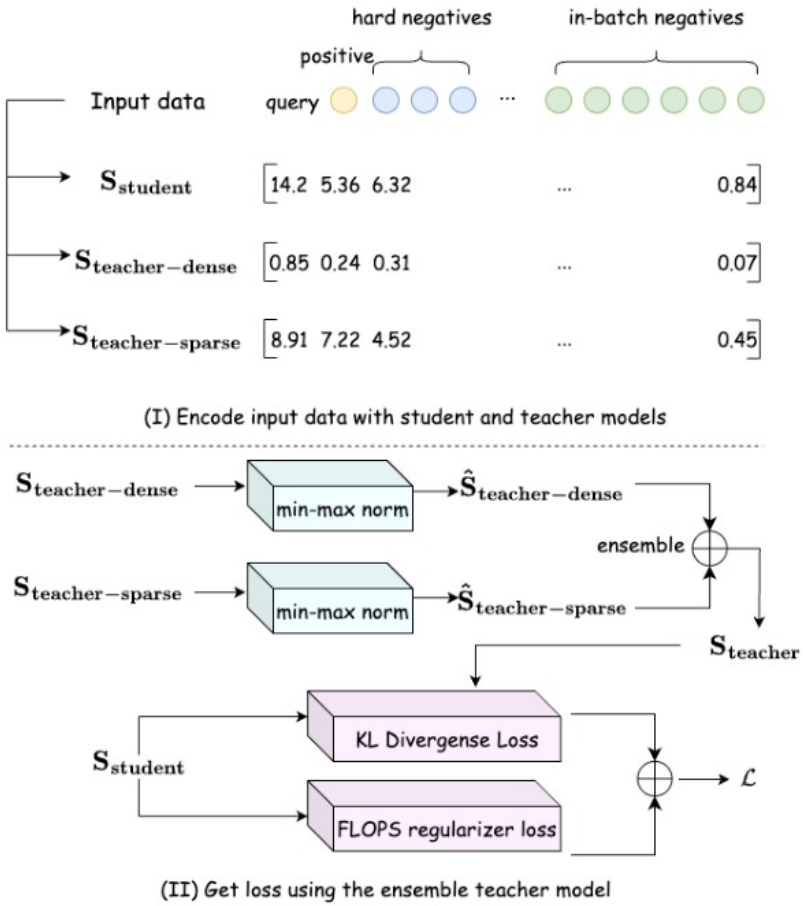
Train Method

[Loss]

- Dense와 Sparse 모델의 ensemble score를 **KLDiv Distillation**
- Score ensemble 시, min-max scaling 수행

[DATA]

1. Weakly-Supervised FT (Pre-Finetuning) <5.3M>
2. SFT <503K>



Training details & Experimental Setup

- Dataset: MSMARCO[train] for SFT
- Training Details
  - Backbone: co-condense—marco
  - Teacher [dense]: gte-large-en-v.15
  - Teacher [sparse]: opensearch-neural-sparse-encoding-v1

Inference-Free SPLADE

Result (In-domain)

Model	MS MARCO dev		TREC DL 2019	
	M@10	R@1000	NDCG	R@1000
Dense Retrievers				
ANCE	33.0	95.9	64.8	-
TCT-ColBERT	35.9	97.0	71.9	76.0
ColBERTv2	39.7	98.4	-	-
RocketQA	37.0	97.9	-	-
RocketQAv2	38.8	98.1	-	-
CoCondenser	38.2	98.4	-	-
TAS-B	34.7	97.8	71.7	84.3
Sparse Retrievers				
SparTerm	27.9	92.5	-	-
DistilSPLADE-max	36.8	97.9	72.9	86.5
SPLADE-v3-DistilBERT	38.7	-	75.2	-
Inference-free Sparse Retrievers				
BM25	18.4	85.3	50.6	74.5
DeepCT	24.3	91.3	55.1	75.6
doc2query-T5	27.7	94.7	64.2	**82.7**
SPLADE-doc	32.2	94.6	66.7	94.7
SPLADE-doc-distill†	36.5	96.9	69.8	74.2
SPLADE-v3-Doc	37.8	-	71.5	-
Our Model†	**37.8**	**97.5**	**72.1**	79.8

Inference-Free SPLADE

Result (Out-domain)

	Inference-free Sparse Retriever				Sparse Retriever		Dense Retriever		
Dataset	Our Model†	BM25	SPLADE-doc-distill†	SPLADE-v3-Doc	SPLADE++-SelfDistil	SPLADE-v3-Distil	ColBERTv2	Contriever	TAS-B
TREC-COVID	72.4	68.8	68.4	68.1	71.0	70.0	**73.8**	59.6	48.1
NFCorpus	**34.9**	32.7	34.0	33.8	33.4	34.8	33.8	32.8	31.9
NQ	53.1	32.6	48.8	52.1	52.1	54.9	**56.2**	49.8	46.3
HotpotQA	67.9	60.2	62.6	66.9	**68.4**	67.8	66.7	63.8	58.4
FiQA-2018	**36.4**	25.4	31.2	33.6	33.6	33.9	35.6	32.9	30.0
ArguAna	**49.1**	47.2	37.7	46.7	47.9	48.4	46.3	44.6	42.9
Touche-2020	28.7	34.7	25.6	27.0	**36.4**	30.1	26.3	23.0	16.2
DBPedia-entity	40.5	28.7	35.9	36.1	43.5	42.6	**44.6**	41.3	38.4
SCIDOCS	**16.7**	16.5	14.7	15.2	15.8	14.8	15.4	16.5	14.9
FEVER	78.5	64.9	67.4	68.9	78.6	**79.6**	78.5	75.8	70.0
Climate-FEVER	19.2	18.6	15.1	15.9	23.5	22.8	17.6	**23.7**	22.8
SciFact	**72.9**	69.0	70.8	68.8	69.3	68.5	69.3	67.7	64.3
Quora	84.2	78.9	73.0	77.5	83.8	81.7	85.2	**86.5**	83.5
Average	**50.35**	44.48	45.02	46.97	**50.56**	49.99	**49.95**	47.54	43.67

Inference-Free SPLADE

Result (Search Time)

	Client-side P99 latency			Mean throughput		
Client #	BM25†	Ours	Ours†	BM25†	Ours	Ours†
5	13.4	21.7	17.6	784.2	484.8	586.2
10	20.9	25.2	22.9	1150.9	910.4	1024.5
20	35.4	38.2	38.7	1342.1	1183.4	1154.0
40	56.7	66.2	62.3	1658.6	1460.5	1537.7
80	74.7	91.7	81.1	2330.6	1858.19	2073.9

Thank you

Q&A