



# Question Answering over Temporal Knowledge Graphs

시간적 지식 그래프에서의 질의 응답

**DongCheon Lee (이동천)**

Supervisor: Byungkook Oh

Graph & Language Intelligence Laboratory  
Department of Computer Science and Engineering  
Konkuk University

2026.01.13



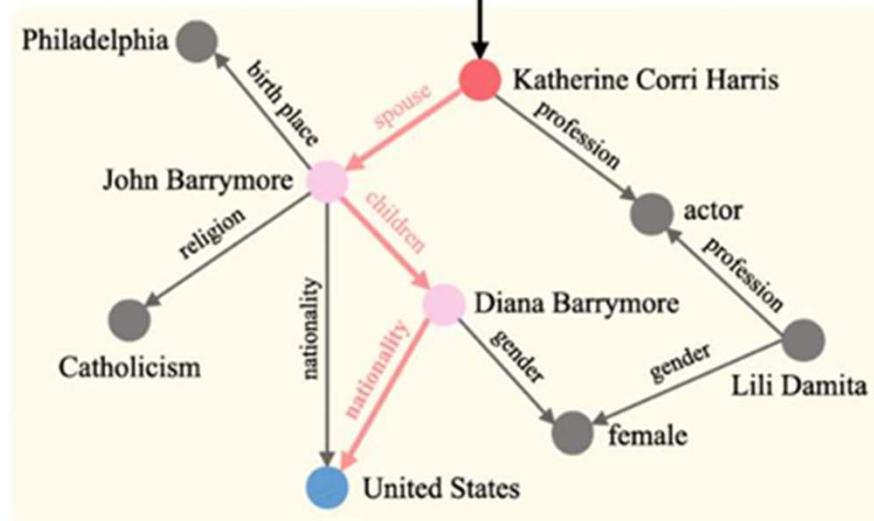
# CONTENTS

1. Background
  - Challenges in TKGQA
2. Approach
  - Traditional Approach
  - Recent Approach (LLM-based TKGQA)
3. 관련 논문
4. 작성 논문 소개

# Background

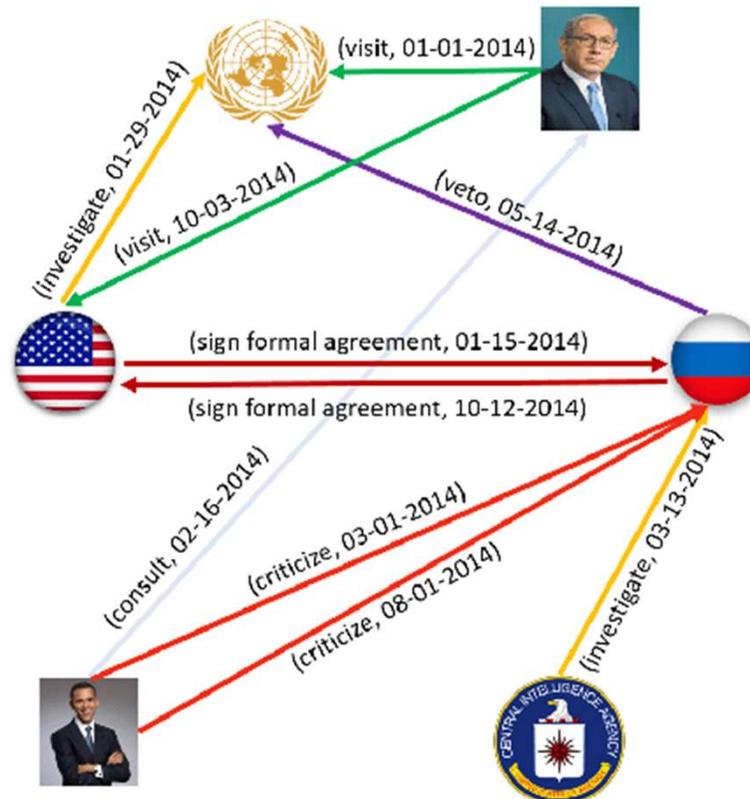
- KGQA (Knowledge Graph Question Answering)
  - ✓ 자연어 질문을 입력 받아
    - 질문과 관련된 정보를 Knowledge Graph에서 탐색
    - 해당 정보를 기반으로 정확한 답변을 생성하는 것을 목표
  - ✓ 즉, KG 내 Entity와 Relation을 활용한 검색, 추론 기반 QA
- KGQA는 정적(Static) 지식그래프를 가정
  - ✓ 실제 세계의 지식은 **정적이지 않음!**

What is the nationality of Katherine Corri Harris 's couple 's children?



# Background

- TKGQA (Temporal Knowledge Graph Question Answering)
  - ✓ 시간 정보가 포함된 지식 그래프(Temporal KG)를 활용하여
    - 시간적 제약이 포함된 질문에 답하는 것을 목표
  - ✓ (Head, Relation, Tail, Timestamp) 형태의 Quadruple로 표현
    - (Joe Biden, make\_a\_visit, Belgium, 2021-06-14)



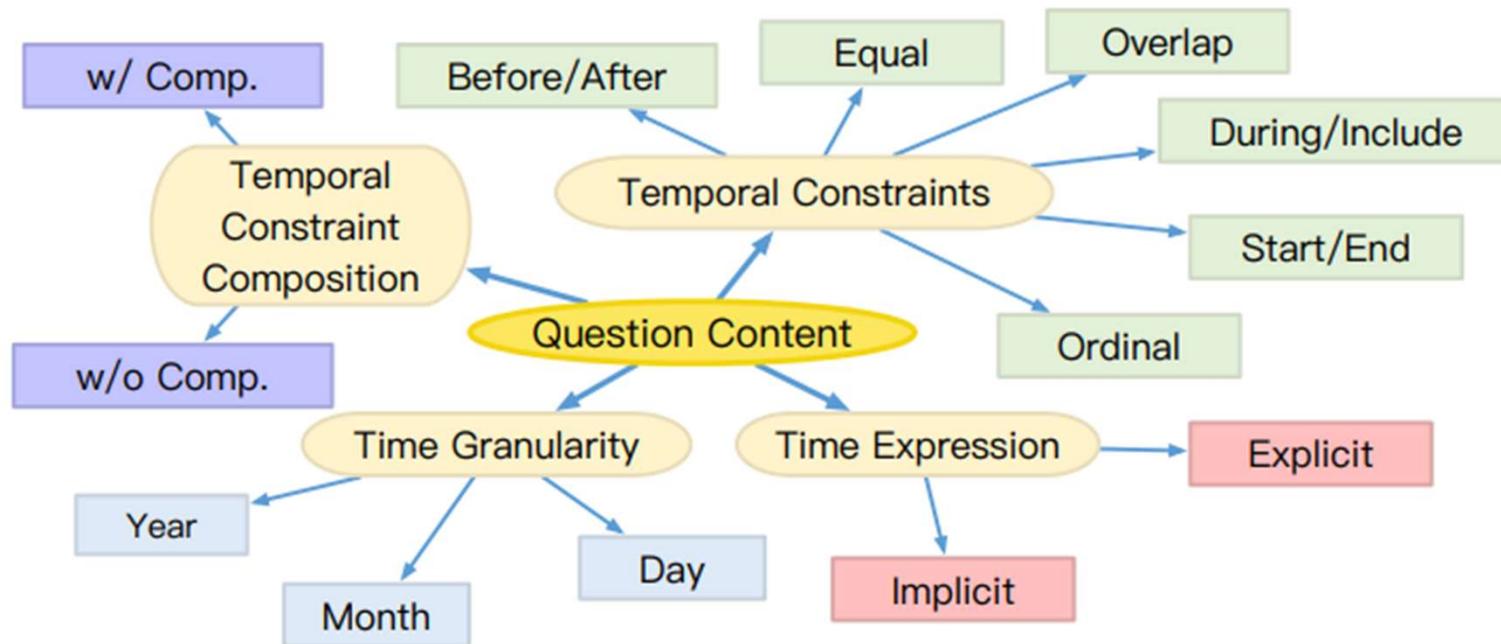
# Background

- KGQA vs TKGQA

	KGQA	TKGQA
지식 그래프	정적(Static) KG	시간 정보를 포함한 Temporal KG
지식 표현	Triple (Head, Relation , Tail )	Quadruple (Head, Relation, Tail, Timestamp)
Answer type	Entity	Entity, Timestamp (또는 시간 조건을 만족하는 Entity)
문제 복잡도	구조적 복잡성 중심	구조적 추론 + 시간 제약을 동시 만족

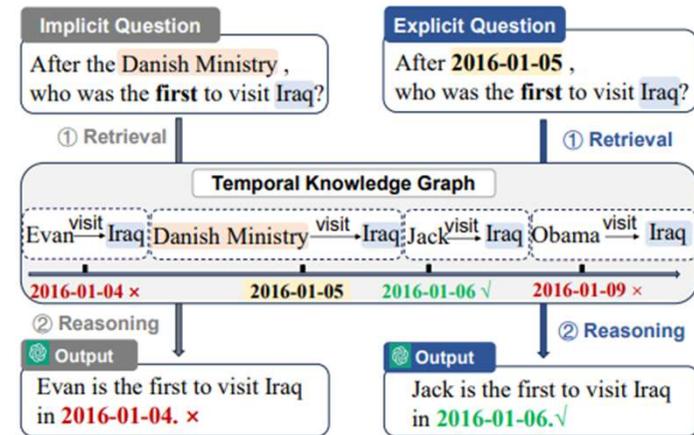
# Background

- 시간적 질문(Temporal Question )
  - ✓ 최소 하나 이상의 시간 제약을 포함하거나, 답변으로 Timestamp를 요구하는 질문
  - ✓ 동일한 질문이라도 시간 제약에 따라 답이 크게 달라질 수 있음
    - Ex) UFC Strawweight의 우승자는 누구야?  
2022년 UFC's strawweight 우승자 -> Carla Esparza  
2024년 UFC's strawweight 우승자 -> Weili Zhang



# Background

- 시간 표현의 세분성 (Granularity)
  - ✓ Year, Month, Day
- 시간 표현 (Temporal Expression)
  - ✓ Explicit
    - Ex) 2026-01-13, September 2023
  - ✓ Implicit
    - 추론을 하기 위해 추가적인 문맥 정보 필요
    - Ex) 2024 파리 올림픽
- 시간 제약 (Time Constraint)
  - ✓ 6가지 관계 유형의 조합
    - Before/After
    - Equal
    - Overlap
    - During/Include
    - Start/End
    - Ordinal



# Background

---

- Answer Type
  - ✓ 질문에서 사용된 의문사에 의해 결정
  - ✓ 종류
    - Entity
      - Who ?
    - Timestamp
      - What Year?, When ?
- Complexity
  - ✓ Simple Question
    - 하나의 Fact만으로 답을 도출
  - ✓ Complex Question
    - 여러 Fact를 통합하여 답을 도출
    - Ex) 오바마 이전의 미국 대통령은 누구인가?
      - (Obama, President of, USA, 2009, 2017) -> 2009년 이전
      - (George W. Bush, President of, USA, 2001, 2009)

# Challenges in TKGQA

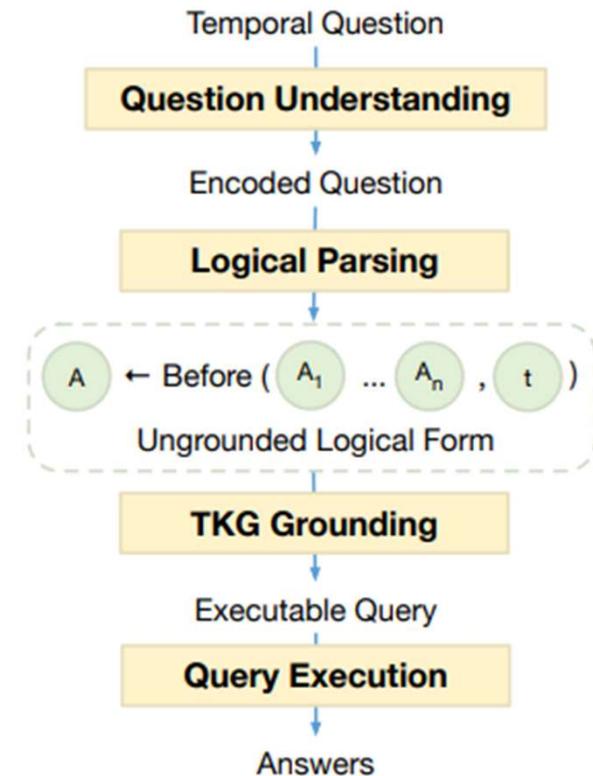
---

- Temporal Expression Understanding
  - ✓ 많은 질문들이 명시적인 Timestamp 대신 암시적인 시간 표현을 포함
  - ✓ Ex) After the Danish Ministry, who was the first to visit Iraq?
  - 질문을 명확한 시간 조건으로 변환하는 추론 필요
- Temporal Constraint Reasoning
  - ✓ 시간 관계는 단순한 필터링이 아님
  - ✓ Ex) Before/ After, During/Overlap
  - 다중 제약의 조합을 동시 만족해야 함
- Multi-hop Temporal Reasoning
  - ✓ 복잡한 질문은 여러 Fact, 시간 제약, 순서를 모두 고려해야 함
  - ✓ Ex) Who was the U.S. president before Obama?
  - 시간 + 구조적 추론 수행

# Traditional Approach

## 1. Semantic Parsing (SP-based)

- ✓ 자연어 질문을 TKG를 질의하기 위한 논리 표현으로 변환
- ✓ 질문에 포함된
  - Entity
  - Relation
  - Time Constraint를 추출하여 논리적으로 해석 가능한 형태로 변환
- ✓ Ex)
  - TEQUILA
  - SYGMA
- ✓ 장점
  - 질의가 잘 구성된 경우 높은 정확도를 보임
- ✓ 단점
  - Logical form 내의 구문 오류로 인해 질의 실행 실패



# Traditional Approach

## 2. TKG Embedding (TKGE-based)

✓ 질문과 TKG의 Quadruple의 저차원 벡터로 인코딩한 두 벡터 의미 유사도에 따라 랭킹

✓ Ex)

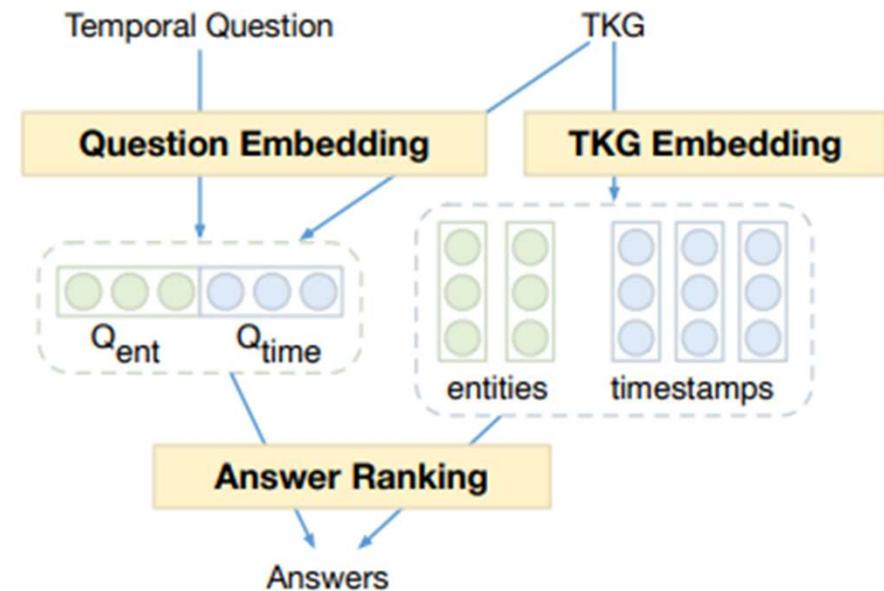
- TTransE
- CronKGQA
- TComplEx

✓ 장점

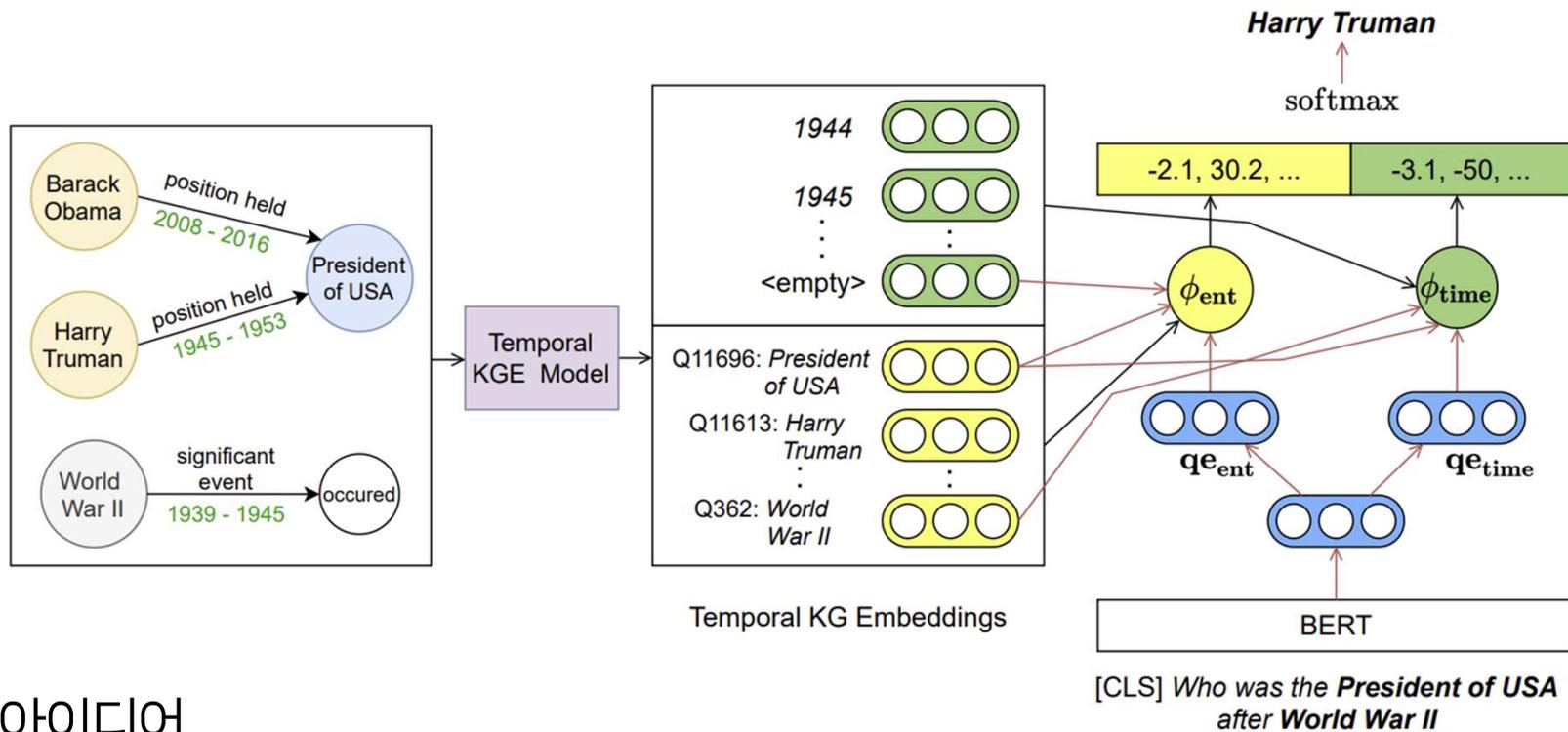
- 확장성, 일반화 능력
- 효율적 검색 가능

✓ 단점

- 복잡한 시간적 질문에서는 성능이 크게 저하
  - (암묵적) 시간 제약의 의미 정보를 이해하지 못함
- 시간 표현 정밀 반영 어려움



# [2021][ACL][CronKGQA]



- 핵심 아이디어
  - ✓ 질문과 TKG를 같은 임베딩 공간에 정렬
- 방법
  - ✓ 질문을 Query embedding으로 변환
  - ✓ TKG Fact를 TKGC 모델을 통해 Time-aware Fact Embedding으로 표현
  - ✓ Query embedding과 Fact Embedding 간의 유사도 비교
- 한계점
  - ✓ 시간 제약을 암묵적 표현에 의존

# Recent Approach (LLM-based TKGQA)

- 전통적인 TKGQA 방법은 고정된 규칙 또는 임베딩 공간에 의존
- 복잡한 시간 질의를 구조적으로 처리하는 데 한계
  - ✓ Implicit
  - ✓ Multi-hop
  - ✓ Multi-constraint
- 최근 LLM의 추론 능력을 활용하여
  - 시간 질의를 단계적으로 이해하고 해결하려는 접근 등장



# Core Idea of LLM-based TKGQA

---

- Query Decomposition with LLM

Q : Before Kuwait, which country received the visit last?

- ✓ Sub-question Decomposition

- 복잡한 질문을 여러 개의 하위 질문으로 분해
- 각 Sub-question은 독립적으로 답할 수 있는 형태

Ex )

- Q1: When did Kuwait receive the visit?
- Q2: Which countries received the visit before #1?
- Q3: Which one is the latest?

- ✓ Plan- based Decomposition

- 질문을 푸는 과정을 Plan으로 분해
- 질문이 아닌 연산 단위를 생성

EX

- 1. FIND event: Kuwait received visit → t
- 2. FILTER events before t
- 3. SELECT last event

# Core Idea of LLM-based TKGQA

---

- Retrieval-Augmented Generation
  - ✓ 질문과 관련된 지식을 외부지식(TKG)로부터 검색한 후 추론
    - 근거 기반의 답을 추출할 수 있음
  - ✓ 단순히 유사도 기반이 아닌 질문 시간 조건에 맞는 근거 검색/재정렬
    - 일반적인 RAG는 유사도 기반의 검색
      - 시간 정보가 무시
    - Retrieval 단계에서 시간 정보 고려
      - Timestamp
      - 기간
      - Time Constraint



# [2025][EMNLP][RTQA]

Query Decomposition (Sub-question)

RTQA : Recursive Thinking for Complex Temporal Knowledge Graph Question  
Answering with Large Language Models

# [2025][EMNLP][RTQA]

---

- 기존 문제점

- ✓ 복잡한 시간 질의에 대한 제한적인 추론 능력
  - LLM의 내부지식 의존
  - 암시적 시간 제약 처리 X
  - 복합적인 제약조건을 처리하는데 어려움
- ✓ Decomposition 기반 방식의 오류 전파 문제

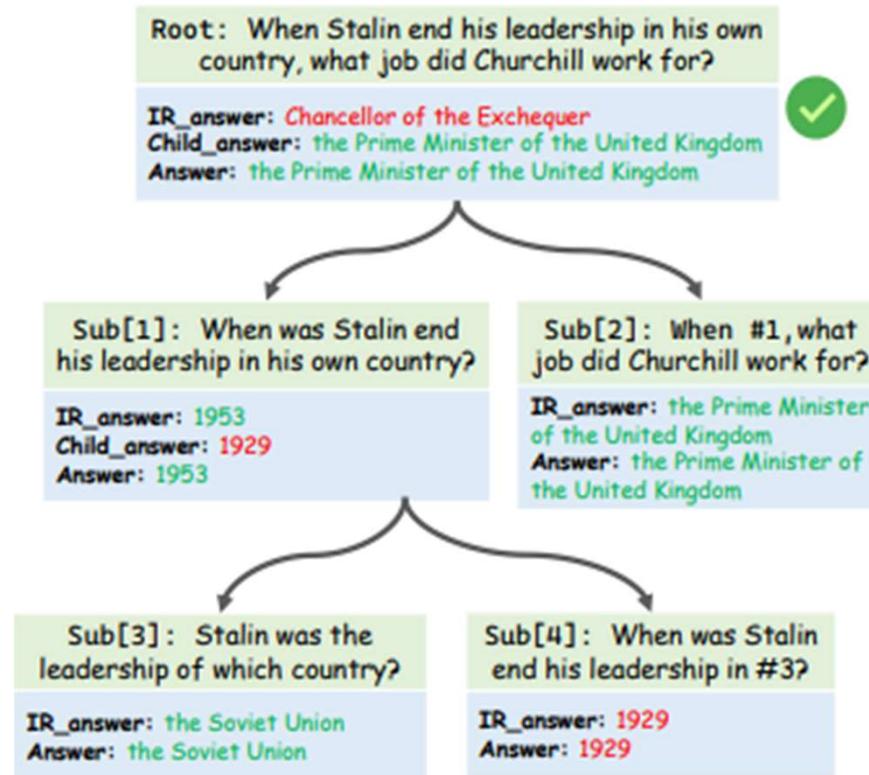
- 핵심 아이디어

- ✓ 질의를 하위 질문으로 분해
  - 재귀적인 Bottom-up 추론을 수행
- ✓ Divide-and-conquer 전략
  - 암시적 시간 정보 추출
  - Before, Last 제약 적용
- ✓ Multi-path answer aggregation 모듈을 통해 신뢰도 높은 답변 선택

# [2025][EMNLP][RTQA]

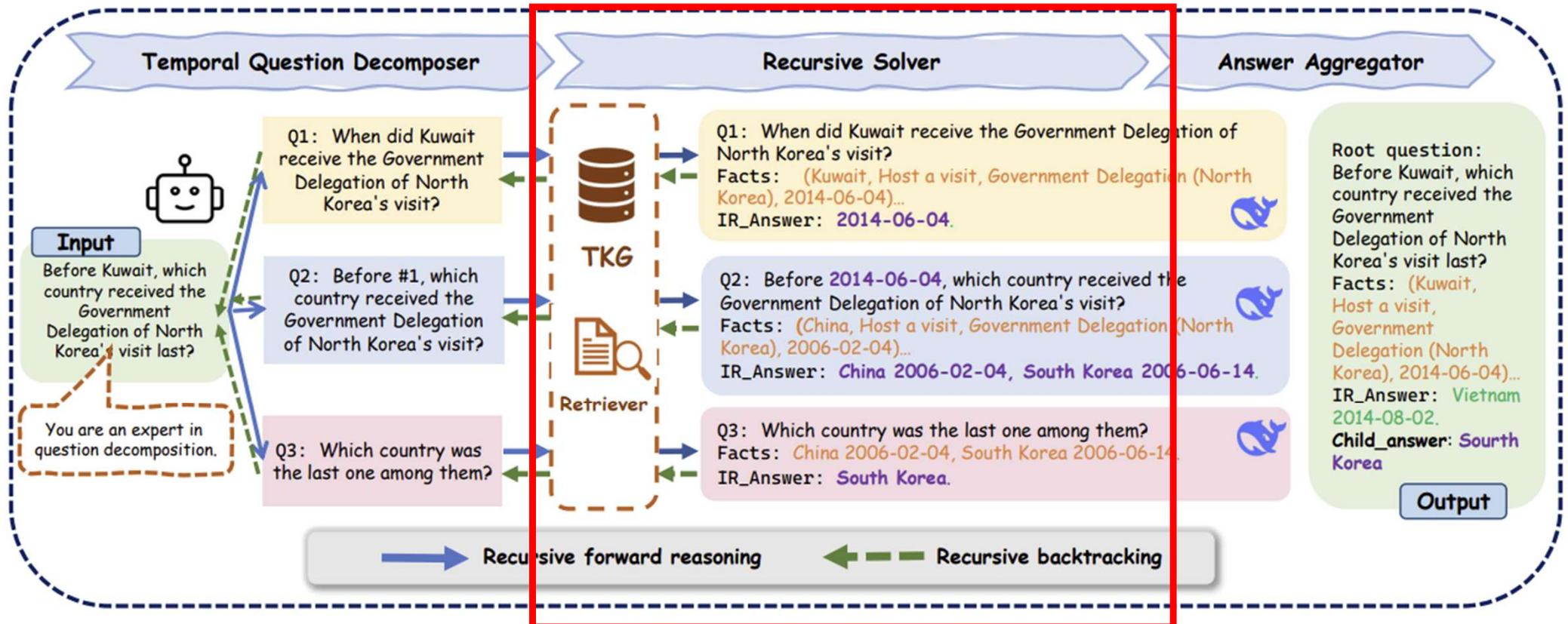
- Temporal Question Decomposer
  - ✓ LLM을 통해 복잡한 시간 질의를 여러 하위 질문들로 구성된 Tree로 분해
  - ✓ 시간 질문의 유형별로 프롬프트를 다르게 설정

(a) RTQA



# [2025][EMNLP][RTQA]

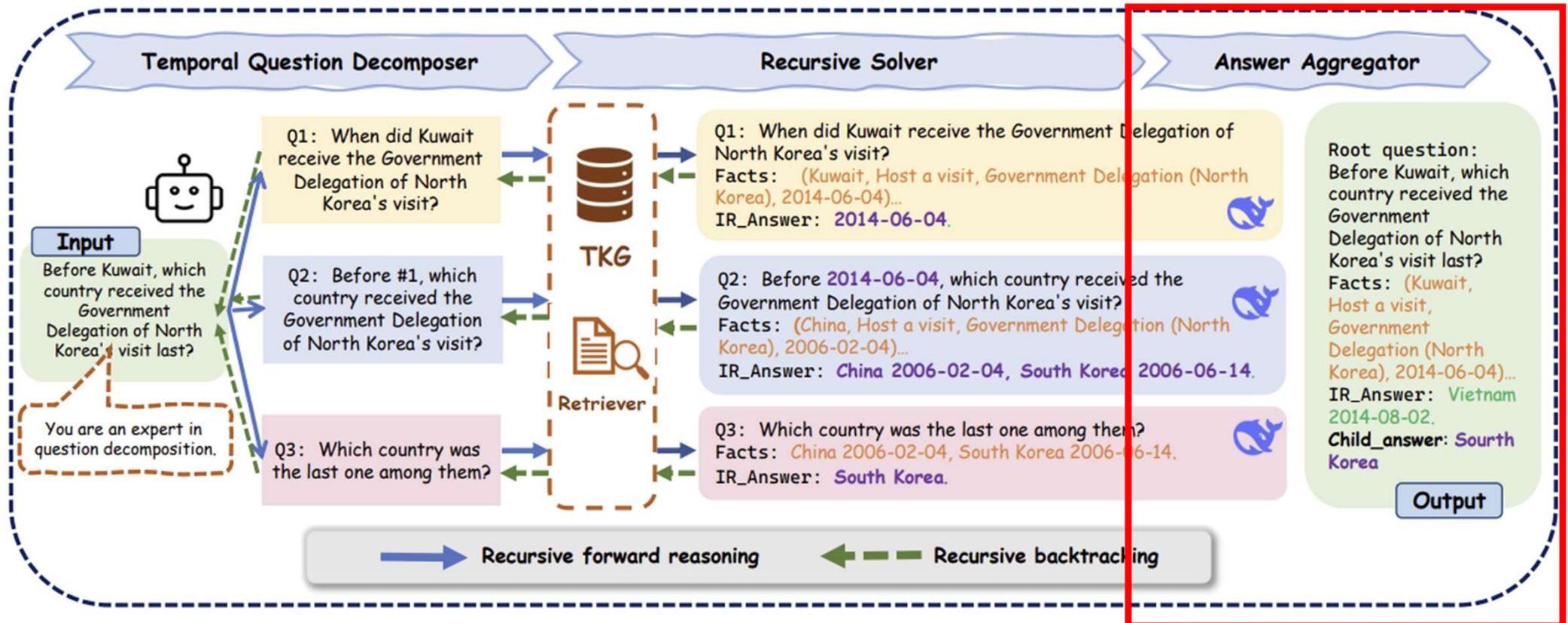
- Recursive Solver
  - ✓ Botton-up 방식
  - ✓ Leaf 노드를 먼저 TKG로부터 관련 사실 검색 후, LLM을 호출하여 답변생성
    - Verbalization 후 Dense Encoder를 통해 유사도 기반 Top-k
  - ✓ Leaf 노드가 아닌 경우, 해당 노드의 모든 자식 노드를 순차적으로 처리
    - 이전 하위 질문들의 답변을 반영하여 추론



# [2025][EMNLP][RTQA]

- Answer Aggregator

- ✓ LLM을 통해 다음 두 답변을 집계하여 최종 답변 선택
  - 원본 질문과 관련된 TKG Fact를 검색한 후 LLM 추론을 적용하여 생성된 답변
  - 질문 Tree의 자식 노드들로부터 얻어진 답변





# [2025][Arxiv][PoK]

Query Decomposition (Plan)

Plan of Knowledge: Retrieval-Augmented Large Language Models for Temporal Knowledge Graph Question Answering

# [2025][Arxiv][PoK]

- 기존 문제점
  - ✓ 자연어 질문에 내재된 복잡한 의미적, 시간적 제약을 완전히 포착하기 어려움
  - ✓ Temporal constraint를 정확히 이해하지 못함
  - ✓ 텍스트 기반 RAG는 의미적 유사도에 의존
    - 질문에 포함된 Temporal constraint를 무시
- 핵심 아이디어
  - ✓ 복잡한 질문을 Sub-objective로 분해하여 TKG의 시간 지식을 탐색
  - ✓ Semantic 정보와 Temporal 정보를 동시에 고려하는 Retriever
    - Question representation에 Task Prompt 주입
    - Contrastive Learning

## (5) Plan of Knowledge

*Break task into sub-objectives:*

**Response:**

1. Retrieve: When Danish visit Iraq?
2. Rank: Rank events by time.
3. Reason: Who is the first?

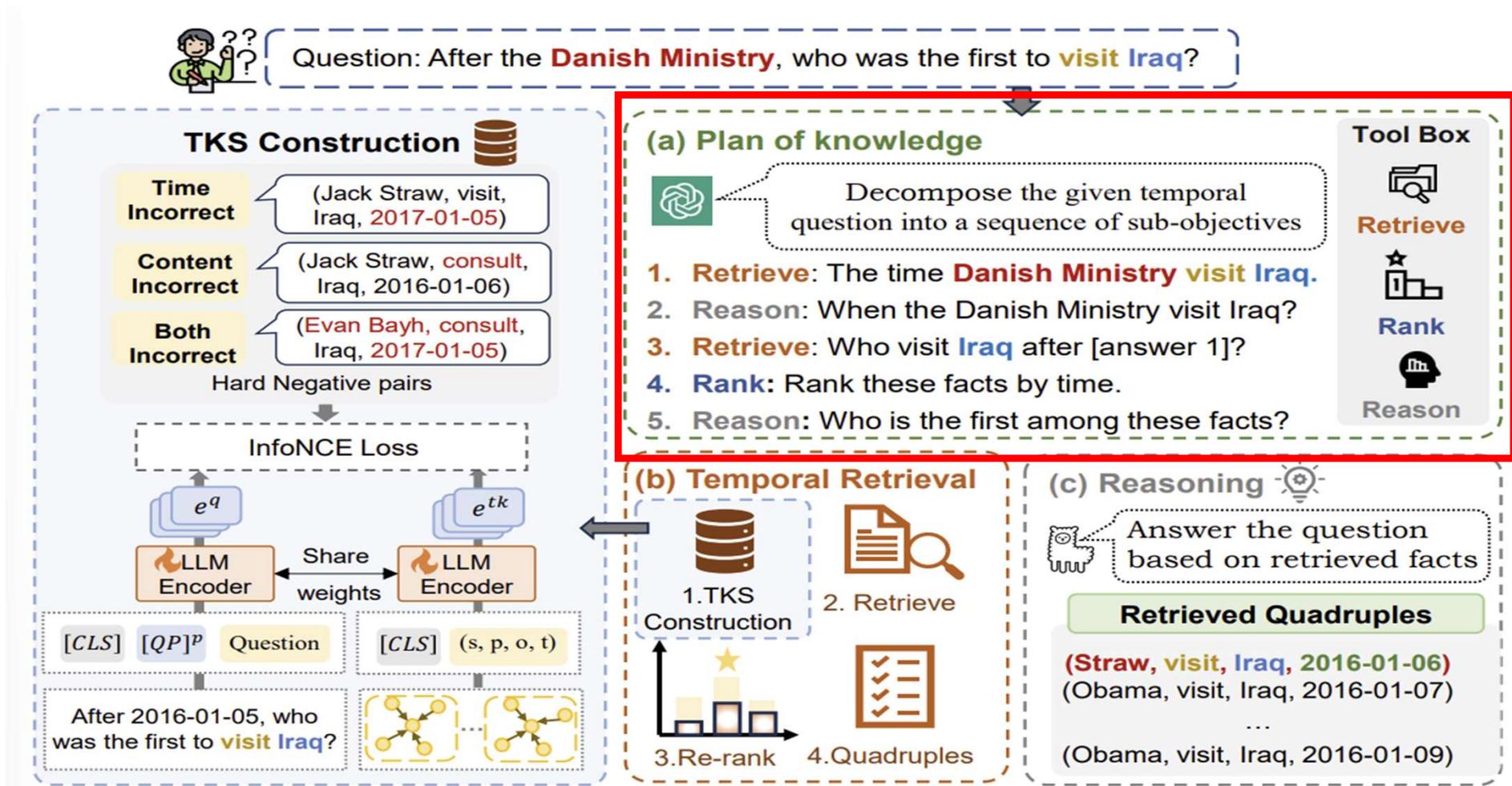


# [2025][Arxiv][PoK]

- Plan of Knowledge

- ✓ LLM을 활용하여 Plan을 생성

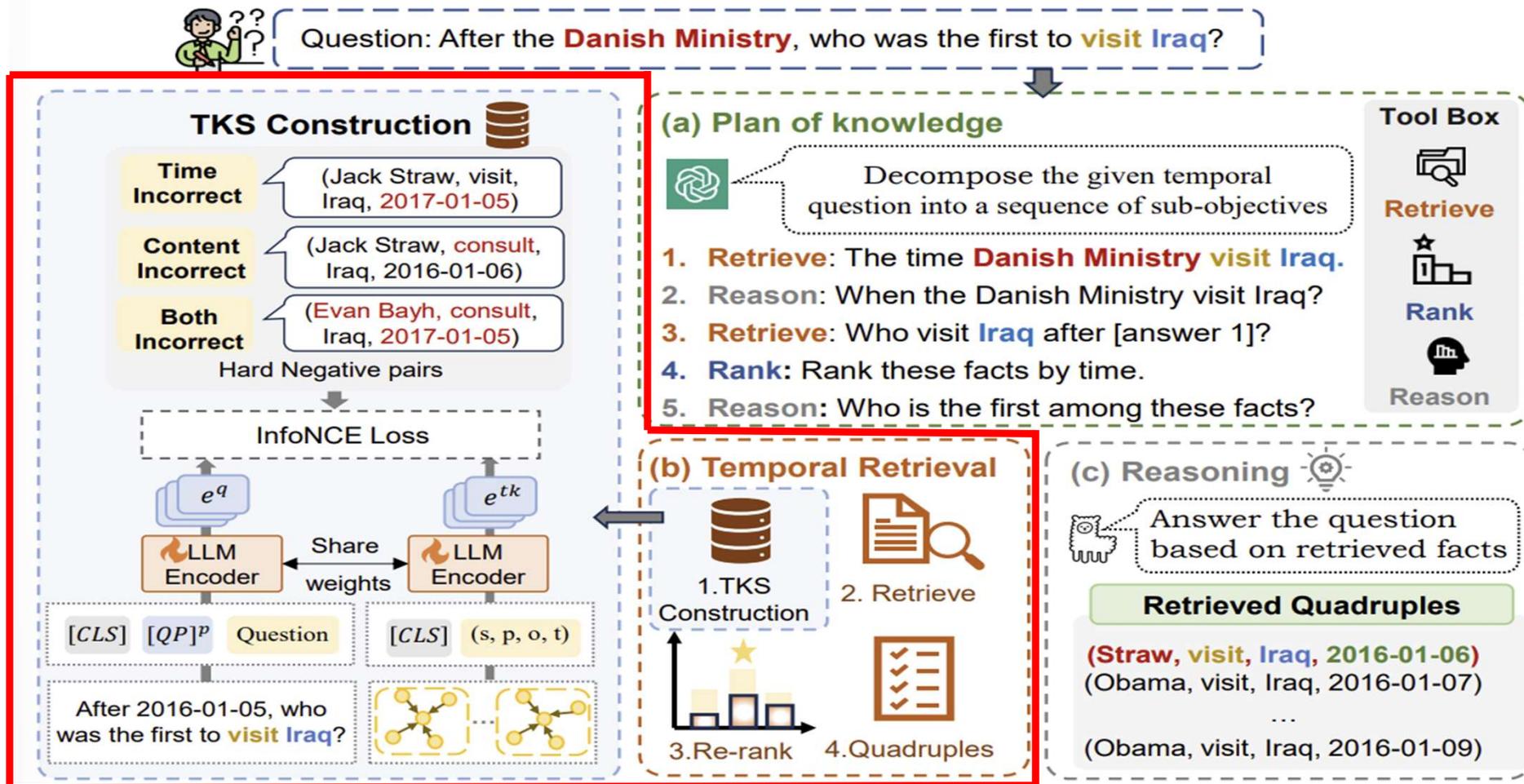
- 이후 Retrieval, Reranking, Reasoning의 가이드 역할



# [2025][Arxiv][PoK]

- Prompt-guided Temporal Encoding

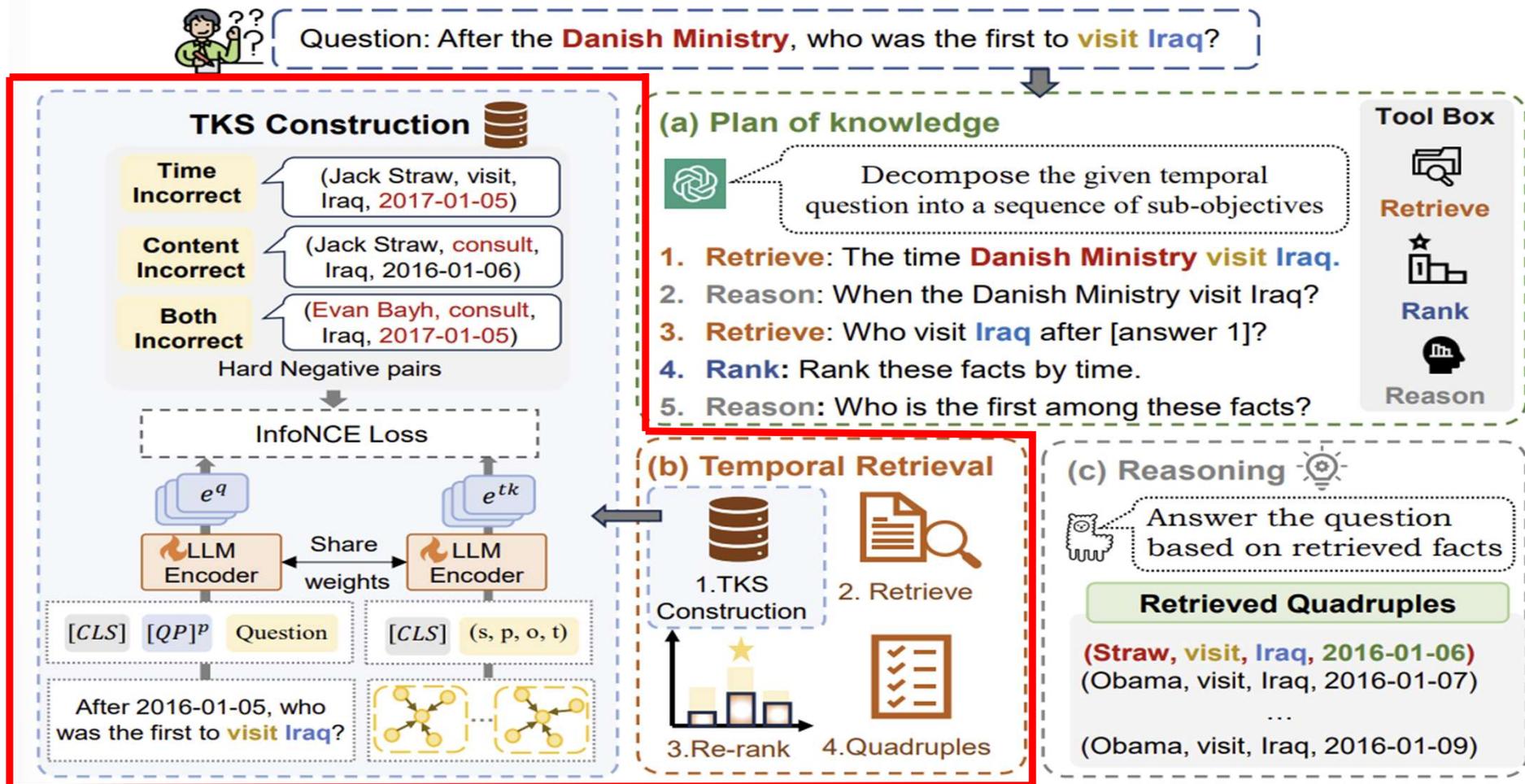
- ✓ Prompt Token을 입력 sequence에 삽입하여, Temporal Constraint에 집중
- ✓ 학습 가능한 벡터로 표현



# [2025][Arxiv][PoK]

- Contrastive Time-aware Fine-Tuning

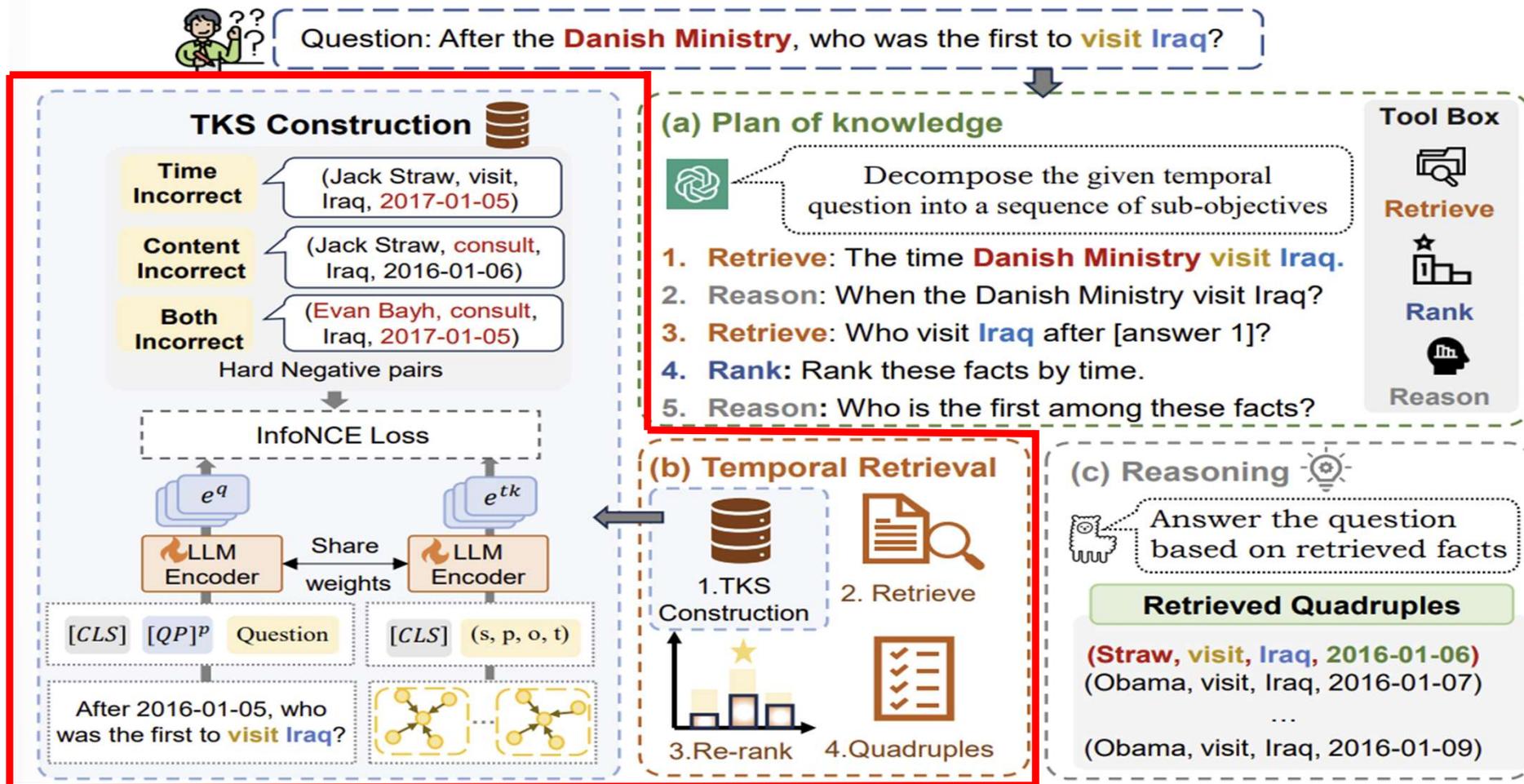
- ✓ 시간적 제약을 고려하기 위해, 공유 임베딩 공간 기반 Contrastive learning 도입
- ✓ 3가지 유형의 negative sample 생성
  - InfoNCE Loss를 통해 의미적으로 비슷하지만 시간적으로 잘못된 Fact 구별



# [2025][Arxiv][PoK]

- Temporal Retrieval and Reranking

- ✓ 질문 임베딩과 TKS에 저장된 모든 Fact와 코사인 유사도를 계산 (FAISS)
- ✓ Time Filtering





# [2026][ICLR\_under][EvoReasoner]

Query Decomposition (Plan)

Temporal Reasoning over Evolving Knowledge Graphs

# [2026][ICLR\_under][EvoReasoner]

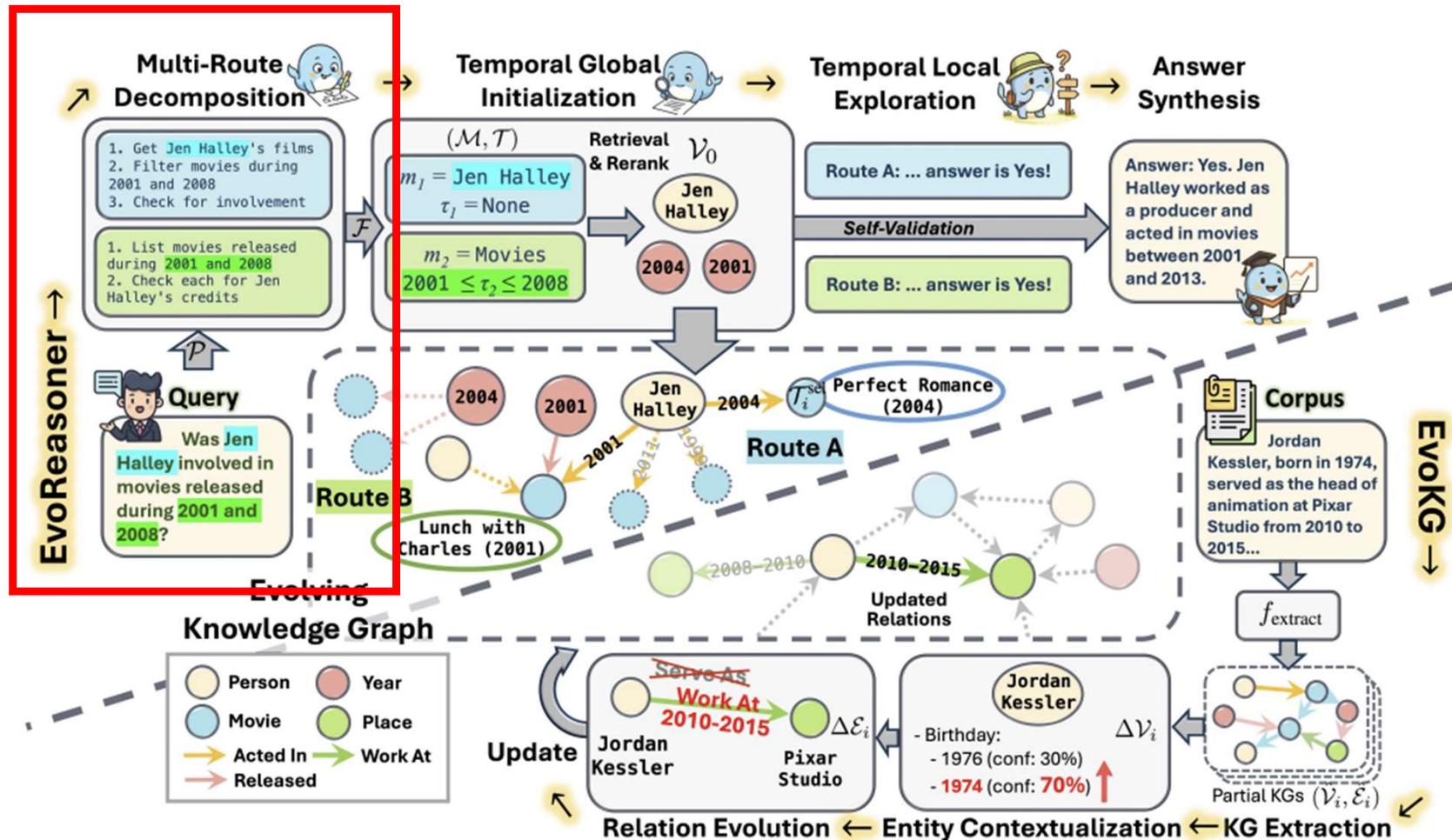
---

- 기존 문제점
  - ✓ KG-augmented 접근법은 KG의 Static Snapshot에 의존
  - ✓ 실세계의 지식에 내재된 시간적으로 변화하는 동적 특성을 간과
- 제안 방법
  - ✓ Multi-Route Decomposition
    - 질문을 여러 개의 Reasoning route(Plan)로 분해
  - ✓ Global Initialization
    - 의미적, 시간적으로 정렬되는 Topic entity 선택
  - ✓ Local Exploration
    - Beam search
    - 이웃하는 relation과 entity에 대해서 질문과의 관련성을 점수화

# [2026][ICLR\_under][EvoReasoner]

- Multi-Route Decomposition

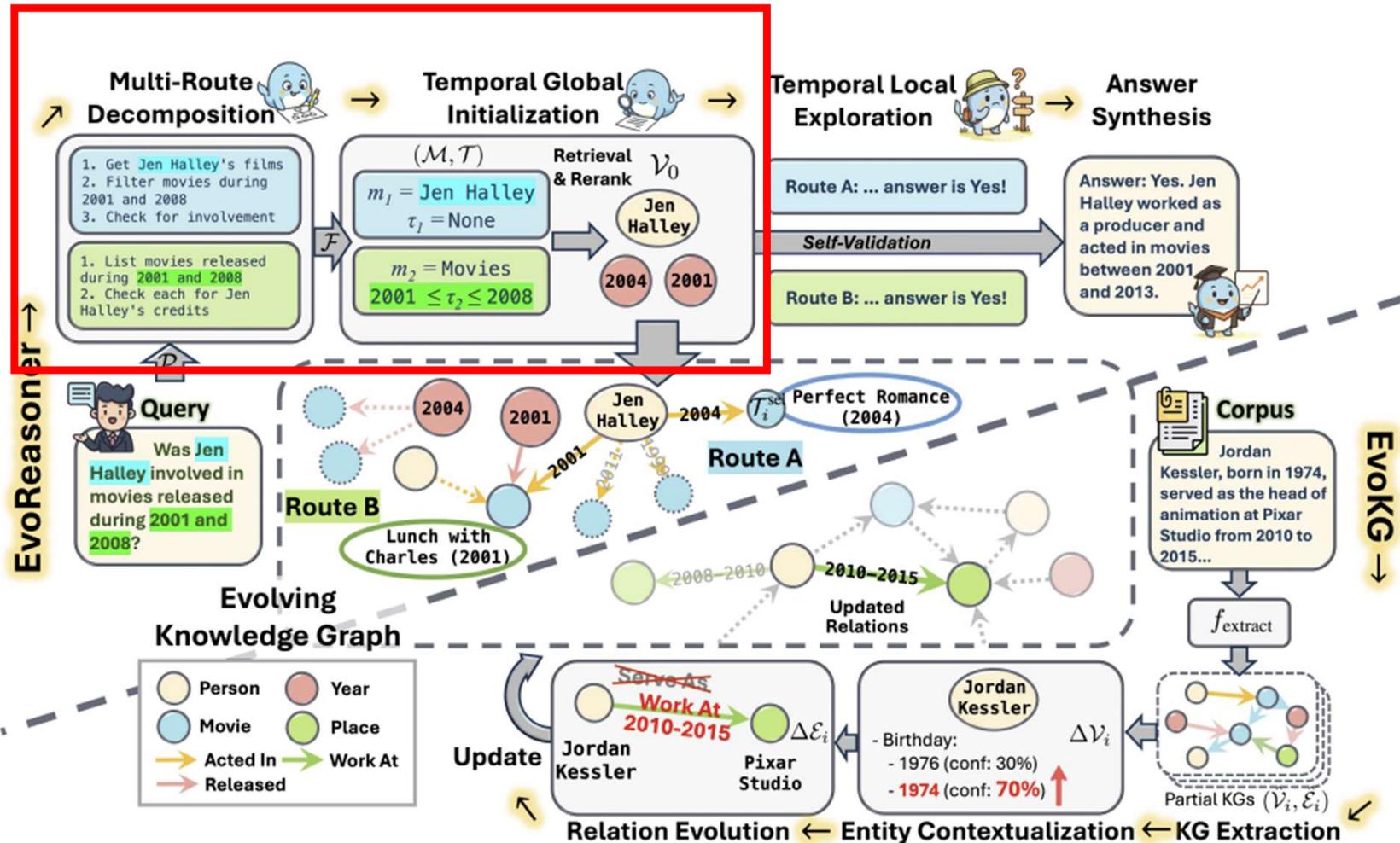
- ✓ 질문을 여러 개의 Route(Plan)으로 분해
- ✓ 각 Path의 Cost를 계산하기 위해 탐색 복잡도 반영
  - 가능한 후보 relation type 개수, relation을 통해도달 가능한 entity 수, 필요 hop 수



# [2026][ICLR\_under][EvoReasoner]

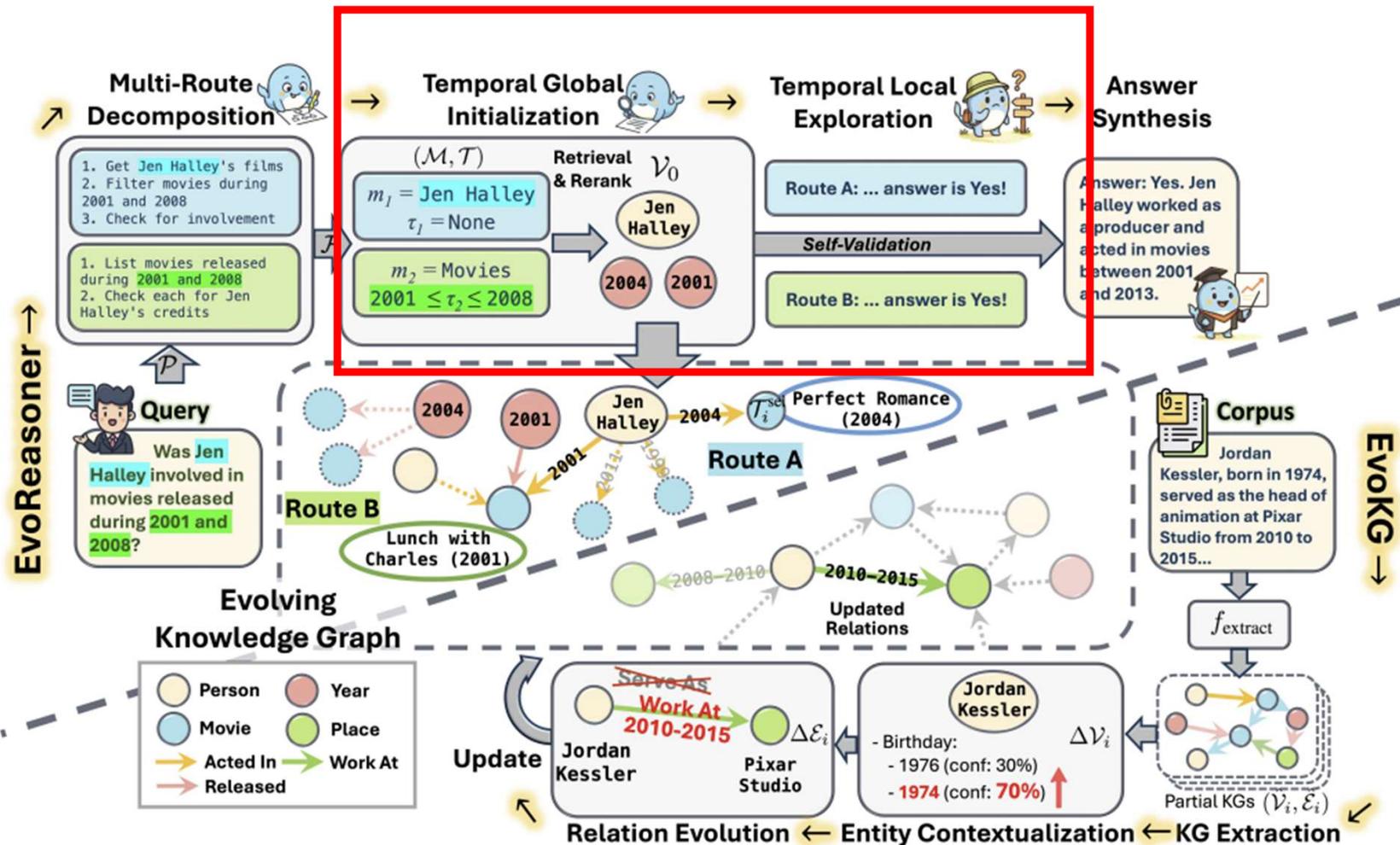
- Global Initialization

- ✓ 질문에서 언급된 entity( $M$ )와 그에 해당하는 시간 문맥 추출( $T$ )  $\rightarrow (M, T)$
- ✓  $(M, T)$ 를 함께 임베딩 후, 모든 entity embedding과 유사도 기반 Top-k 노드 선택
- ✓ LLM을 기반으로 관련성 기준 Reranking



# [2026][ICLR\_under][EvoReasoner]

- Temporal-aware Local Exploration
  - ✓ LLM이 Subgoal과 relation 관련도를 scoring하여 Relation 후보 선택
  - ✓ 후보 edge/entity를 질문 시간과의 Overlap으로 Temporal Filtering
  - ✓ 각 quadruple을 Verbalization 후 유사도 비교





# [2026][ICLR\_under][STAR-RAG]

Right Answer At The Right Time –temporal Retrieval-augmented Generation via Graph Summarization

# [2026][ICLR\_under][STAR-RAG]

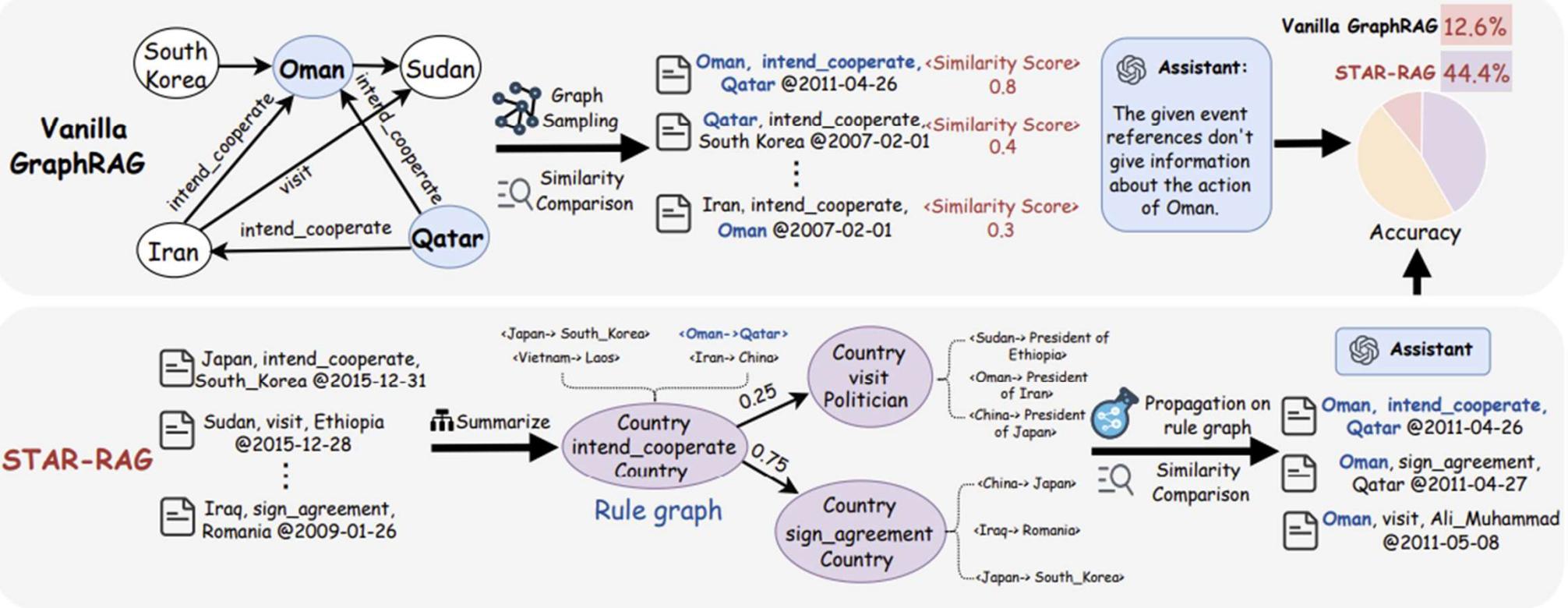
---

- 기존 문제점
  - ✓ 실세계의 지식은 본질적으로 Temporal
  - ✓ Vanilla GraphRAG는 Time-sensitive Query에 대해서 한계를 가짐
    - 의미적 유사도 기반의 검색은 temporal constraint를 간과
  
- 핵심 아이디어
  - ✓ 시간과 정렬된 Rule Graph
    - Rule graph에서 Propagation 수행하여 검색 공간 ↓
  - ✓ Query가 주어지면, Seed event를 식별하고 Personalized PageRank

# [2026][ICLR\_under][STAR-RAG]

- Vanilla GraphRAG와의 비교

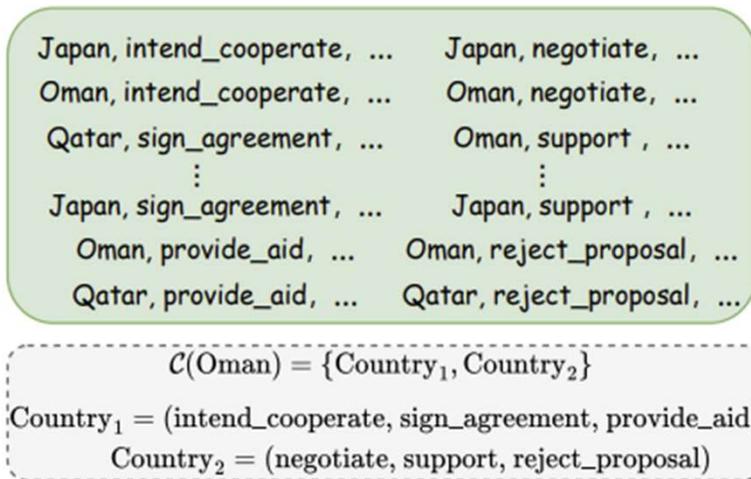
Question: After establishing diplomatic cooperation with Qatar, what did Oman immediately do?



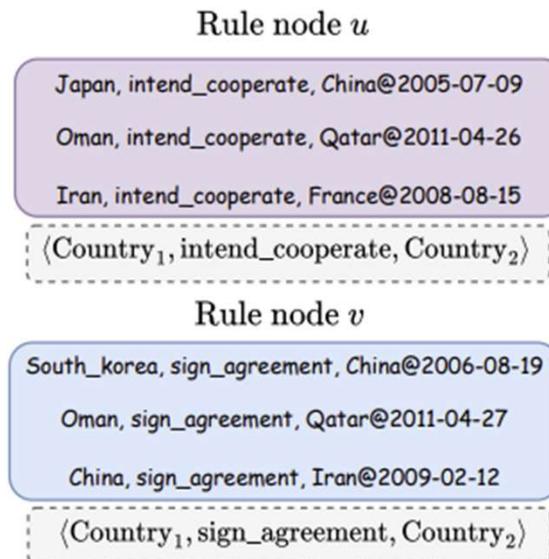
# [2026][ICLR\_under][STAR-RAG]

- Rule Graph Building

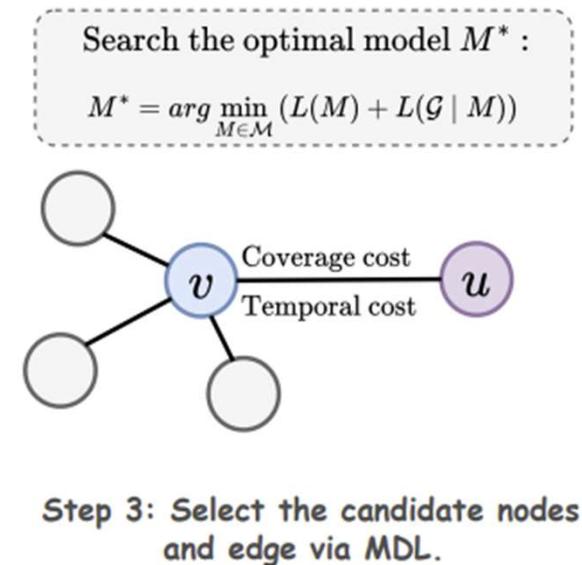
- ✓ Event를 Rule node로 요약하고 데이터로부터 학습된 Time-sensitive edge를 연결
  - Temporal “What-follows-What” 신호를 보존하고 효과적인 전파 가능



Step 1: Entity labeling via mine frequent relation subsets.



Step 2: Map the events to candidate rule nodes.

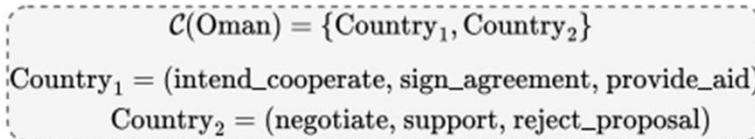


Step 3: Select the candidate nodes and edge via MDL.

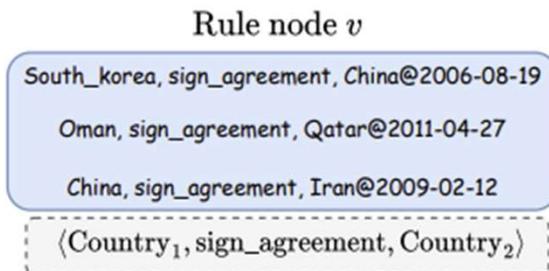
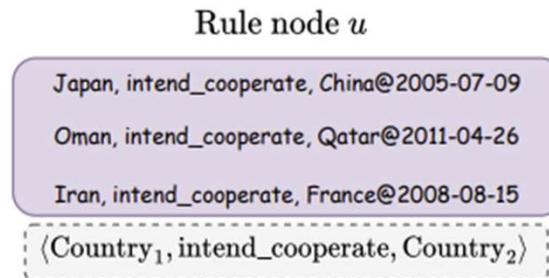
# [2026][ICLR\_under][STAR-RAG]

- Entity Labeling

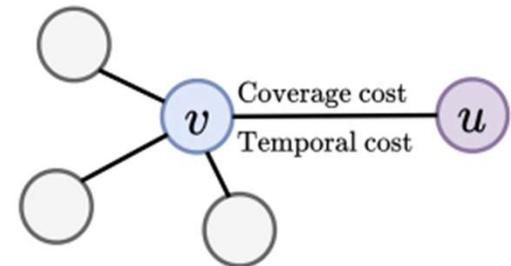
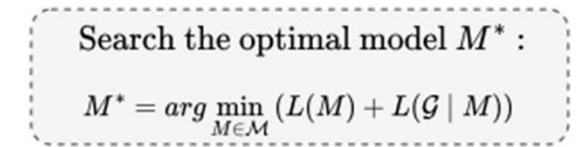
- ✓ 모든 Entity에 대해, 해당 entity가 등장하는 Relation 집합을 수집
- ✓ Apriori 알고리즘으로 자주 함께 등장하는 relation subset을 찾고, 조합에 Labeling
- ✓ 각 entity는 자신이 포함하는 relation 조합에 대응되는 label로 매핑
  - Ex) Oman → Country 1



Step 1: Entity labeling via mine frequent relation subsets.



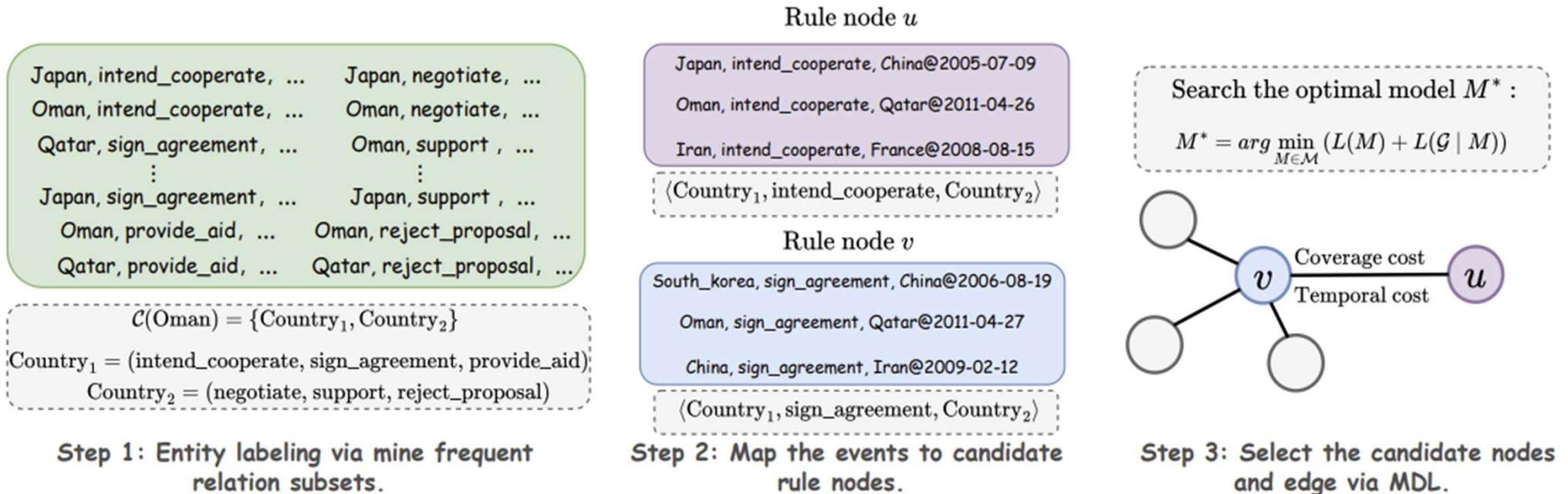
Step 2: Map the events to candidate rule nodes.



Step 3: Select the candidate nodes and edge via MDL.

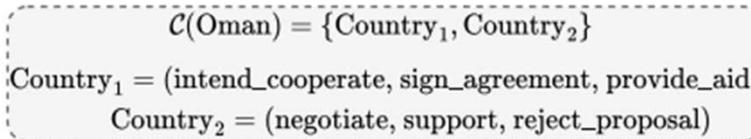
# [2026][ICLR\_under][STAR-RAG]

- Node
  - ✓ 각 Temporal Event (Quadruple)을 (label, relation, label) 형태의 Rule Schema로 바꿈
    - (Oman, Intend\_cooperate, Qatar, 2011-04-26) → (Country, intend\_cooperate, County)
- Edge
  - ✓ Hamming 거리로 후보 Edge 생성
    - $\langle a_s, r, a_o \rangle, \langle b_s, r', b_o \rangle \Rightarrow 1[a_s \neq b_s] + 1[r \neq r'] + 1[a_o \neq b_o]$
  - ✓ Rule Node에 포함된 event 중에서 많이 겹치는 event 쌍만 골라 시간 차이 계산
    - 시간 적으로 어떤 패턴이 이어지는지 확인

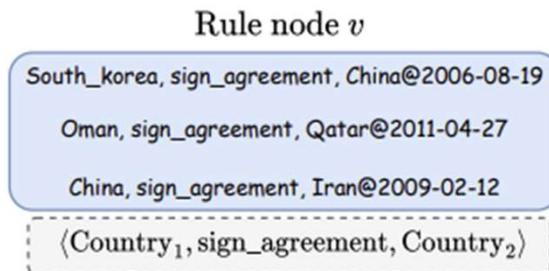
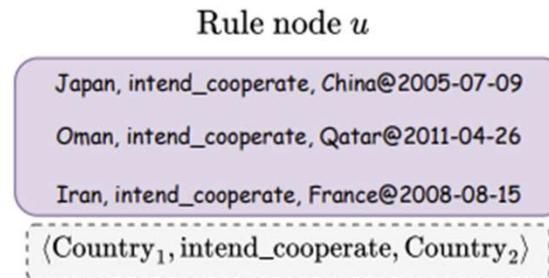


# [2026][ICLR\_under][STAR-RAG]

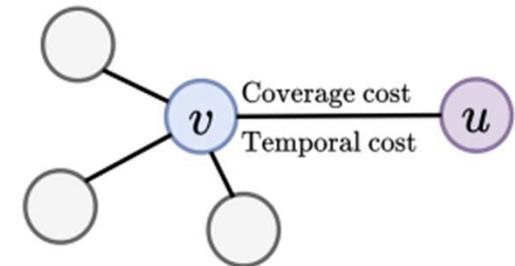
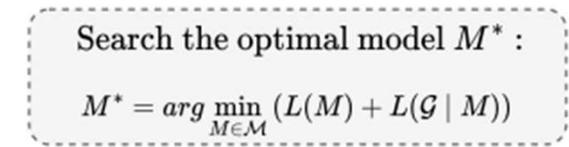
- MDL (Minimum Description Length)
  - ✓ Coverage Cost
    - 자주 이어지는지 (얼마나 많은 event)
    - Support set의 크기
  - ✓ Temporal Cost
    - 시간 차이가 일정한 패턴을 보이는지



Step 1: Entity labeling via mine frequent relation subsets.



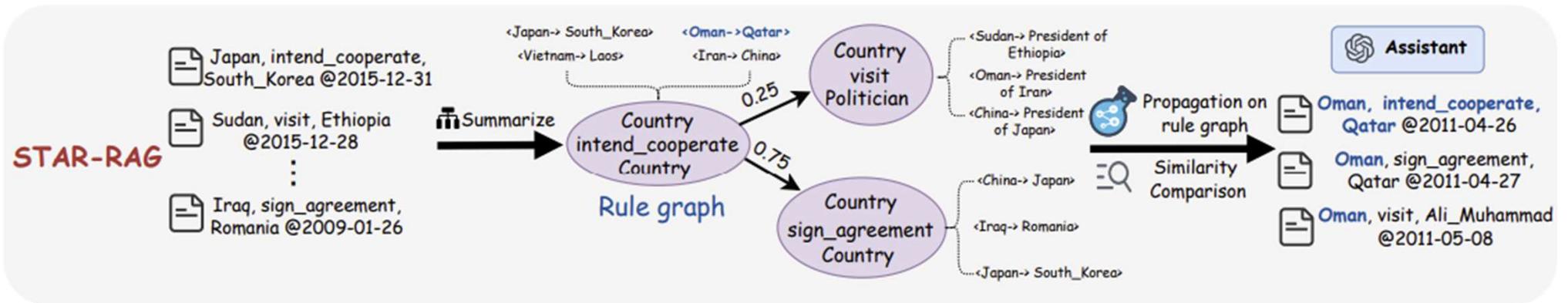
Step 2: Map the events to candidate rule nodes.



Step 3: Select the candidate nodes and edge via MDL.

# [2026][ICLR\_under][STAR-RAG]

- 자연어 질문을 Rule Graph 상의 Seed로 변환 ( Personalization Vector)
- Rule Graph 위에서의 Propagation
  - ✓ Personalized PageRank (PPR)
    - 질문과 관련된 Rule에 가중치
- Top-k Rule Node 선택 후 Rule Node에 포함된 Fact 수집





# 작성 논문 소개 : TARD

Diversifying Differentiable Graph Retrieval with Topic-Adaptive Multi-Intent Learning

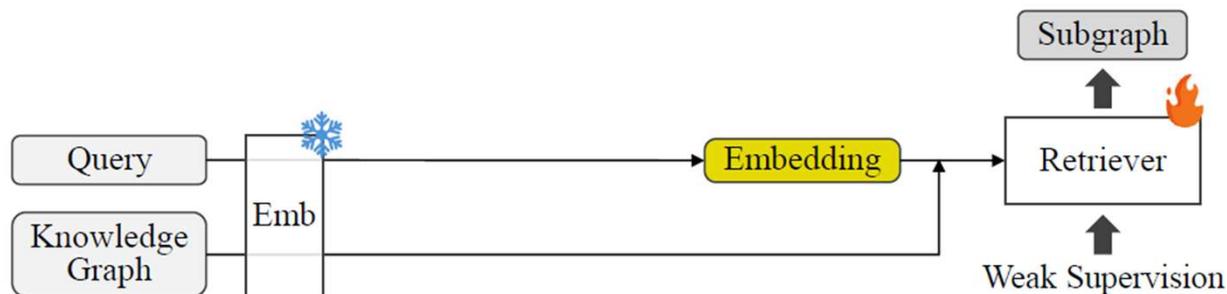
# TARD

- 기존 문제점

- ✓ 각 질의를 하나의 표현으로 압축
  - 실제 질의는 여러 의미적인 측면을 포함하기 때문에 다양한 관점에서의 검색 필요
- ✓ 미분 불가능한 이산적인 선택에 의존
  - Retriever, Reasoner를 독립적으로 최적화

- 핵심 아이디어

- ✓ 토픽 모델링으로부터 추출된 다중 의도를 활용하여 Retriever와 Reasoner를 공동 최적화
  - Gumbel Softmax기반 STE(Straight Through Estimator)
- ✓ Adaptive DPO
  - 정보성이 높은 트리플은 강화하고 정보성이 낮지만 빈도만 높은 트리플 억제

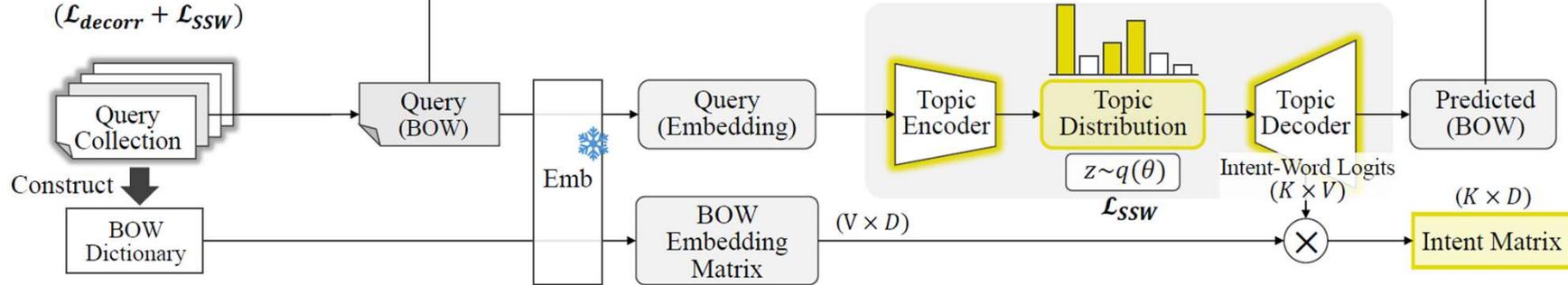


(a) Single-Embedding Retrieval

# TARD

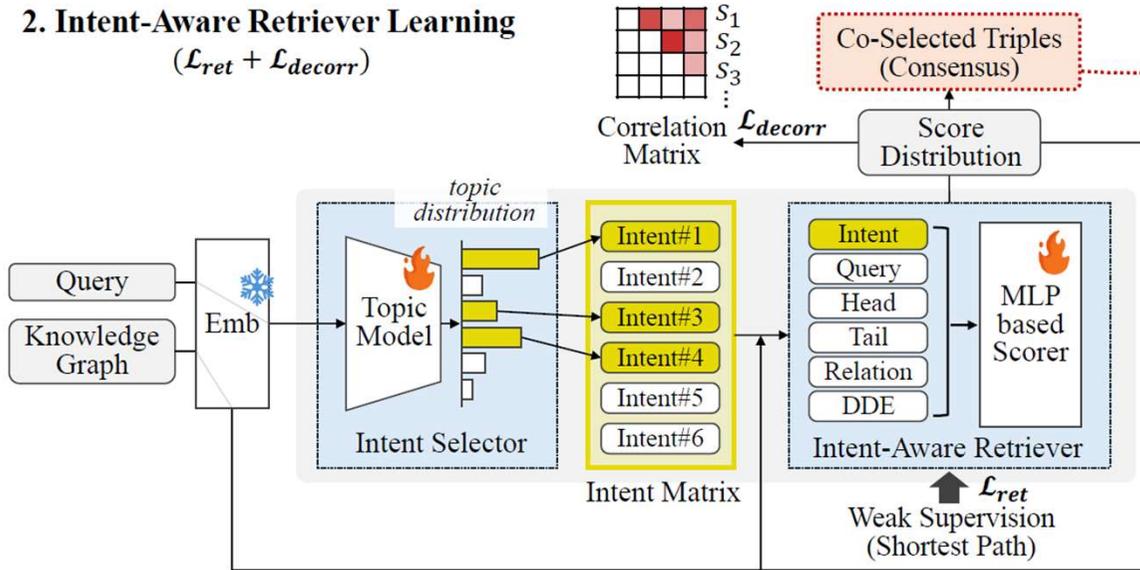
- Overview

## 1. Topic-Adaptive Intent Learning



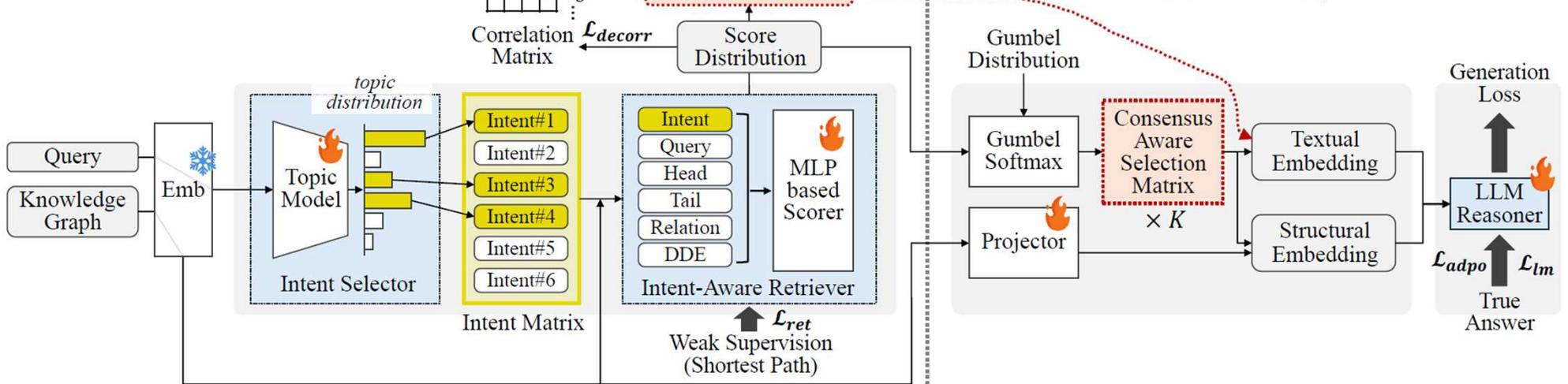
## 2. Intent-Aware Retriever Learning

$(\mathcal{L}_{ret} + \mathcal{L}_{decorr})$



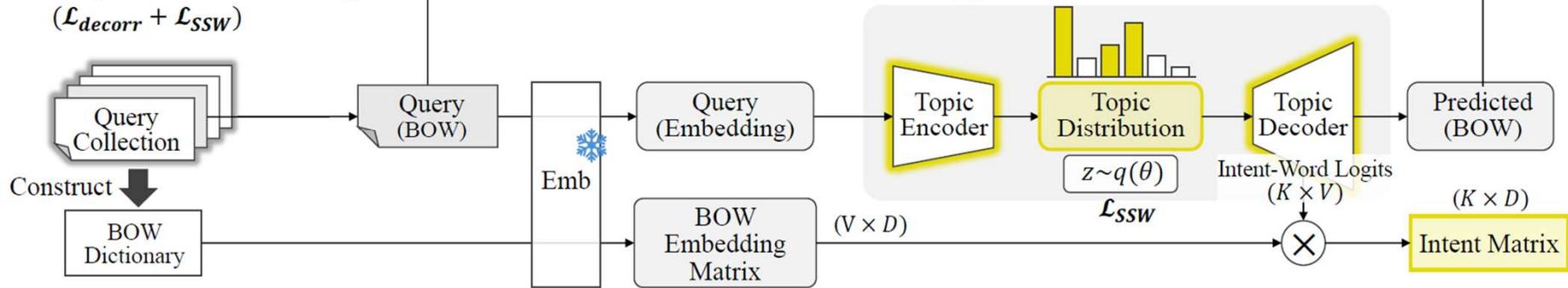
## 3. End-to-End Differentiable Optimization

$(\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo})$



## Topic-Adaptive Intent Learning

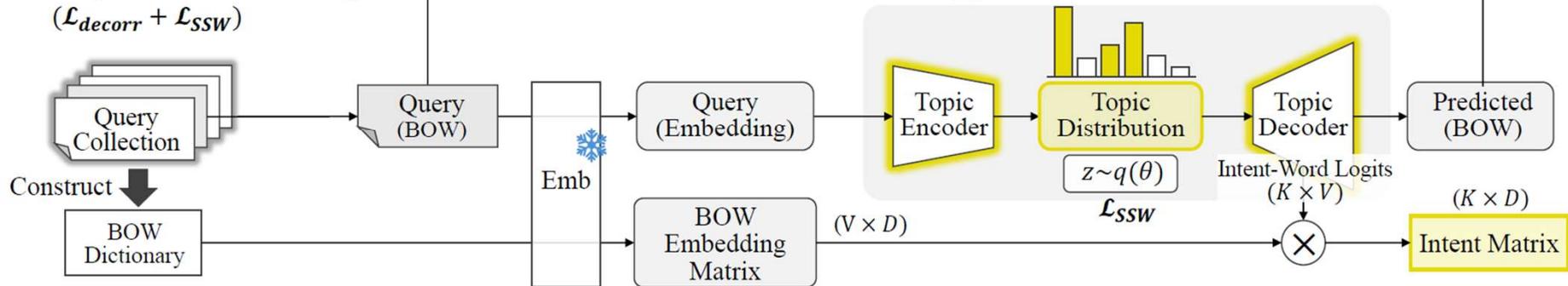
### 1. Topic-Adaptive Intent Learning



- 다양한 검색 측면을 포착하기 위해, 여러 질의에 공유되는 잠재 의도를 학습하는 적응형 의도 선택기 도입
  - ✓ 토픽 모델링을 통해 질의 간에 공유되는 잠재 의도(latent intents)를 학습하도록 설계
  - ✓ 의미적으로 유사한 질의들이 유사한 의도 분포(intent distribution)를 형성
- 질의 내 존재하는 다중 의도를 추출하여 다양한 측면의 검색을 유도

## Topic-Adaptive Intent Learning

### 1. Topic-Adaptive Intent Learning



- 학습 목표

- ✓ Reconstruction Loss  $\mathcal{L}_{rec}$

- Topic Encoder와 Decoder가 학습한  $\theta$ 가 BoW를 제대로 복원하는 지

- ✓ Spherical Sliced-Wasserstein Loss  $\mathcal{L}_{SSW}$

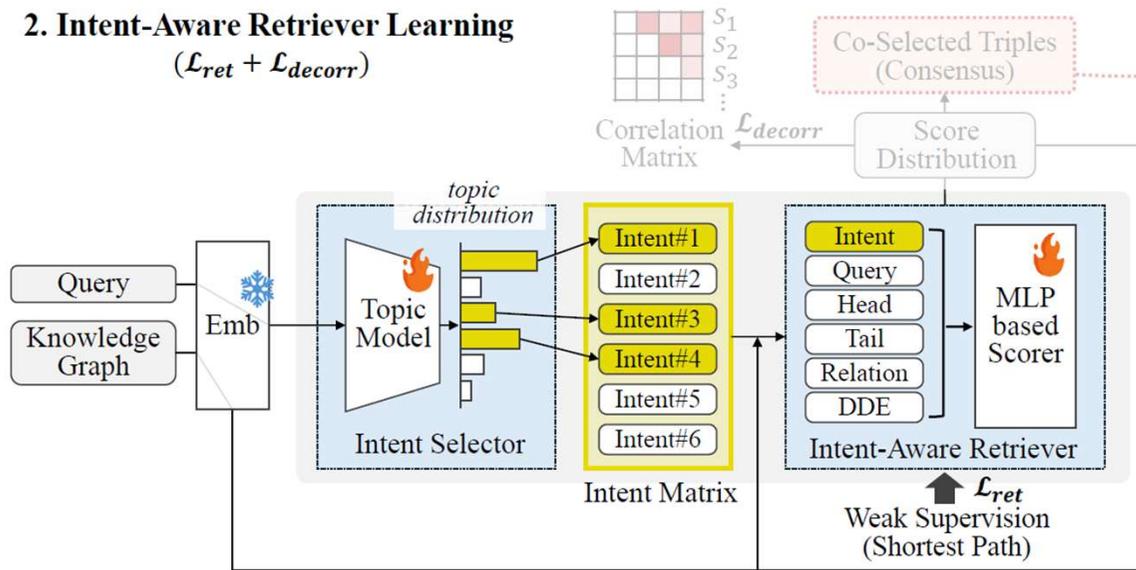
- Encoder가 만든  $z$ 와 구형 분포 사이의 구조적 정렬을 유지하도록 정규화
    - 잠재 벡터가 한쪽에 몰리지 않게 유도하여, 의미적으로 유사한 질의는 가까이, 다른 질의는 멀리 배치

- 최종 Loss

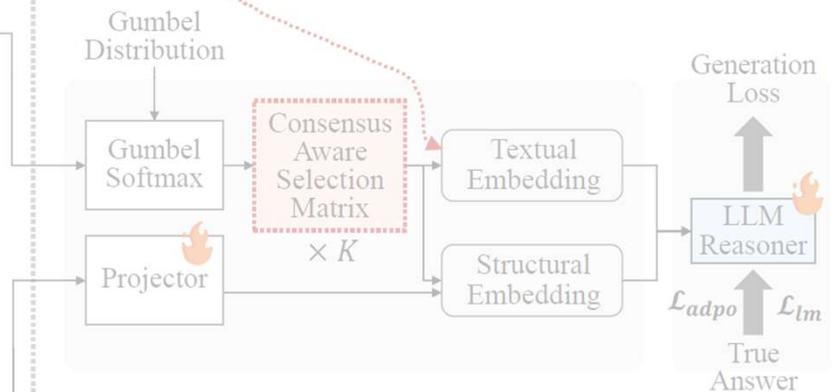
$$\mathcal{L}_{selector} = \mathcal{L}_{rec} + \mathcal{L}_{SSW}$$

## Intent-aware Retriever

### 2. Intent-Aware Retriever Learning ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr}$ )



### 3. End-to-End Differentiable Optimization ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo}$ )



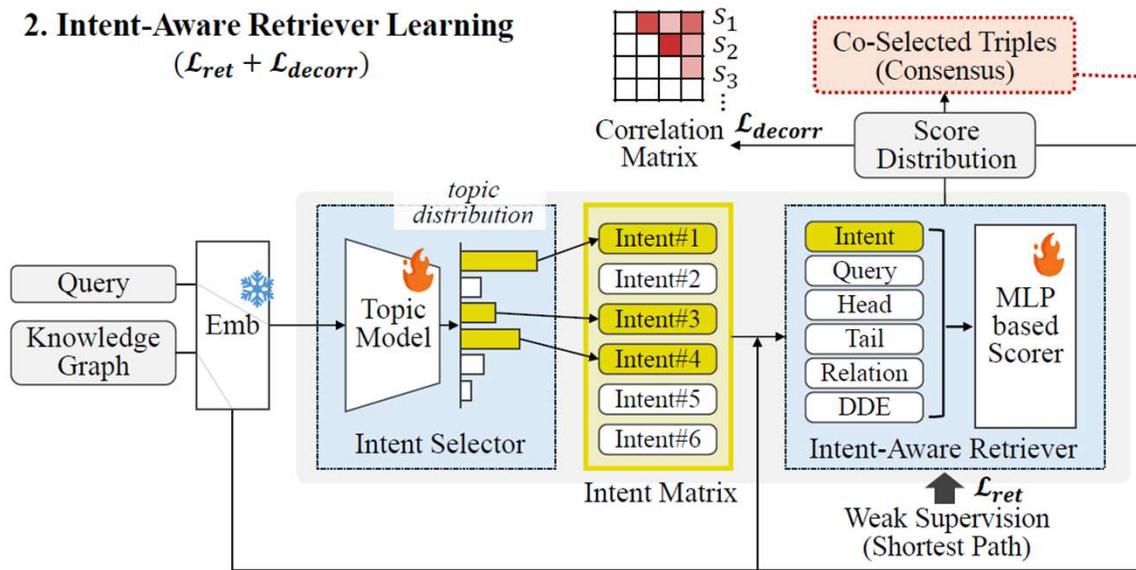
- Adaptive Intent Selector

- ✓ 토픽 분포를 높은 순으로 누적하며, 사전 정의된 임계값  $\gamma$  이하의 의도만 활성화
  - 질의별로 다른 수의 의도가 활성화

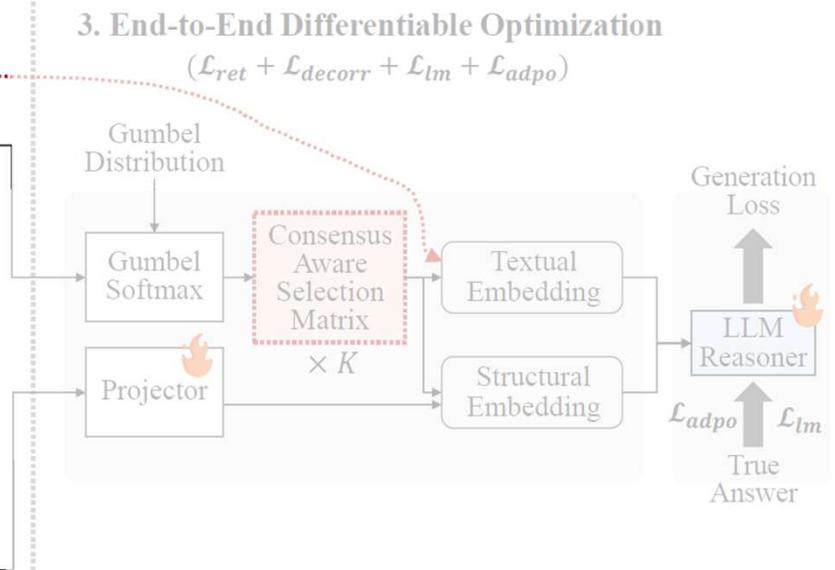
$$I(q) = \{k \mid \sum_{j \leq k} \theta_{(j)} \leq \gamma\}$$

## Intent-aware Retriever

### 2. Intent-Aware Retriever Learning ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr}$ )



### 3. End-to-End Differentiable Optimization ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo}$ )



- Intent-specific Triple Scoring

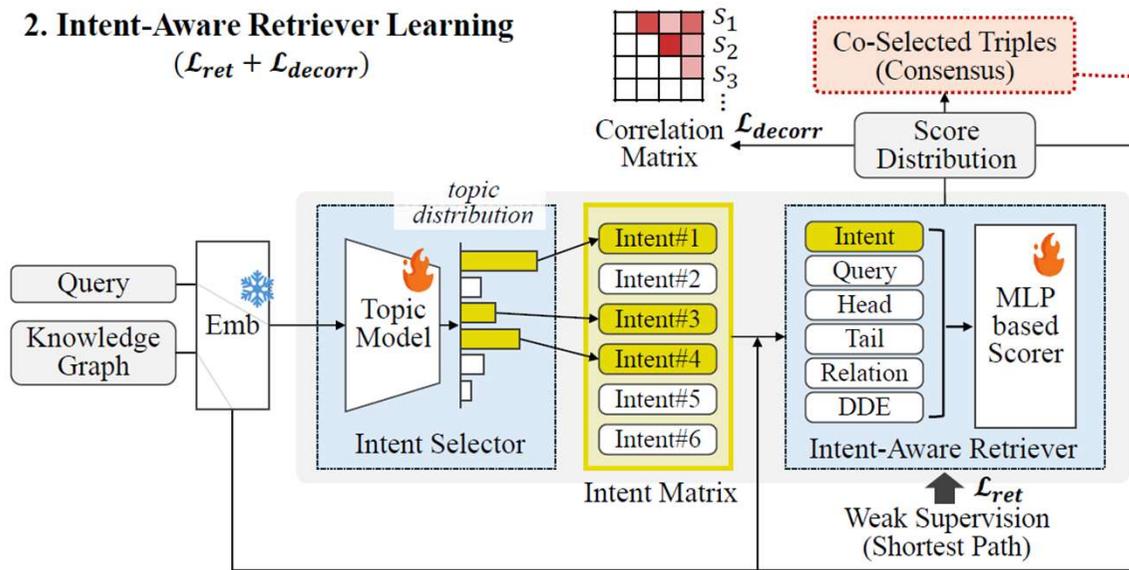
- ✓ 기존 SOTA 모델인 SubgraphRAG의 구조를 기반으로 의도를 반영
- ✓ 앞서 선택된 의도를 기반으로 한 질문에 대해 의도 별 점수를 계산

$$s_k(q, h, r, t) = f_{pred}([u_k || q || h || r || t || \phi_{DDE}])$$

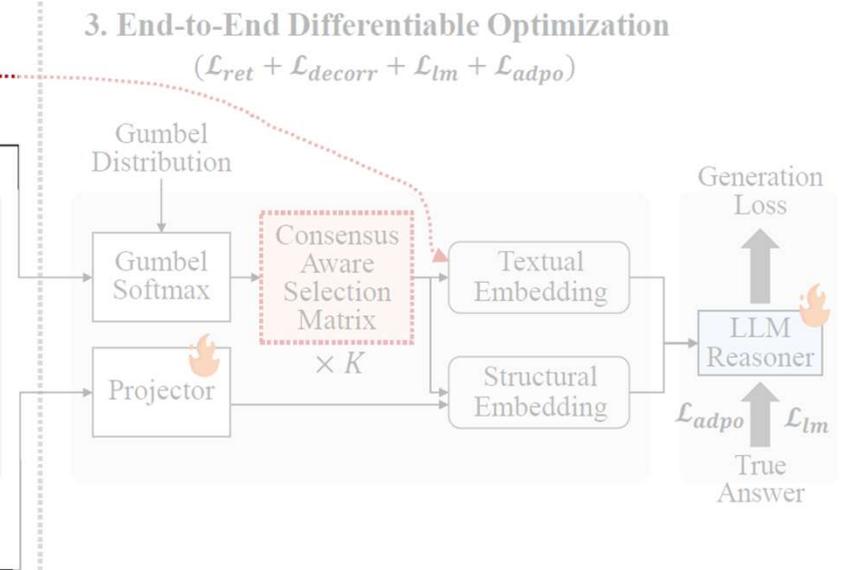
- 동일한 트리플이라도 의도마다 점수가 다를 수 있음

## Intent-aware Retriever

### 2. Intent-Aware Retriever Learning ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr}$ )



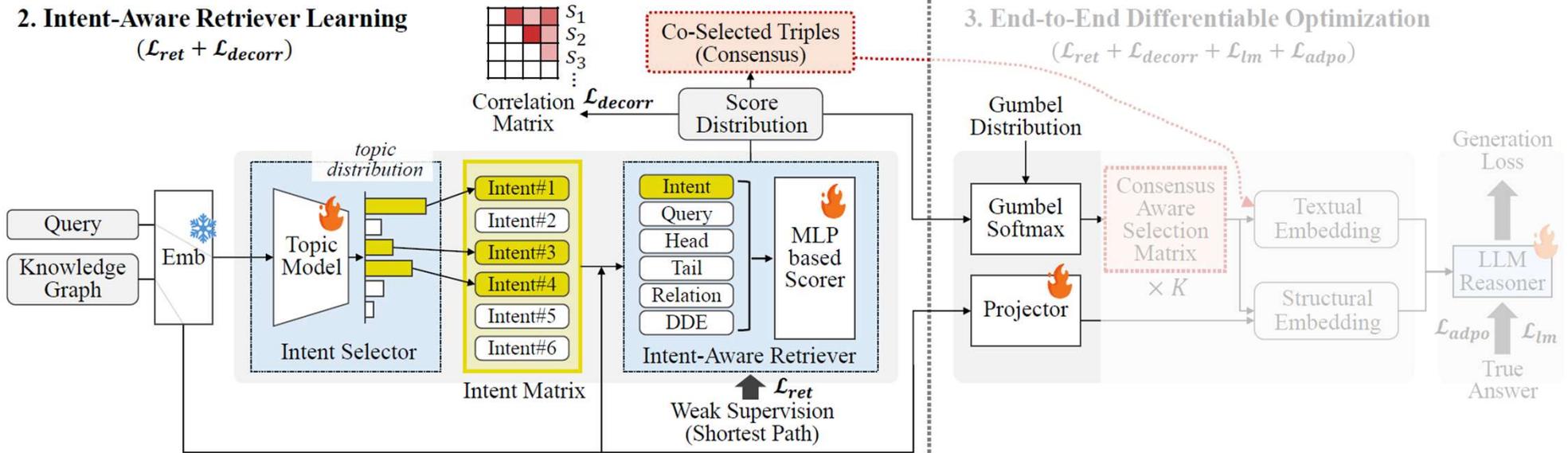
### 3. End-to-End Differentiable Optimization ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo}$ )



### • 학습 목표

- ✓ Retriever Loss  $\mathcal{L}_{ret}$ 
  - Ranking Loss  $\mathcal{L}_{rank}$
  - Binary Cross Entropy Loss  $\mathcal{L}_{bce}$
  
- ✓ Decorrelation Loss(비상관)  $\mathcal{L}_{decorr}$ 
  - 의도 간 점수 분포가 유사해지는 것을 방지하기 위해 적용

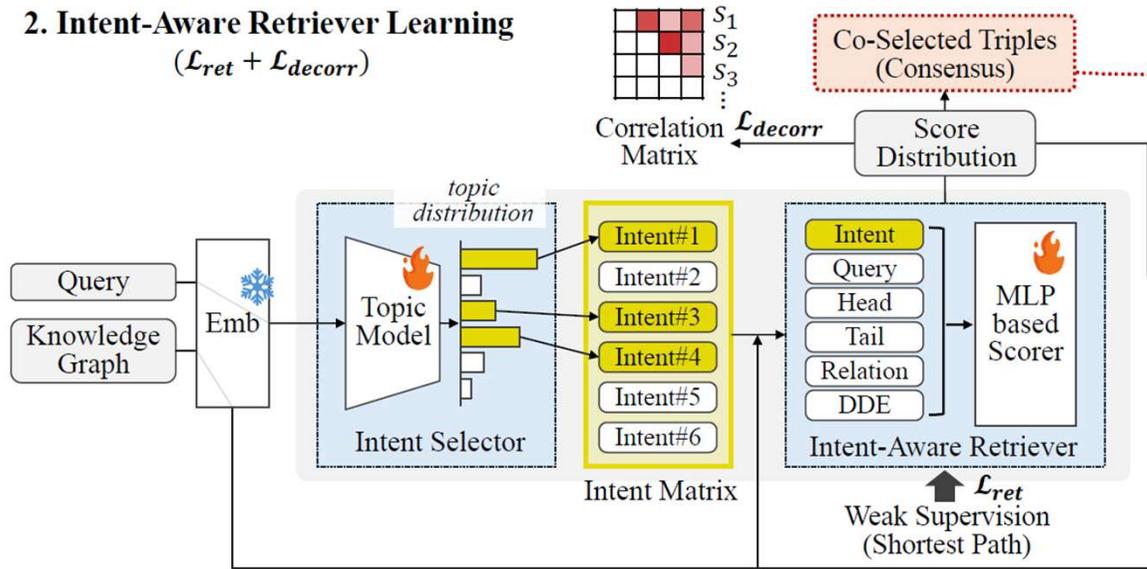
## Differentiable Triple Selection



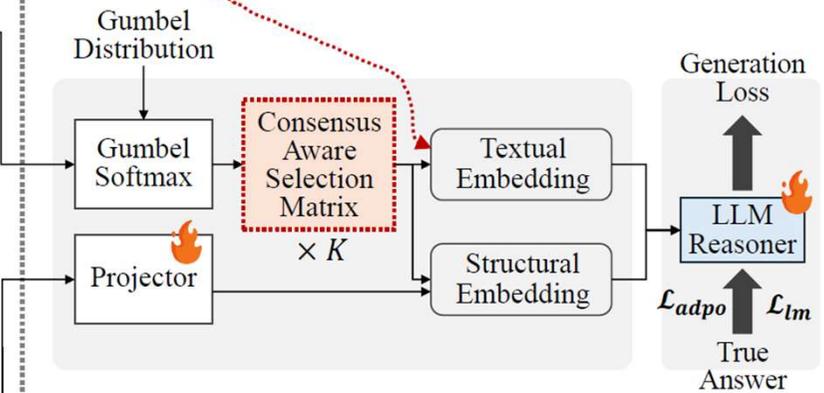
- 기존 Triple Selection은 이산적인 연산이기 때문에 미분 불가능
  - ✓ Gumbel softmax를 통해 이산적인 선택 과정을 확률적 형태로 근사하여 Gradient가 흐르게 만들어줌
  - ✓ 이를 통해 트리플 선택이 Differentiable sampling 형태로 변환되어 Retriever와 LLM 공동 최적화 가능
- 의도 간 학습 균형 및 계산 효율성을 위해 의도별 Top-k개의 트리플 검색
  - ✓ 트리플 개수를 다르게 둘 경우, 일부 의도의 영향력이 커지거나 무시될 수 있음

## Consensus-aware Prompt Construction

### 2. Intent-Aware Retriever Learning ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr}$ )



### 3. End-to-End Differentiable Optimization ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo}$ )



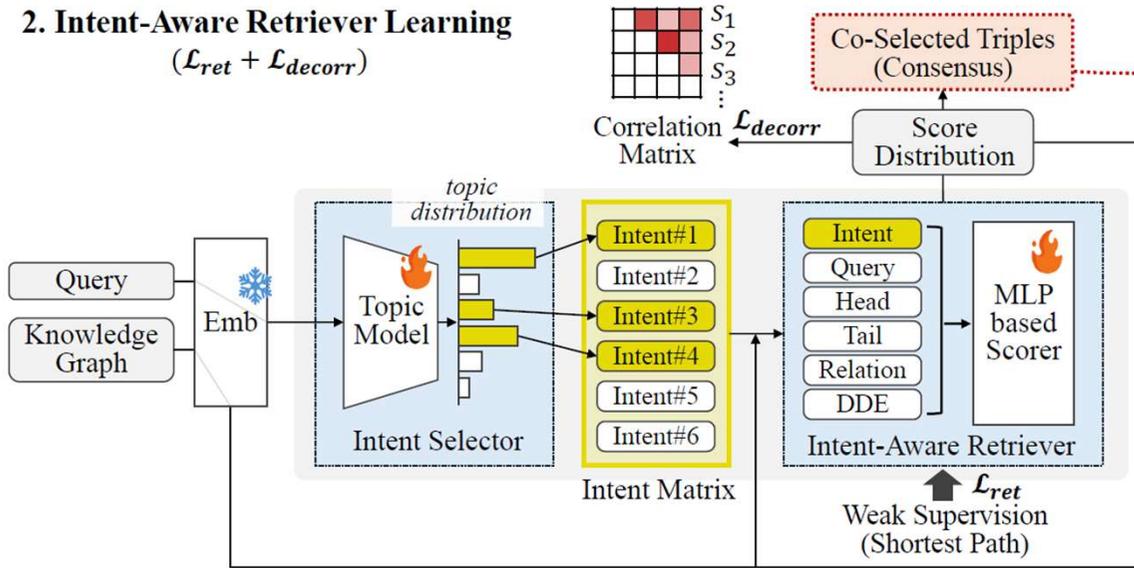
- 여러 의도가 공통적으로 선택한 트리플은 추론에 도움이 되는 증거일 가능성이 높음
  - ✓ 트리플 별로 의도들이 선택한 횟수와 confidence(신뢰도)를 계산하여 합의 정도를 반영한 마스크

$$c_t = \sum_{i=1}^I y_{t,i}, \quad g_t = 1 + \frac{1}{I} \sum_{i=1}^I y_{t,i}$$

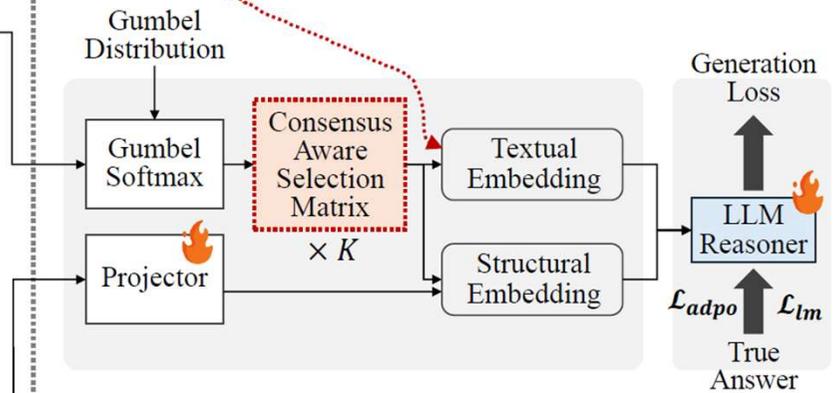
- ✓ 즉, Consensus가 높을수록 강조 (가중치 부여)

## Consensus-aware Prompt Construction

### 2. Intent-Aware Retriever Learning ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr}$ )



### 3. End-to-End Differentiable Optimization ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo}$ )

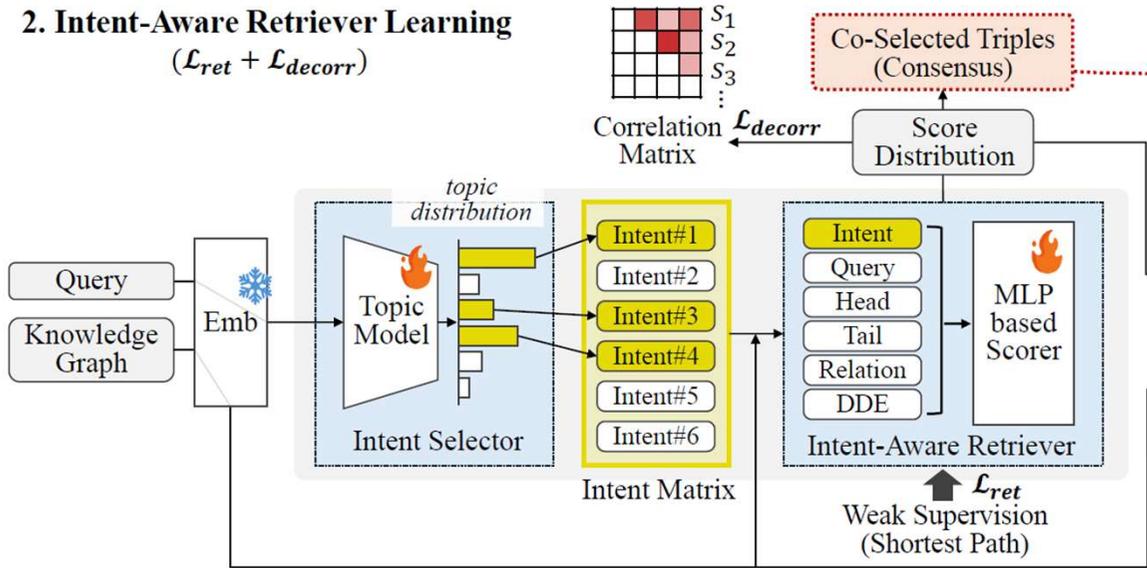


- 각 트리플은 구조적 (Structural), 의미적 (Semantic) 두 관점에서 표현
  - ✓ 구조적 정보는 MLP 기반 Projector를 통해 head-relation-tail 간의 관계 패턴을 인코딩
  - ✓ 의미적 정보는 triple을 자연어 토큰으로 변환한 뒤 LLM의 Embedding Layer를 통해 문맥적 의미를 반영

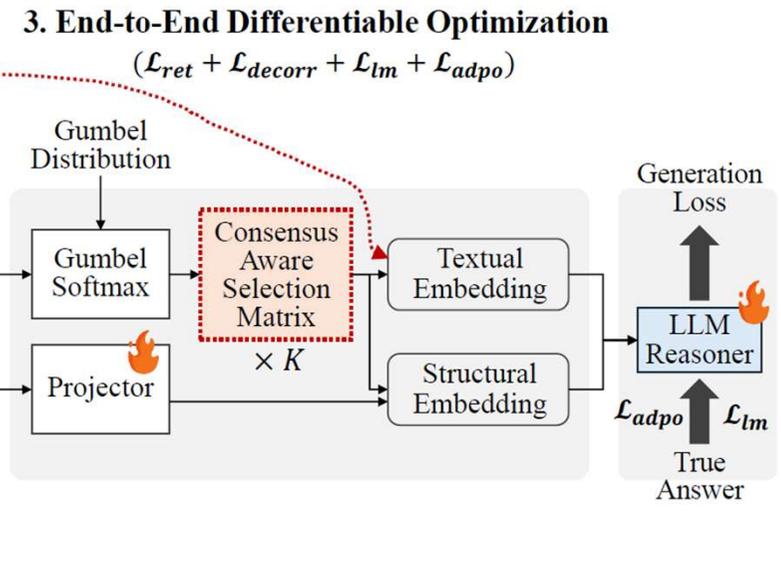
$$Struct(t) = MLP([h|r|t]), \quad Sem(t) = E_{LLM}(Tok(h,r,t))$$

## Consensus-aware Prompt Construction

### 2. Intent-Aware Retriever Learning ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr}$ )



### 3. End-to-End Differentiable Optimization ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo}$ )



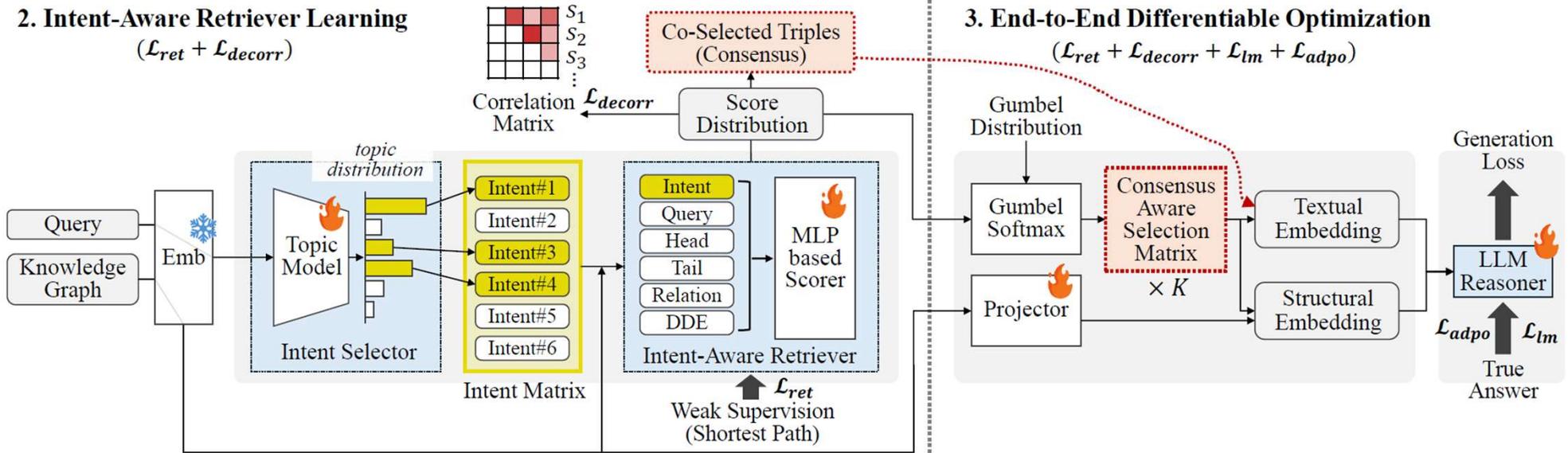
- 두 관점의 표현과 앞서 계산한 gate  $g_t$ 를 곱하여 triple 단위의 Triple block를 생성
  - ✓ Gate는 Triple selection을 미분가능하게 해줌으로, 두 표현에 곱해줌으로써 end-to-end 학습 가능

$$\hat{B}(t) = [g_t \cdot Struct(t) \parallel g_t \cdot Sem(t)]$$

- 최종 프롬프트

$$\mathcal{P}(q) = Concat(\{\hat{B}(t) \mid t \in \mathcal{T}(q), g_t > 0\})$$

## Adaptive DPO



- 모든 합의 증거가 실제 Reasoning 품질을 향상시키는 것은 아님
  - ✓ 오히려 Noise 유발 가능
    - Ex) 여러 의도에서 공통적으로 선택한 Triple이 단순 hub node인 경우
  - ✓ 즉, Consensus Frequency  $\neq$  Informativeness

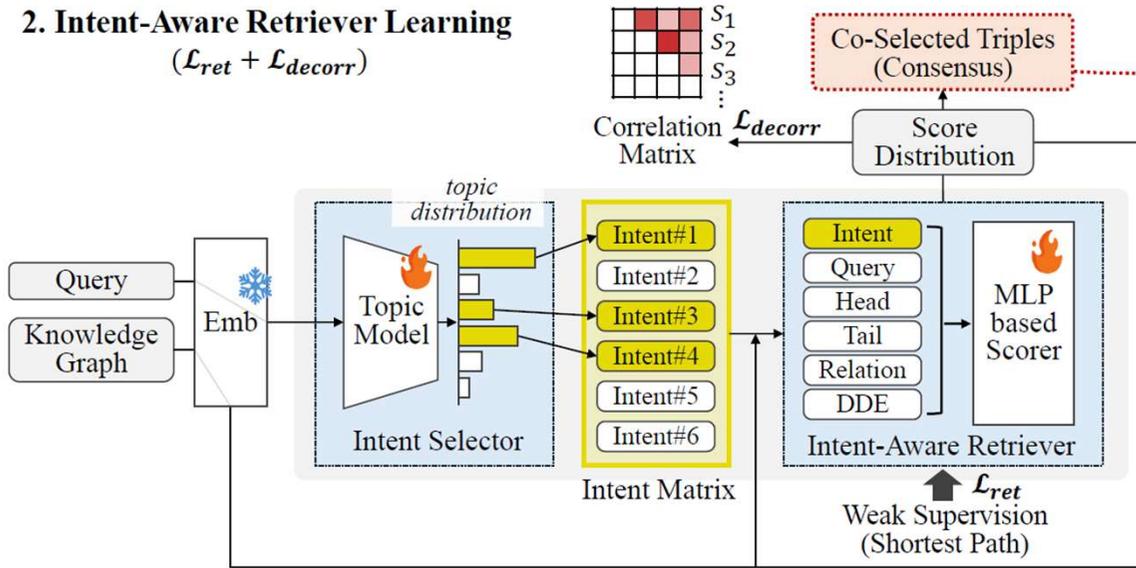
### • 학습 목표

- ✓ 합의 증거가 실제 Reasoning 품질을 높이는 경우에만 강화  $\Rightarrow$  Adaptive

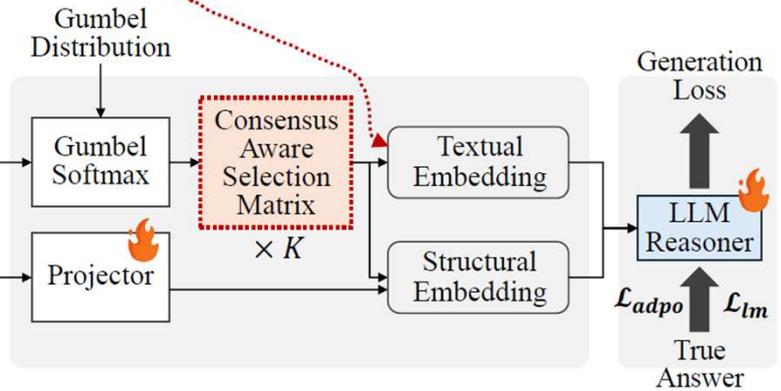
$$\mathcal{L}_{ADPO}(q) = -\log \sigma(\beta \cdot \Delta(q)) \cdot 1\{Pref(q) \neq None\}$$

## Adaptive DPO

### 2. Intent-Aware Retriever Learning ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr}$ )



### 3. End-to-End Differentiable Optimization ( $\mathcal{L}_{ret} + \mathcal{L}_{decorr} + \mathcal{L}_{lm} + \mathcal{L}_{adpo}$ )



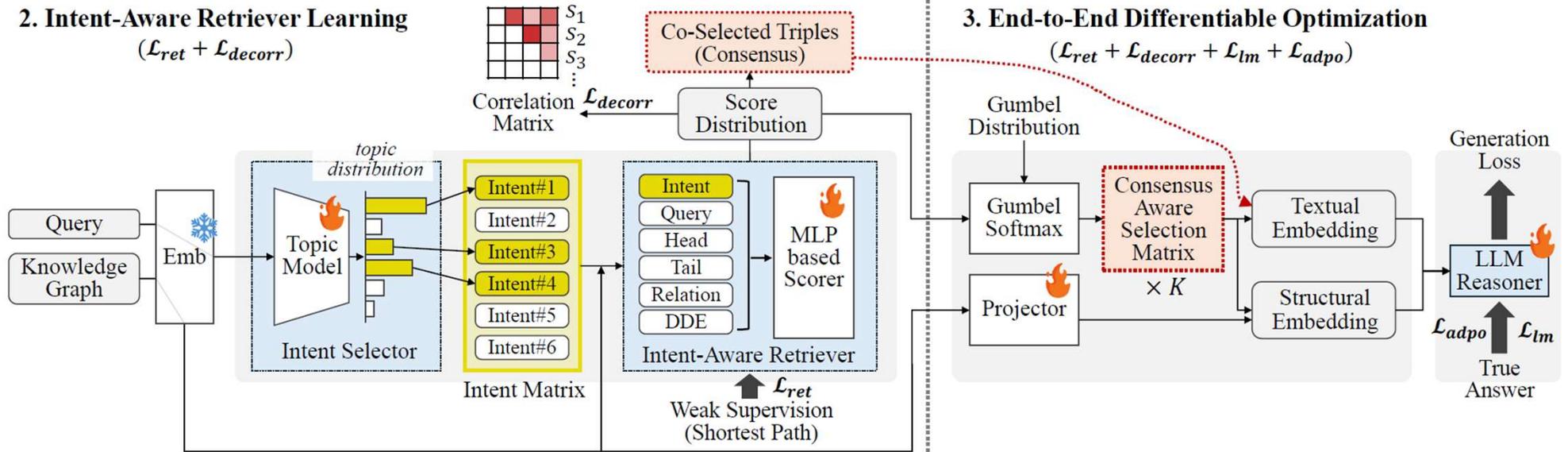
### • 학습 방법

- ✓ 두 가지 Prompt Context를 고려
  - Weighted Context ( $y^w$ ): 합의 증거가 강조된 프롬프트
  - Plain Context ( $y^p$ ): 합의 증거가 강조되지 않은 기본 프롬프트
- ✓ 두 Context 각각에서의 log-likelihood 차이를 통해 Preference 결정
 
$$\Delta(q) = \log \pi_{\theta}(y^w | q) - \log \pi_{\theta}(y^p | q)$$
- ✓  $\Delta(q)$  크기에 따라 어느 쪽이 더 나은지 판단 (불확실한 경우 업데이트 X)

$$Pref(q) = \begin{cases} y^w & \text{if } \Delta(q) > \delta, \\ y^p & \text{if } \Delta(q) < \delta, \\ None & \text{otherwise} \end{cases}$$

# Method

## Training Strategy



- Phase 1. Intent Selector, Intent-Aware Retriever Pretraining

$$\mathcal{L}_{selector} = \mathcal{L}_{rec} + \mathcal{L}_{SSW}$$

$$\mathcal{L}_{ret} = \mathcal{L}_{bce} + \mathcal{L}_{rank} + \mathcal{L}_{decorr}$$

- Phase 2. Joint training

$$\mathcal{L}_{joint} = \rho \cdot \mathcal{L}_{ret} + (1 - \rho) \cdot \mathcal{L}_{lm} + \rho_{decorr} \cdot \mathcal{L}_{decorr} + \rho_{adpo} \cdot \mathcal{L}_{ADPO}$$



**Thank You for Listening**