


# 나무위키 지식그래프 구축 및 question answering (v2)

👤 작성자	Ⓢ Sumin Seo
≡ 기수	
≡ 작성 팀	S6_깃헙에 NLP 잔디심기 2
≡ 자료 타입	
≡ 기술 분류	
≡ keywords	
🖼 Thumbnail	
☀ 상태	Not started
🕒 최종 편집 일시	@July 16, 2023 4:38 PM

레퍼런스 / 학습 자료

발표 자료

0. 프로젝트 소개

1. 데이터 전처리

1-1. 나무위키 데이터 구하기

1-2. text의 table에서 entity가 내용인 것만 필터링

1.3 table 데이터를 triple 데이터로 변환

1.4 triple 후처리

2. GraphDB 적재

3. KBQA (Knowledge Based Question Answering)

3.1 KBQA 방법론

3.2 KBQA와 생성모델 QA 답변 비교하기

3.3 Natural Language to GraphQuery with gpt

4. Future Work

작성자 소개

Q&A

댓글

## 레퍼런스 / 학습 자료

•

## 발표 자료

## 0. 프로젝트 소개

- 나무위키로 한국어 지식그래프 만들고, 지식그래프 기반 question answering 해보기
  - 유명한 지식그래프 목록
    - wikidata: [https://www.wikidata.org/wiki/Wikidata:Main\\_Page](https://www.wikidata.org/wiki/Wikidata:Main_Page)
    - dbpedia: [https://dbpedia.org/page/The\\_Lord\\_of\\_the\\_Rings](https://dbpedia.org/page/The_Lord_of_the_Rings)
    - YAGO: <https://yago-knowledge.org/>
- question answering 결과를 gpt 결과와 비교해보기 (정확성 측면 질적 비교)
- graphQL (SPARQL)을 gpt로 생성해보기

## 1. 데이터 전처리

### 1-1. 나무위키 데이터 구하기

heegyu/namuwiki at main

We're on a journey to advance and democratize artificial intelligence through open source and open science.

🤖 <https://huggingface.co/datasets/heegyu/namuwiki/tree/main>

heegyu  
/namuwiki

🤖 huggingface.co

- 나무위키는 20210301까지 dump 데이터를 제공한다.
- 이 dump 데이터를 parquet으로 가공해놓은 데이터를 찾았다.
- 데이터는 title(나무위키 제목)과 text(나무위키 본문 html)로 이루어져 있다

	title	text
0	!	#redirect 느낌표\n
1	!!아앗!!	\n[목차]\n\n''''{{+1 ! ! ああっと ! ! }}'''\n\n== 개요 ==\n...
2	"....."	<-2><tablebordercolor=#878787><tablealign=ri...
3	#	[[분류:특수 문자]]\n\n[include(틀:다른 뜻1, other1=음악에서 사용...
4	#FairyJoke	[include(틀:링크시 주의, 링크=\\#FairyJoke]] 또는 [[#F...
...	...	...
867019	광저우 푸리	#redirect 광저우 시티\n
867020	에스시더	#redirect 이스시\n
867021	칭다오 황하이	#redirect 칭다오 FC\n
867022	순혜옹주	[[분류:조선의 후궁]]\n\n[include(틀:조선의 후궁(환조 ~ 연산군))]\n...
867023	허난 젠예	#redirect 허난 쑹산 룽먼\n

867024 rows x 2 columns

- 나무위키의 한 페이지가 한 개의 엔티티에 대응한다. 나무위키의 url은 '[https://namu.wiki/w/Stray\\_Kids](https://namu.wiki/w/Stray_Kids)'와 같은 형태이고, /의 맨 뒤에 있는 부분이 title이다. 이 url을 엔티티의 아이디로 사용한다.
- 나무위키의 본문은 table 부분과 설명 부분으로 나누어져 있다. 이 중 table 정보에서 triple을 추출하여 지식그래프를 만든다.
  - 테이블 영역

## 목차

1. 개요
2. 멤버
  - 2.1. 멤버 간 케미
3. 특징
  - 3.1. 그룹명
  - 3.2. 자체 프로듀싱
  - 3.3. 데뷔 배경
  - 3.4. 음악 스타일
  - 3.5. 로고
  - 3.6. 인기 및 기록
4. 음반
  - 4.1. 뮤직 비디오
  - 4.2. 노래방 수록곡
5. 수상
6. 활동
  - 6.1. 연도별 활동 내역
  - 6.2. 음악 방송 직캠
  - 6.3. 콘텐츠
    - 6.3.1. CH.SKZ (채널 스키즈)
    - 6.3.2. LIVE
  - 6.4. 공연 및 행사
  - 6.5. 광고
  - 6.6. 홍보대사
7. 팬덤
  - 7.1. 응원법
  - 7.2. 굿즈
  - 7.3. 언어유희 및 레퍼런스
8. 여담
9. 논란 및 사건 사고
10. 역대 프로필 사진

 <b>Stray Kids</b> 스트레이 키즈	
그룹명	[ 펼치기 · 접기 ]
결성일	2017년 12월 19일 <sup>[2]</sup> (결성일로부터 +2010일, 5주년)
데뷔일	 2018년 3월 25일 <sup>[3]</sup> (데뷔일로부터 +1914일, 5주년)
	 2020년 3월 18일 (데뷔일로부터 +1190일, 3주년)
데뷔 음반	 미니 1집 <b>I am NOT</b>
	 베스트 1집 <b>SKZ2020</b>
최신 음반	 정규 3집 <b>★★★★★</b>
	 정규 1집 <b>THE SOUND</b>
장르	댄스, 랩/힙합, 일렉트로니카, 얼터너티브, R&B, 발라드, 어쿠스틱, 락, 믹스팝
리더	방찬
프로듀서	<b>3RACHA</b>
소속사	<b>JYP</b> LEADER IN ENTERTAINMENT
에이전시	 <b>republic</b>
유통사	 <b>Dreamus</b>
	 <b>Sony Music</b>
	 <b>UNIVERSAL</b> UNIVERSAL MUSIC GROUP
슬로건	Stray Kids Everywhere All Around the World <sup>[4]</sup> You Make Stray Kids Stay <sup>[5]</sup>
팬덤	
응원봉	 <b>나침봉</b>
링크	[ 펼치기 · 접기 ]

## ◦ 설명 영역

### 3. 특징

[편집]

3RACHA의 랩/힙합을 기반으로 한 프로듀싱과 보컬라인<sup>[8]</sup>의 서정적인 보컬, 댄스라인<sup>[9]</sup>을 중심으로 한 다이나믹한 퍼포먼스가 합쳐져 8명이 유기적인 조화를 이루는 것이 이 그룹의 특징이다.

데뷔 때부터 한 걸음씩 점진적으로 상승세를 보이고 있는 성장형 아이돌이다. 쥬대 있게 데뷔부터 현재까지 오직 '스트레이 키즈 고유 스타일'의 전곡 자작곡을 선보이며, **킹덤: 레전더리** 위에서 우승하고, 2022년에는 빌보드 200 1위를 2번 거머쥐고 있다.<sup>[10]</sup> 단연 4세대 대표 남자 아이돌로 손꼽히고 있다.<sup>[11]</sup>

#### 3.1. 그룹명

[편집]

스트레이 키즈(Stray Kids)는 '방황하는 아이들'이라는 뜻으로, 기존의 틀에서 벗어나 8명이 뚫을 뚫어 자신의 길을 스스로 찾아 나가겠다는 포부를 담고 있다.

길 잃은 잠민이라는 말도 있다...

#### 3.2. 자체 프로듀싱

[편집]

그룹 내 프로듀싱팀 3RACHA가 앨범 전곡을 작사, 작곡<sup>[12]</sup>하는 자체 프로듀싱 그룹이다. 현재<sup>[13]</sup>까지 3RACHA 멤버가 작사에 참여하지 않은 곡은 없으며 작곡 미참여 곡은 'Booster', 'Phobia', '청사진', 'Star Lost'로 단 4곡 뿐이다.<sup>[14]</sup>

이와 관련해 박진영이 참여한 Stray Kids 곡은 2023년 5월 기준 3곡이며, 이 중 작곡에 직접 참여한 곡은 단 1곡이다. 미니 2집의 타이틀 곡 **My Pace**와 미니 5집의 타이틀 곡 **바람 (Levanter)**에 작사 참여, 일본 미니 1집의 타이틀 곡 **ALL IN**에 작사, 작곡, 편곡 모두 참여했다.

다른 멤버들의 수록곡 작사·작곡 참여도도 높은 편이다. **I am NOT, I am WHO, I am YOU, Clé 1 : MIROH, Clé : LEVANTER**에는 멤버 전원 작사·작곡에 참여한 곡을 Mixtape이라는 이름으로 각 앨범마다 수록하였으며<sup>[15]</sup> **IN生, NOEASY, ODDINARY, MAXIDENT**에는 유닛별로 작사·작곡한 곡이 수록되어있다. 자세한 내용은 멤버별 음악 활동 문서의 작사 및 작곡 문단 참조.

## 1-2. text의 table에서 entity가 내용인 것만 필터링

- table 영역은 내용으로 다른 엔티티(=나무위키 페이지)를 가지는 경우와 literal을 가지는 경우가 있다.
- 위의 예시에서 리더, 소속사는 엔티티를 내용으로 가지고, 장르의 믹스팝이나 슬로건은 literal이다. (나무위키 링크가 걸려있지 않아서)
- raw text로 보면 [[방찬]], [[춤|댄스]]과 같이 이중 꺾쇠로 되어있는 것은 엔티티이다. 이 경우 나무위키 url prefix인 <https://namu.wiki/w/>에 title을 더해서 <https://namu.wiki/w/방찬>, <https://namu.wiki/w/춤> 과 같이 url을 만들면 나무위키 페이지랑 연결된다
- 이중꺾쇠 없이 그냥 글자가 있으면 나무위키 페이지가 없는 것이므로 엔티티로 취급하지 않는다. 사람 이름, 회사명이라든가 나무위키에 없으면 literal로 본다.

```
||<bgcolor=#ca9a6d> {{{#06172b \'\''리더\''\''}}} ||<-2> \'\''[[방찬]]\''\'' ||\n
||<bgcolor=#ca9a6d> {{{#06172b \'\''장르\''\''}}} ||<-2> [[춤|댄스]], [[랩(음악)|랩]], [[힙합]], 믹스팝 ||
||<bgcolor=#ca9a6d> {{{#06172b \'\''슬로건\''\''}}} ||<-2> Stray Kids Everywhere All Around the World ||
```

- 간단하게 하기 위해 literal은 사용하지 않고, subject-predicate-object 관계에서 subject, object가 둘 다 엔티티인(나무위키 페이지) 것만 사용해서 트리플을 만든다.

### 1.3 table 데이터를 triple 데이터로 변환

- 예를 들어, 스트레이키즈 페이지의 table에서 엔티티가 들어간 항목만 뽑았다면, 해당 페이지 title이 subject, table 필드가 predicate, table의 값이 object가 된다.

```

||<bgcolor=#ca9a6d> {{#06172b \'\'\'리더\'\'\'}} ||<-2> \'\'\'[[방찬]]\'\'\' ||\n
-> https://namu.wiki/w/Stray Kids \t 리더 \t https://namu.wiki/w/방찬

||<bgcolor=#ca9a6d> {{#06172b \'\'\'장르\'\'\'}} ||<-2> [[춤|댄스]], [[랩(음악)|랩]], [[힙합]], 믹스팝 ||
-> https://namu.wiki/w/Stray Kids \t 장르 \t https://namu.wiki/w/춤
-> https://namu.wiki/w/Stray Kids \t 장르 \t https://namu.wiki/w/랩(음악)
-> https://namu.wiki/w/Stray Kids \t 장르 \t https://namu.wiki/w/힙합

```

- 최종적으로 아래와 같은 rdf 데이터를 만든다.
  - RDF(Resource Description Framework)는 **자원의 정보를 표현하고 상호 연결 및 교환하기 위하여 월드 와이드 웹 컨소시움(W3C)에 의해 개발되고 있는 언어 규격**이다.
  - 지식그래프는 기본적으로 웹페이지간의 연결 관계를 표현하기 위한 것이고, 이걸 어떻게 잘 정리해서 표현해야되는지에 대한 표준이 자세한 규칙으로 정의되어 있다.(<https://www.w3.org/TR/rdf-syntax-grammar/>)
  - 예를 들어 엔티티는 url 형태로 표현되어야 한다는 것도 rdf 표준이다.
  - rdf 외에도 n-triple, owl, json-ld 등 다양한 linked data에 대한 표준 데이터 포맷이 있으며 이 포맷을 맞추어 주어야 graphDB에 올릴 수 있다.

	subject	predicate	object
0	<https://namu.wiki/w/.....>	<https://predicate/시작>	<https://namu.wiki/w/2013년>
1	<https://namu.wiki/w/.....>	<https://predicate/시작>	<https://namu.wiki/w/5월_26일>
2	<https://namu.wiki/w/.....>	<https://predicate/종료>	<https://namu.wiki/w/2013년>
3	<https://namu.wiki/w/.....>	<https://predicate/종료>	<https://namu.wiki/w/10월_13일>
4	<https://namu.wiki/w/.....>	<https://predicate/출연진>	<https://namu.wiki/w/유민상>
...	...	...	...
1153325	<https://namu.wiki/w/아즈사_아이자와>	<https://predicate/성우>	<https://namu.wiki/w/타나_데그레차프>
1153326	<https://namu.wiki/w/아즈사_아이자와>	<https://predicate/성우>	<https://namu.wiki/w/거미코>
1153327	<https://namu.wiki/w/순혜응주>	<https://predicate/사망>	<https://namu.wiki/w/1423년>
1153328	<https://namu.wiki/w/순혜응주>	<https://predicate/사망>	<https://namu.wiki/w/세종(조선)>
1153329	<https://namu.wiki/w/순혜응주>	<https://predicate/사망>	<https://namu.wiki/w/7월_26일>

### 1.4 triple 후처리

- predicate: 200번 이상 쓰인 것만 남기기
- rdf 규칙에서 허용하지 않는 문자(% 등)가 들어간 엔티티 날리기 (귀찮아서..)
- 최종 결과

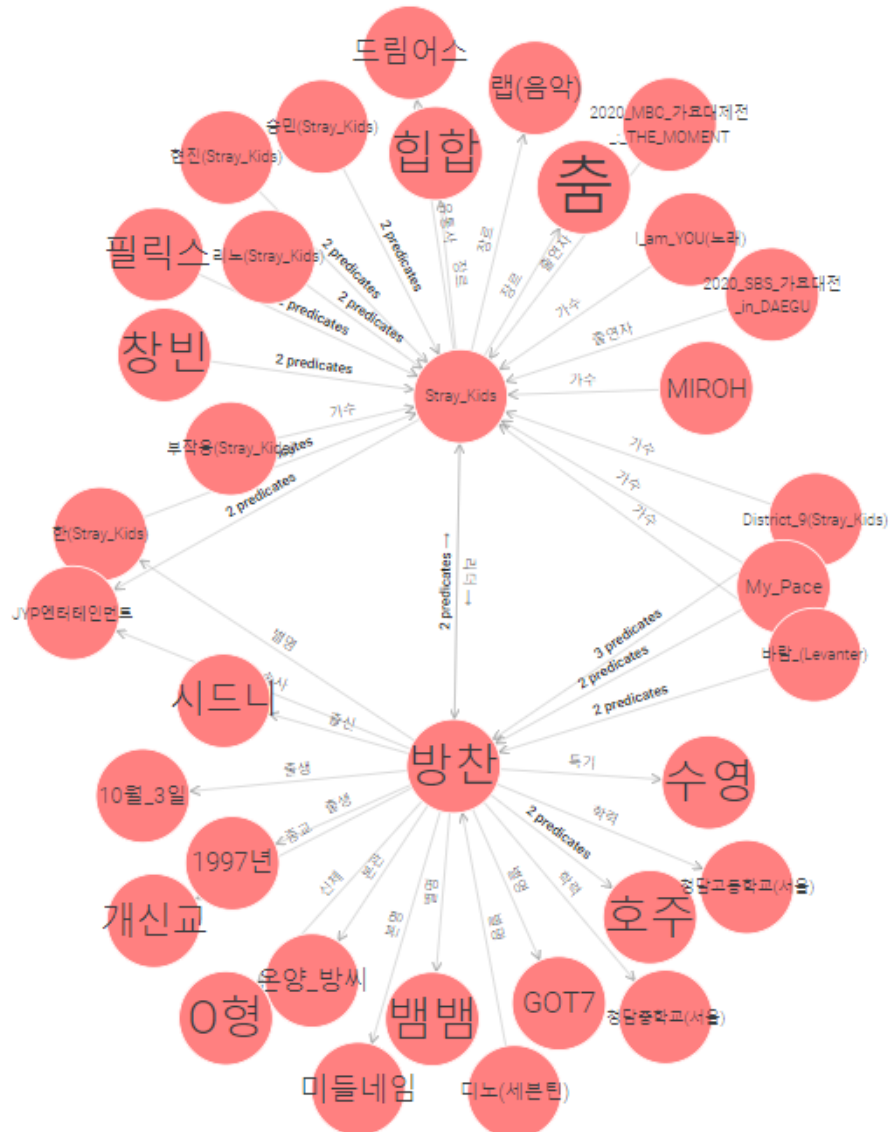
- triple: 1,029,179
- entity: subject 기준 125,632개, object에만 쓰인 엔티티까지 포함하면 280,257개
- predicate: 481개
- predicate TOP 30 : 아마 사람에 대한 정보가 가장 많은 듯

	<b>predicate</b>	<b>cnt</b>
0	<https://predicate/출생>	74207
1	<https://predicate/학력>	46891
2	<https://predicate/소속>	40076
3	<https://predicate/링크>	38388
4	<https://predicate/국적>	36197
5	<https://predicate/장르>	31290
6	<https://predicate/생년월일>	30797
7	<https://predicate/직업>	20326
8	<https://predicate/소속팀>	17995
9	<https://predicate/출신지>	17396
10	<https://predicate/포지션>	16805
11	<https://predicate/출연>	14827
12	<https://predicate/개봉일>	14182
13	<https://predicate/성우>	14053
14	<https://predicate/사망>	13727
15	<https://predicate/데뷔>	13035
16	<https://predicate/방송>	11890
17	<https://predicate/플랫폼>	11599
18	<https://predicate/감독>	11404
19	<https://predicate/주소>	11257
20	<https://predicate/신체>	11235
21	<https://predicate/종교>	11201
22	<https://predicate/경력>	10351
23	<https://predicate/소재지>	10190
24	<https://predicate/관련>	9271
25	<https://predicate/홈페이지>	8834
26	<https://predicate/발매일>	7621
27	<https://predicate/주요>	7602

28	< <a href="https://predicate/연재처">https://predicate/연재처</a> >	7376
29	< <a href="https://predicate/소속사">https://predicate/소속사</a> >	6793

## 2. GraphDB 적재

- graph structured query로 가장 전통적인 방법의 KBQA를 하기 위해서 나무위키 rdf를 GraphDB라는 그래프디비에 올렸다.





### 3. KBQA (Knowledge Based Question Answering)

#### 3.1 KBQA 방법론

- 가장 기본적/전통적인 KBQA 방법론으로 두 가지가 있다.
- Semantic Parsing Method**: 자연어 → 그래프쿼리(SPARQL, Cypher)로 변환(규칙기반, 템플릿, seq2seq generation 등) → 그래프에 쿼리 날리기
  - interpretable한게 가장 큰 장점. 그런데 자연어를 그래프쿼리로 바꾸는게 병목
- Information Retrivel Method**: 자연어 질문과 지식그래프 엔티티를 모두 임베딩 하고, 자연어 질문과 가장 가까운 지식그래프 엔티티를 n개 찾아서 reranking 하기
  - end-to-end training에 적합. 답변 도출 과정에 대한 설명력은 SP방법보다는 아무래도 낮음.
- gpt활용해서 sp방식 실험해보고자 하는게 프로젝트의 목적이었음.

#### 3.2 KBQA와 생성모델 QA 답변 비교하기

- 질문: 스트레이키즈 멤버의 국적과 출신 학교를 알려줘

1. sparql 쿼리 : 나무위키 기반 사실 정보가 나온다

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?소속?국적?학력
WHERE {
    ?소속 <https://predicate/학력> ?학력 .
    ?소속 <https://predicate/국적> ?국적 .
    ?소속 <https://predicate/소속> ?그룹 .
    ?그룹 <rdf:label> "스트레이키즈" .
}
```

	소속	국적	학력
1	<a href="https://namu.wiki/w/방찬">https://namu.wiki/w/방찬</a>	<a href="https://namu.wiki/w/호주">https://namu.wiki/w/호주</a>	<a href="https://namu.wiki/w/정당중학교(서울)">https://namu.wiki/w/정당중학교(서울)</a>
2	<a href="https://namu.wiki/w/방찬">https://namu.wiki/w/방찬</a>	<a href="https://namu.wiki/w/호주">https://namu.wiki/w/호주</a>	<a href="https://namu.wiki/w/정당중학교(서울)">https://namu.wiki/w/정당중학교(서울)</a>
3	<a href="https://namu.wiki/w/방찬">https://namu.wiki/w/방찬</a>	<a href="https://namu.wiki/w/호주_여권">https://namu.wiki/w/호주_여권</a>	<a href="https://namu.wiki/w/정당중학교(서울)">https://namu.wiki/w/정당중학교(서울)</a>
4	<a href="https://namu.wiki/w/방찬">https://namu.wiki/w/방찬</a>	<a href="https://namu.wiki/w/호주_여권">https://namu.wiki/w/호주_여권</a>	<a href="https://namu.wiki/w/정당중학교(서울)">https://namu.wiki/w/정당중학교(서울)</a>
5	<a href="https://namu.wiki/w/현진(Stray_Kids)">https://namu.wiki/w/현진(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/글로벌사이버대학교">https://namu.wiki/w/글로벌사이버대학교</a>
6	<a href="https://namu.wiki/w/현진(Stray_Kids)">https://namu.wiki/w/현진(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/서울공연예술고등학교">https://namu.wiki/w/서울공연예술고등학교</a>
7	<a href="https://namu.wiki/w/현진(Stray_Kids)">https://namu.wiki/w/현진(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/동북고등학교">https://namu.wiki/w/동북고등학교</a>
8	<a href="https://namu.wiki/w/현진(Stray_Kids)">https://namu.wiki/w/현진(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/서울성일초등학교">https://namu.wiki/w/서울성일초등학교</a>
9	<a href="https://namu.wiki/w/현진(Stray_Kids)">https://namu.wiki/w/현진(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/성나중학교(서울)">https://namu.wiki/w/성나중학교(서울)</a>
10	<a href="https://namu.wiki/w/린노(Stray_Kids)">https://namu.wiki/w/린노(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/김포제일공업고등학교">https://namu.wiki/w/김포제일공업고등학교</a>
11	<a href="https://namu.wiki/w/승민(Stray_Kids)">https://namu.wiki/w/승민(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/서울동은초등학교">https://namu.wiki/w/서울동은초등학교</a>
12	<a href="https://namu.wiki/w/승민(Stray_Kids)">https://namu.wiki/w/승민(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/동은중학교">https://namu.wiki/w/동은중학교</a>
13	<a href="https://namu.wiki/w/승민(Stray_Kids)">https://namu.wiki/w/승민(Stray_Kids)</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/정당고등학교(서울)">https://namu.wiki/w/정당고등학교(서울)</a>
14	<a href="https://namu.wiki/w/장빈">https://namu.wiki/w/장빈</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/보라고등학교">https://namu.wiki/w/보라고등학교</a>
15	<a href="https://namu.wiki/w/아이옌">https://namu.wiki/w/아이옌</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/서울공연예술고등학교">https://namu.wiki/w/서울공연예술고등학교</a>
16	<a href="https://namu.wiki/w/아이옌">https://namu.wiki/w/아이옌</a>	<a href="https://namu.wiki/w/대한민국">https://namu.wiki/w/대한민국</a>	<a href="https://namu.wiki/w/영로고등학교">https://namu.wiki/w/영로고등학교</a>

## 2. chatGPT

- 국적은 맞는데 출신학교가 엉망진창임(다 틀린 수준)



스트레이 키즈 멤버들의 국적과 출신 학교 정보는 다음과 같습니다:

1. 방찬 (Bang Chan):
  - 국적: 오스트레일리아
  - 출신 학교: Newtown High School of the Performing Arts (뉴타운 고등예술학교)
2. 리노 (Lee Know):
  - 국적: 대한민국
  - 출신 학교: 서울공연예술고등학교
3. 창빈 (Changbin):
  - 국적: 대한민국
  - 출신 학교: 서울공연예술고등학교
4. 현진 (Hyunjin):
  - 국적: 대한민국
  - 출신 학교: 서울공연예술고등학교
5. 한 (Han):
  - 국적: 대한민국
  - 출신 학교: 연세대학교 국제캠퍼스 부속 외국인학교
6. 필릭스 (Felix):
  - 국적: 오스트레일리아
  - 출신 학교: St. Patrick's Marist College (세인트 패트릭 마리스트 고등학교)
7. 승민 (Seungmin):
  - 국적: 대한민국
  - 출신 학교: 신도중학교
8. 아이엔 (I.N):
  - 국적: 대한민국
  - 출신 학교: 해양고등학교

이는 2021년 9월까지의 정보이며, 멤버들의 출신 학교는 변경될 수 있습니다.

## 3. Bing

스트레이키즈 멤버의 국적과 출신 학교를 알려줘

✓ "스트레이키즈 멤버 국적 출신 학교" 검색 중

✓ "스트레이키즈 멤버 국적 출신 학교" 검색 중

✓ 답변을 생성하는 중...

스트레이 키즈는 JYP 엔터테인먼트 소속의 8인조 보이그룹이에요. 멤버들은 방찬, 리노, 창빈, 현진, 한, 필릭스, 승민, 아이엔으로 구성되어 있어요 <sup>1</sup>. 하지만 제가 찾은 정보 중에는 멤버들의 출신 학교에 대한 정보가 없네요. 다른 정보가 필요하신가요?

자세한 정보: [1. ko.wikipedia.org](https://ko.wikipedia.org) [2. ko.wikipedia.org](https://ko.wikipedia.org) 1 / 30 ●

?

스트레이키즈 멤버들의 생년월일을 알려줘.

스트레이키즈의 노래 중에서 인기 있는 노래는 뭐야?

#### 4. bard

- 국적도 틀리고 학력도 틀림

다른 답안.

✦ 스트레이 키즈 멤버의 국적과 출신 학교는 다음과 같습니다.


멤버	국적	출신 학교
방찬	대한민국	서울공연예술고등학교
리노	대한민국	서울공연예술고등학교
창빈	대한민국	한림연예예술고등학교
리키	대한민국	한림연예예술고등학교
아이엔	대한민국	한림연예예술고등학교
한	대한민국	한림연예예술고등학교
필릭스	오스트리아	멜버른 예술 대학교
승민	대한민국	한림연예예술고등학교
현진	대한민국	한림연예예술고등학교
Sheets로 내보내기		

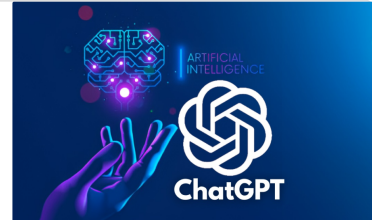
스트레이 키즈는 대한민국의 9인조 보이 그룹입니다. 2018년 3월 25일 미니 앨범 "I am NOT"으로 데뷔했습니다.

### 3.3 Natural Language to GraphQuery with gpt

- 자연어 입력을 sparql로 정확하고 빠르게 바꿀 수 있으면 신뢰성 높은 답변을 가져올 수 있다.

SPARQL queries, GPTs and Large Language Models – where are we currently?

 <https://www.wisecube.ai/blog/sparql-queries-gpts-and-large-language-models-where-are-we-currently/>



#### ChatGPT for Information Retrieval from Knowledge Graph

This article proposes a workflow for information retrieval from a knowledge graph, combining ChatGPT and other tools, to improve the factuality of ChatGPT in domain-specific tasks. The key objective is to test ontology-guided


 <https://apex974.com/articles/chatgpt-for-info-retrieval-from-knowledge-graph>

#### ChatGPT for Information Retrieval from Knowledge Graph



**Fan Li, Ph.D.**  
apex974.com

#### 1. ontology extraction

 You are an ontologist and your job is to identify keywords related to entities and properties for a given question.

Format:

```
{
  "entities": ["<entity_1>", ...],
  "properties": ["<property_1>", ...]
}
```

Example:

Question: "더글로리에 출연한 배우와 그 배우들의 태어난 곳을 알려줘"

Answer:

```
{
  "entities": ["더글로리", "배우"],
  "properties": ["출연", "태어난곳"]
}
```

Question: "스트레이키즈 멤버의 국적과 출신 학교를 알려줘."

Answer:



```
{
  "entities": ["스트레이키즈", "멤버"],
  "properties": ["국적", "출신 학교"]
}
```



## 2. SPARQL Generation



Create a SPARQL query based on the following ontology to answer question: 스트레이 키즈 멤버의 국적과 출신 학교를 알려줘

<https://predicate/학력>  
<https://predicate/국적>  
<https://predicate/소속>



Sure! Here's a SPARQL query based on the given ontology to answer the question "스트레이 키즈 멤버의 국적과 출신 학교를 알려줘" (Provide the nationality and alma mater of Stray Kids members):

```
sparql
PREFIX : <https://predicate/>

SELECT ?member ?nationality ?almaMater
WHERE {
  ?member :국적 ?nationality ;
          :학력 ?almaMater .
  FILTER (?member = <https://entity/스트레이키즈>).
}
```

This query selects the variables `?member`, `?nationality`, and `?almaMater`. It retrieves the `:국적` (nationality) and `:학력` (alma mater) properties for the entity `<https://entity/스트레이키즈>`, which represents Stray Kids. The query will return the member, their nationality, and alma mater for each member of Stray Kids.

## 4. Future Work

- sparql에 대한 이해 부족으로 여러가지 케이스를 테스트 해보지 못함.
- 다양한 alternate name 추가하거나, name에 대해 임베딩 기반 접근 할 수 있으면 답변에 대한 유연성이 커질 것 같다.
- 전처리를 제대로 안해서 잘못된 triple들이 남아있다.
- 온톨로지가 있으면 정답 찾기 더 좋다.
  - 온톨로지란: 지식베이스의 스키마

- 개념/클래스간의 관계(company - subclass\_of - organization)를 나타내고, 각 엔티티/인스턴스는 클래스에 속함(삼성전자 - instance\_of - company)
- 온톨로지란 사람들이 세상에 대하여 보고 듣고 느끼고 생각하는 것에 대하여 서로 간의 토론을 통하여 합의를 이룬 바를, 개념적이고 컴퓨터에서 다룰 수 있는 형태로 표현한 모델로, 개념의 타입이나 사용상의 제약조건들을 명시적으로 정의한 기술이다.
- 온톨로지 예시
  - <https://schema.org/docs/full.html>
  - <https://yago-knowledge.org/>
- 온톨로지 설계하고 엔티티를 맵핑하는 것은 또 하나의 커다란 태스크....
- [사족] 제너레이션 모델은 애초에 디비검색의 보완적 역할일지도???

## How does Bard work?

Google wants Bard to supplement the [Knowledge Graph Cards](#) you see in Search when making queries that have a simple answer. While a Knowledge Graph Card can supply you with a word's definition or an overview of a person or place, Bard's responses are meant to address NORA questions (searches with No One Right Answer), as Google calls them.

## 작성자 소개

.

## Q&A

- 찬란: 공유의 문화에 동참해주시고, 머신러닝 생태계를 건강하게 만들어주셔서 감사합니다! 이 지식이 확산되어 더 큰 가치를 형성할 것입니다!

## 댓글

- 찬란: 공유의 문화에 동참해주시고, 머신러닝 생태계를 건강하게 만들어주셔서 감사합니다! 이 지식이 확산되어 더 큰 가치를 형성할 것입니다!