

## Text to SPARQL을 위한 지식 증강 프롬프팅 연구

이연진<sup>°</sup>, 남정재<sup>°</sup>, 김우영, 김우주\*  
연세대학교 스마트 시스템 연구실  
{leeanna828, njj97, timothy, wkim}@yonsei.ac.kr  
<sup>°</sup>공동 1저자, \*교신저자

### Study on Knowledge Augmented Prompting for Text to SPARQL

Yeonjin Lee<sup>°</sup>, Jeongjae Nam<sup>°</sup>, Wooyoung Kim, Wooju Kim\*  
Smart System Lab, Yonsei University  
<sup>°</sup>Co-First Author, \*Corresponding Author

#### 요약

Text to SPARQL은 지식 그래프 기반 질의응답의 한 형태로 자연어 질문을 지식 그래프 검색 쿼리로 변환하는 태스크이다. SPARQL 쿼리는 지식 그래프의 정보를 기반으로 작성되어야 하기 때문에 기존 언어 모델을 통한 코드 생성 방법으로는 잘 동작하지 않는다. 이에 우리는 거대 언어 모델을 활용하여 Text to SPARQL을 해결하기 위해 프롬프트에 지식 그래프의 정보를 증강시켜주는 방법론을 제안한다. 이에 더하여 다국어 정보 활용에 대한 영향을 검증하기 위해 한국어, 영어 각각의 레이블을 교차적으로 실험하였다. 추가로 한국어 Text to SPARQL 실험을 위하여 대표적인 Text to SPARQL 벤치마크 데이터셋 QALD-10을 한국어로 번역하여 공개하였다. 위 데이터를 이용해 지식 증강 프롬프팅의 효과를 실험적으로 입증하였다.

**주제어:** 지식 그래프 기반 질의응답(KGQA), Text to SPARQL, 지식 증강 프롬프팅, 거대언어 모델

## 1. 서론

지식 그래프 기반 질의응답(Knowledge Graph Question Answering, KGQA)은 사용자 질의에 대해 지식 그래프에서 정확한 답변을 찾는 태스크이다. 기존 언어 모델은 사전학습 데이터에 의존적이며, 정보의 추가 및 변화에 대처하기 어렵다. 뿐만 아니라, 언어 모델은 단순한 프로그래밍으로 처리 할 수 있는 집계 연산에 대해서도 취약하다. 반면에, 지식 그래프는 정보의 추가 및 변화에 쉽게 대처할 수 있으며, 지식 그래프의 쿼리인 SPARQL을 이용하면 집계연산을 쉽게 처리할 수 있다. 따라서 이러한 언어 모델의 한계를 극복하기 위한 방법으로 Text to SPARQL은 훌륭한 보완제가 될 수 있다. 하지만 SPARQL을 생성하기 위해서는 지식 그래프의 구조 정보가 필요하기 때문에 기존 코드 생성 방법으로는 해결이 어렵다. 이를 극복하기 위해 지식 그래프 정보를 활용하는 지식 증강 프롬프팅 방법론을 제안한다.

본 고는 2장에서 연구와 관련된 지식 그래프 기반 질의응답, 거대 언어 모델 및 거대 언어 모델을 활용하기 위한 지식 증강 프롬프팅에 대한 사전 지식을 설명한다. 3장에서는 우리가 제안하는 방법론, 이어지는 4장에서는 이를 검증하기 위한 한국어 데이터의 정보를 상술하였다. 5장에서는 지식 증강 프롬프팅 및 교차언어 실험에 대한 결과를 보고한다. 마지막으로 6, 7장에서는 최종적인 결론과 연구의 한계 및 향후 연구 방향에 대하여 제안한다.

## 2. 관련연구

### 2.1 지식 그래프 기반 질의응답

지식 그래프 기반 질의응답은 질문에 대한 정답을 지식 그래프(Knowledge Graph, KG)에서 추출하는 태스크이다. KGQA는 자연어 질의를 지식 그래프의 구조와 결합시키는 Semantic Parsing[1], 지식 그래프의 Embedding을 언어 모델의 Latent Space와 결합시키는 Knowledge Enhanced Language Model[2], 자연어 질문을 SPARQL 쿼리 형태로 구조화시킨 후 지식 그래프에서 해당하는 정보를 추출하는 Text to SPARQL 등이 있다.[3][4]

여러 접근 방법 중 Text to SPARQL은 SPARQL의 실행 결과를 명시적으로 해석할 수 있으며, SPARQL 문법에서 제공하는 집계 연산 등을 활용 할 수 있어 기존 언어 모델만으로 해결이 어려운 문제를 풀어나갈 수 있다.[5] 또한 지속적으로 업데이트 되는 정보에 대해 미세조정을 해야하는 기존 언어 모델과는 다르게 새로운 지식을 즉각적으로 반영 가능한 지식 그래프를 활용할 수 있다는 점에서 주목받고 있다.

QALD(Question Answering over Linked Data)는 Text to SPARQL 태스크의 대표적인 벤치마크 데이터셋으로 최근 다양한 국적의 사용자가 접근가능 하도록 꾸준히 다중 언어(multilingual) 버전을 발표한다. QALD-9-plus[6]는 9개의 언어, QALD-10[7]은 4개의 언어를 지원하지만 한국어 데이터셋은 포함하지 않는다. 한국어 Text to SPARQL 연구를 위해서는 한국어로 표현된 데이터셋이 필요하다. 이를 위해 우리는 QALD 데이터셋을 한국어로 확장하였다.(§4.)

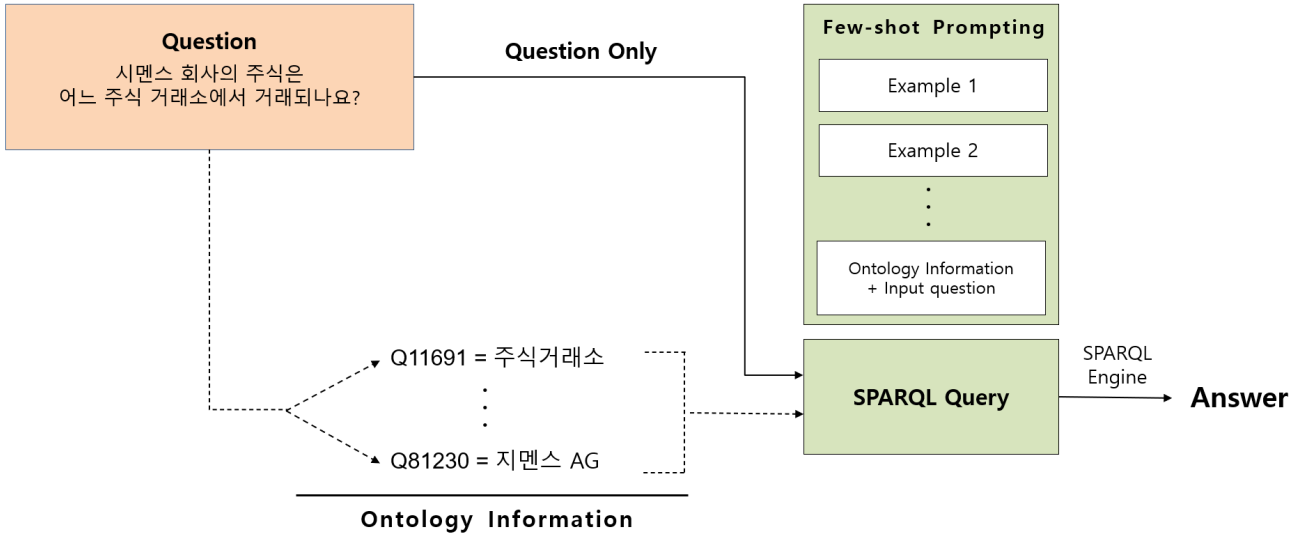


그림 1. 지식 증강 프롬프팅 프레임워크

## 2.2 지식 증강 프롬프팅

사전학습된 거대 언어 모델(Large Language Model, LLM)은 파라미터에 저장된 내부 지식을 기반으로 제로샷(zero-shot), 퓨샷(few-shot) 질의응답 태스크를 수행할 수 있다.[8] 하지만, 언어 모델의 매개변수에 내장된 지식은 사전학습 데이터에 의존적이기 때문에 외부 지식을 필요로 하는 태스크(Knowledge Intensive Task)를 해결하기에는 불충분하다. 이러한 한계를 극복하기 위해 언어 모델의 프롬프트에 직접적으로 지식을 주입하는 지식 증강 프롬프팅 연구가 활발히 보고되고 있다. [9] 대표적으로 KAPING[10]은 LLM에 지식 그래프 정보를 증강시켜 언어 모델에 외부 정보를 효과적으로 증강시켰다. Text to SPARQL 태스크의 경우 지식 그래프의 구조 정보를 기반으로 SPARQL을 생성해야되기 때문에 지식 증강 프롬프팅이 필연적으로 요구된다.

## 2.3 거대모델 기반 코드 생성 (LLM code generation)

최근 거대 언어 모델은 기존 전통적인 NLP 태스크뿐만 아니라 코드생성에서도 널리 활용되고 있다. GPT[8]를 미세조정된 OpenAI의 Codex[11]는 Python, JavaScript 등 12개 이상의 프로그래밍 언어 생성이 가능하다. DeepMind의 AlphaCode[12] 또한 복잡한 알고리즘을 프로그래밍 언어로 변환할 수 있도록 깃허브에 존재하는 대량의 코드에 대해 학습하여 코드 생성에 널리 활용된다.

하지만 SPARQL은 지식 그래프 정보를 참고하여 작성해야 하기 때문에 다른 프로그래밍 언어와 다르게 기존 LLM의 코드 생성 방법으로는 잘 작동하지 않는다. 따라서 Text to SPARQL 태스크는 일반적인 코드 생성 프로세스와 달리 지식 증강 프롬프팅을 활용하는 방향으로 접근해야 한다.

## 3. 방법론

Text to SPARQL 태스크를 수행하기 위해서 우리는 지식 증강 프롬프팅 방법을 제안한다. 지식 증강 프롬프팅은 거대 언어 모델에 지식 그래프의 엔티티와 릴레이션 정보를 증강해주기 위한 방법이다. 이를 위해서 그림1의 하단과 같이 입력 문장( $x$ )에 해당하는 지식 그래프의 엔티티와 릴레이션을 추출하고, 그에 대한 식별자( $E = \{e_1, e_2, \dots\}$ ,  $R = \{r_1, r_2, \dots\}$ )를 연결해줘야 한다. 입력 문장( $x$ )와 연결된 식별자 집합( $E, R$ )에 대해 퓨샷 인스트럭션 템플릿  $T$ 를 사용하여 프롬프트  $x'$ 를 생성한다. (표 1) 본 연구는 QALD 데이터셋의 입력 문장에 활용된 엔티티와 릴레이션을 골든 라벨에서 추출해 활용하였다.

$$T : (x, E, R) \rightarrow x' \quad (1)$$

거대 언어 모델은 생성된 프롬프트  $x'$ 를 입력으로 받아 SPARQL 쿼리  $q$ 을 생성한다.

$$LLM(x') = q \quad (2)$$

SPARQL Engine ( $SE$ )은 지식 그래프 정보가 증강되어 생성된 쿼리  $q$ 을 실행하여 예측 정답  $\hat{y}$ 를 도출한다.

$$SE(q) = \hat{y} \quad (3)$$

## 4. 한국어 Text to SPARQL 데이터셋

우리는 QALD-10 벤치마크 데이터셋[7]을 활용하여 한국어 Text to SPARQL 데이터셋을 생성하였다. QALD-10 데이터셋은 412개의 학습 데이터와 394개의 평가 데이터로 구성되어 있다. QALD-10은 영어, 독일어, 프랑스어, 스페인어 총 4가지 언어에 기반한 자연어 질문과 그에 대응되는 SPARQL 쿼리 그리고 정답으로 구성되어 있다.

표 1. 프롬프트 예시

분류	프롬프트 예시
한국어 질문(Baseline)	Text : 테슬라의 최고 경영자가 설립한 회사는 어떤 것들이 있나요? Ontology Information :
지식 그래프 영어 레이블 엔티티 정보 증강 + 한국어 질문	Q478214 = Tesla, Inc. Q4830453 = business Text : 테슬라의 최고 경영자가 설립한 회사는 어떤 것들이 있나요? Ontology Information :
지식 그래프 영어 레이블 엔티티, 릴레이션 정보 증강 + 한국어 질문	Q478214 = Tesla, Inc. Q4830453 = business P112 = founded by P169 = chief executive officer Text : 테슬라의 최고 경영자가 설립한 회사는 어떤 것들이 있나요? Ontology Information :
지식 그래프 한글 레이블 엔티티 정보 증강 + 한국어 질문	Q478214 = 테슬라 Q4830453 = 사업 Text : 테슬라의 최고 경영자가 설립한 회사는 어떤 것들이 있나요? Ontology Information :
지식 그래프 한글 레이블 엔티티, 릴레이션 정보 증강 + 한국어 질문	Q478214 = 테슬라 Q4830453 = 사업 P112 = 설립자 P169 = 최고 경영자 Text : 테슬라의 최고 경영자가 설립한 회사는 어떤 것들이 있나요?

우리는 한국어 Text to SPARQL 데이터셋을 구축하기 위해, GPT-3.5-Turbo[13]를 활용하여 QALD-10의 영어 질문을 한국어로 기계번역 하였다. 기계번역 된 한국어 질문은 한국어로 모국어로 사용하는 세 명의 대학원생에 의하여 검수 및 교정되었다. 데이터에 대한 상세한 정보는 GitHub\*를 통해 확인할 수 있다.

## 5. 실험

### 5.1 실험환경

한국어 Text to SPARQL 태스크를 위하여 대표적인 거대 언어 모델인 GPT-3.5-Turbo 모델을 활용하였다. 우리는 지식 증강 프롬프팅의 영향을 검증하기 위해 베이스라인을 포함해 총 다섯 가지 프롬프트를 활용하여 실험을 진행하였다. 또한,

거대 언어 모델이 태스크를 잘 이해할 수 있도록 여덟 개의 SPARQL 생성 예시를 함께 주어 퓨샷 프롬프팅으로 실험을 진행하였다.

베이스라인은 별도의 지식 그래프 정보 증강 없이 한국어 질문만으로 프롬프트를 구성하였다. 지식 증강 프롬프팅은 엔티티 레이블만 추가한 것과 엔티티 및 릴레이션 레이블 모두를 추가하여 구성하였다. 이때 레이블은 언어에 따른 영향이 존재하는지 파악할 수 있도록 한국어 버전과 영어 버전을 각각 활용하였다.

구체적으로 지식 그래프 정보를 영어 레이블로 추가시켜주는 경우는 지식 그래프 엔티티 정보만 영어로 증강된 한국어 질문 프롬프트와 지식 그래프 엔티티, 릴레이션 정보가 모두 영어로 증강된 한국어 질문 프롬프트로 구성하였다. 또한, 지식 그래프 정보를 한국어 레이블로 추가시켜주는 경우는 동일하게 지식 그래프 엔티티 정보 또는 엔티티, 릴레이션 정보가 한국어로

\*[https://github.com/yeonjin-8289/QALD\\_with\\_ko/](https://github.com/yeonjin-8289/QALD_with_ko/)

표 2. 평가결과

분류	Accuracy
한국어 질문(Baseline)	4.57%
지식 그래프 영어 레이블 엔티티 정보 증강 + 한국어 질문	12.44%
지식 그래프 영어 레이블 엔티티, 릴레이션 정보 증강 + 한국어 질문	25.63%
지식 그래프 한국어 레이블 엔티티 정보 증강 + 한국어 질문	14.47%
지식 그래프 한국어 레이블 엔티티, 릴레이션 정보 증강 + 한국어 질문	30.20%

증강된 한국어 질문 프롬프트를 활용하였다. 자세한 프롬프트 예시는 표1에서 확인 가능하다.

본 방법론의 평가지표로는 생성된 SPARQL 쿼리를 실행하여 얻은 결과와 데이터셋에서 제공되는 골든 SPARQL의 실행 결과의 정확도(Accuracy)를 활용하였다.

## 5.2 실험결과

Text to SPARQL 실험을 위해 프롬프트 총 다섯 가지를 설계하였고, 각 프롬프트별 실험에 대한 성능 평가 결과이다. 표 2에서 평가지표인 정확도 기준으로 각 실험별 성능을 확인할 수 있다.

위 결과에서 지식 그래프 엔티티 정보만 증강시키는 것보다 릴레이션 정보도 함께 증강시켜주는 것이 성능이 더 좋음을 알 수 있다. 프롬프트 설계 시, 한국어 질문에 지식 그래프 영어 및 한국어 레이블 엔티티 정보를 증강시켰을 때 결과는 각각 약 12%, 14% 이고, 릴레이션 정보를 함께 증강시켰을 때 결과는 각각 약 25%, 30% 이다. 릴레이션 정보 유무에 따라 성능이 최대 16% 까지 차이 나는 것을 확인할 수 있다. 이로써, 모델 프롬프트에 릴레이션 정보를 함께 제공하여 모델이 SPARQL 쿼리를 생성하는데 있어 적절한 릴레이션 정보를 활용하는 쿼리 생성 능력을 향상시킨다.

다음으로, 지식 그래프 엔티티 및 릴레이션 정보를 제공할 때 각 정보를 영어로 제공하는 것보다 한국어로 제공하는 것이 성능이 더 좋음을 알 수 있다. 영어 레이블과 한국어 레이블에 대한 성능을 비교하였을 때, 성능이 최대 5% 까지 차이난다. 모든 실험에 대해 성능 비교하였을 때, SPARQL 쿼리 생성에 효과적인 프롬프트는 한국어 질문에 지식 그래프 한국어 레이블 엔티티 및 릴레이션 정보를 증강시켜주는 식으로 설계하는 것임을 알 수 있다.

## 6. 결론

우리는 Text to SPARQL을 수행하기 위해 지식 증강 프롬프팅 방법론을 제안하였다. SPARQL 쿼리는 지식 그래프의 정보를 참고하여 생성해야 한다는 점에서 일반적인 코드 생성

태스크와는 다른 매커니즘이 필요하다. 본 논문에서는 지식 그래프의 엔티티와 릴레이션 정보를 거대 언어 모델에 증강하는 방법으로 SPARQL 쿼리 생성의 난점을 극복하였다.

실험적으로, 엔티티에 대한 정보를 증강하였을 때 효과적인 성능 향상이 있음을 보였다. 유사하게 엔티티 및 릴레이션 정보를 완전하게 증강하였을 때 가장 좋은 성능을 보였다. 이에 더하여, 교차언어 실험 결과에서는 한국어 질문에 대하여 동일한 언어의 레이블을 증강하였을 때 보다 큰 성능 향상이 있음을 입증하였다.

## 7. 한계 및 제언

본고의 연구에서는 입력 문장에 해당하는 엔티티와 릴레이션 정보를 데이터셋에서 추출하였다. 하지만 실제 사용 시나리오를 고려하였을 때 타당하지 못 한 방법이다. 이를 해결하기 위해, 주어진 질문으로부터 지식 그래프의 정보를 자동으로 추출하는 연구가 후속되어야 한다. 또한 완전한 지식 그래프의 엔티티 및 릴레이션 정보를 증강하고, 최고 수준의 거대 언어 모델을 사용하였음에도 불구하고 절대적인 성능은 낮은 실정이다. 이를 해결하기 위해 Text to SPARQL 태스크에 대하여 많은 추가적인 연구가 필요로 하며, 우리가 공개한 QALD-10 버전의 한국어 데이터셋이 동료 연구자들의 연구에 도움이 되길 기대한다.

## 감사의 글

본 연구는 국토교통부/국토교통과학기술진흥원의 지원으로 수행되었음(과제번호 RS-2023-00244769).

## 참고문헌

- [1] D. Banerjee, P. A. Nair, J. N. Kaur, R. Usbeck, and C. Biemann, "Modern baselines for sparql semantic parsing," *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2260–2265, 2022.

- [2] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, “Luke: deep contextualized entity representations with entity-aware self-attention,” *arXiv preprint arXiv:2010.01057*, 2020.
- [3] H. Jung and W. Kim, “Automated conversion from natural language query to sparql query,” *Journal of Intelligent Information Systems*, Vol. 55, No. 3, pp. 501–520, 2020.
- [4] M. Yani and A. A. Krisnadhi, “Challenges, techniques, and trends of simple knowledge graph question answering: a survey,” *Information*, Vol. 12, No. 7, p. 271, 2021.
- [5] X. Li, R. Zhao, Y. K. Chia, B. Ding, L. Bing, S. Joty, and S. Poria, “Chain of knowledge: A framework for grounding large language models with structured knowledge bases,” *arXiv preprint arXiv:2305.13269*, 2023.
- [6] A. Perevalov, D. Diefenbach, R. Usbeck, and A. Both, “Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers,” *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*, pp. 229–234, 2022.
- [7] R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. Möller, J. Huang, J. Reineke, A.-C. N. Ngomo *et al.*, “Qald-10—the 10th challenge on question answering over linked data.”
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, Vol. 33, pp. 1877–1901, 2020.
- [9] G. Mialon, R. Dessì, M. Lomeli, C. Nalmpantis, R. Pasunuru, R. Raileanu, B. Rozière, T. Schick, J. Dwivedi-Yu, A. Celikyilmaz *et al.*, “Augmented language models: a survey,” *arXiv preprint arXiv:2302.07842*, 2023.
- [10] J. Baek, A. F. Aji, and A. Saffari, “Knowledge-augmented language model prompting for zero-shot knowledge graph question answering,” *arXiv preprint arXiv:2306.04136*, 2023.
- [11] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, “Evaluating large language models trained on code,” *arXiv preprint arXiv:2107.03374*, 2021.
- [12] Y. Li, D. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, T. Eccles, J. Keeling, F. Gimeno, A. Dal Lago *et al.*, “Competition-level code generation with alphacode,” *Science*, Vol. 378, No. 6624, pp. 1092–1097, 2022.
- [13] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in Neural Information Processing Systems*, Vol. 35, pp. 27 730–27 744, 2022.