ArcaneQA: Dynamic Program Induction and Contextualized Encoding for Knowledge Base Question Answering

THE OHIO STATE UNIVERSITY

Yu Gu and Yu Su

The Ohio State University

Motivation

❑ **Fundamental Challenges in KBQA**

Large search space

- Freebase: 45M entities, 3B facts
- Google KG: 5B entities, 500B facts

Schema linking

- Variation in natural language
- Diversity of KB schemas

Intertwined

❑ **Existing Ranking-Based methods**

- Enumerate candidate queries to a limited complexity

Unscalable

- Perform semantic matching afterwards

Error Propagation

Key Contributions

✓ We propose a **generation-based model** that handles both challenges with a unified framework.

✓ A **dynamic program induction** module is used to intelligently prune the search space.

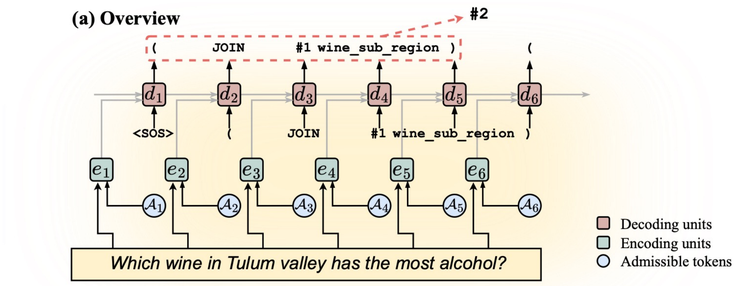
✓ A **dynamic contextualized encoding** module uses BERT to jointly represent a natural language question and the KB schema.

✓ We provide **Unified meaning representations** for multiple KBQA datasets.

Author Contact: gu.826@osu.edu

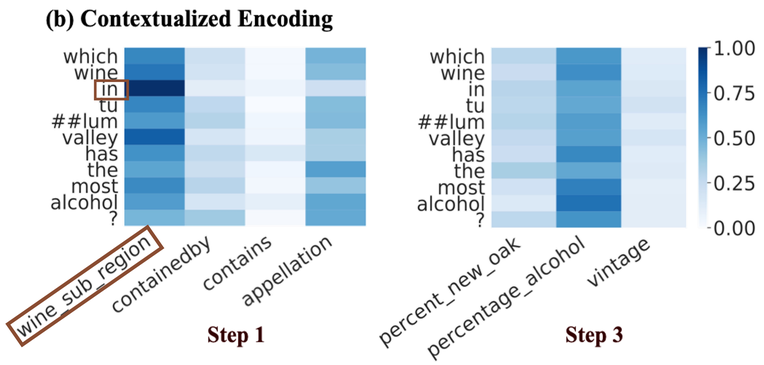
Model Overview

❑ A unified **encoder-decoder model** with BERT-based encoding and execution-based constrained decoding.



Dynamic Contextualized Encoding

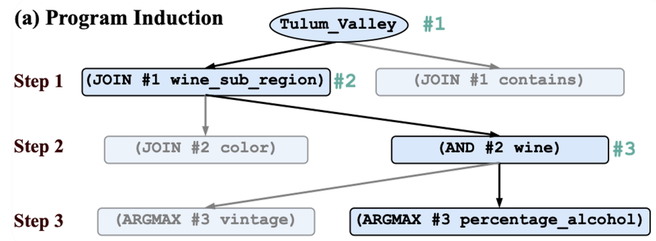
❑ Contextualized encoding from BERT provides better representations for schema linking.



Schema linking is implicitly achieved via BERT's self-attention

Dynamic Program Induction

❑ A complicated KB query can be generated by predicting a sequence of subqueries.



Executions of prior predictions guide the search for the future ones

Current function	Admissible actions
JOIN	$\{r \mid h \in \#, (h, r, t) \in \mathcal{K}_r\}$
AND	$\{v \mid v \in \mathcal{S}, v \cap \# \neq \emptyset\} \cup \{c \mid e \in \#, (e, c) \in \mathcal{K}_c\}$
ARGMAX/ARGMIN	$\{r \mid h \in \#, t \in \mathcal{L}, (h, r, t) \in \mathcal{K}_r\}$
LT(LE/GT/GE)	$\{r \mid t < (\leq / > / \geq) \#, (h, r, t) \in \mathcal{K}_r\}$
COUNT	$\{\}$
CONS	$\{(r, t) \mid h \in \#, (h, r, t) \in \mathcal{K}_r\}$
TC	$\{(r, t) \mid h \in \#, (h, r, t) \in \mathcal{K}_r, t \in \mathcal{L}$ is a temporal expression$\}$

Table 1: A set of rules for constrained decoding given a function name.

Experiments

❑ **Effectiveness Results**

Model	Overall EM	Overall F1	I.I.D. EM	I.I.D. F1	Compositional EM	Compositional F1	Zero-shot EM	Zero-shot F1
QGG* (Lan and Jiang, 2020)	–	36.7	–	40.5	–	33.0	–	36.6
BERT+Transduction* (Gu et al., 2021)	33.3	36.8	51.8	53.9	31.0	36.0	25.7	29.3
BERT+Ranking* (Gu et al., 2021)	50.6	58.0	59.9	67.0	45.5	53.9	48.6	55.7
ReTraCk (Chen et al., 2021)	58.1	65.3	84.4	87.5	61.5	70.9	44.6	52.5
RnG-KBQA* (Ye et al., 2021)	61.4	67.4	78.0	81.8	55.0	63.2	56.7	63.0
ArcaneQA*	58.8	67.2	77.8	81.6	58.0	66.1	50.4	61.8
RnG-KBQA (Ye et al., 2021)	**68.8**	**74.4**	**86.2**	**89.0**	63.8	71.2	**63.0**	**69.2**
ArcaneQA	63.8	73.7	85.6	88.9	**65.8**	**75.3**	52.9	66.0
w/o contextualized encoding	49.7	59.1	77.6	82.1	50.5	59.4	36.5	48.5

Table 2: Overall results on GrailQA across different levels of generalization.

❑ **Efficiency Results**

	BERT+Ranking	RnG-KBQA	ArcaneQA
Time (s)	115.5	82.1	5.6

Table 3: Online execution time per question without caching.