Beyond NED: Fast and Effective Search Space Reduction for Complex Question Answering over Knowledge Bases

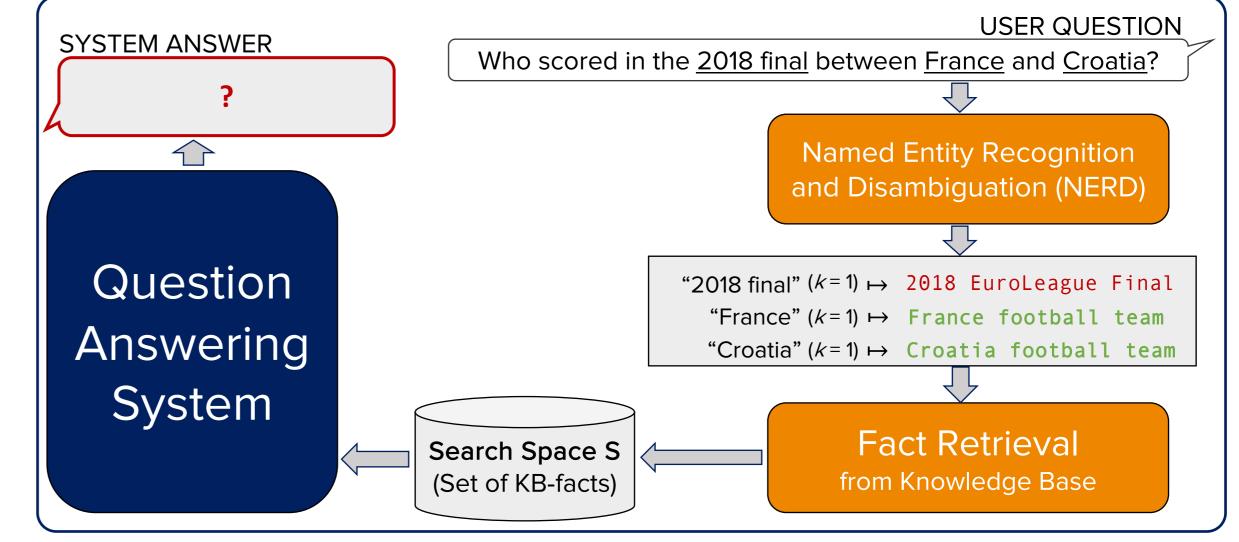
Philipp Christmann, Rishiraj Saha Roy, and Gerhard Weikum Max Planck Institute for Informatics, Germany



CLOCQ can be used for efficient retrieval of relevant KB-facts for any query

SEARCH SPACE REDUCTION: OVERLOOKED PROBLEM IN KB-QA

- * Curated knowledge bases (KB) have billions of facts, with millions of entities and thousands of predicates
- **A** Question answering (QA) system is typically a neural model that requires input to be sufficiently small
- ★ Therefore, QA system not directly applicable to full KB
- **Typical KB-QA pipeline first reduces the search space** to a few thousands of facts, that are **relevant to the question** and likely to contain the answer(s)



Standard KB-QA pipeline, leveraging facts of disambiguated entities to prune the search space.

STANDARD NED: ONE SIZE FITS ALL

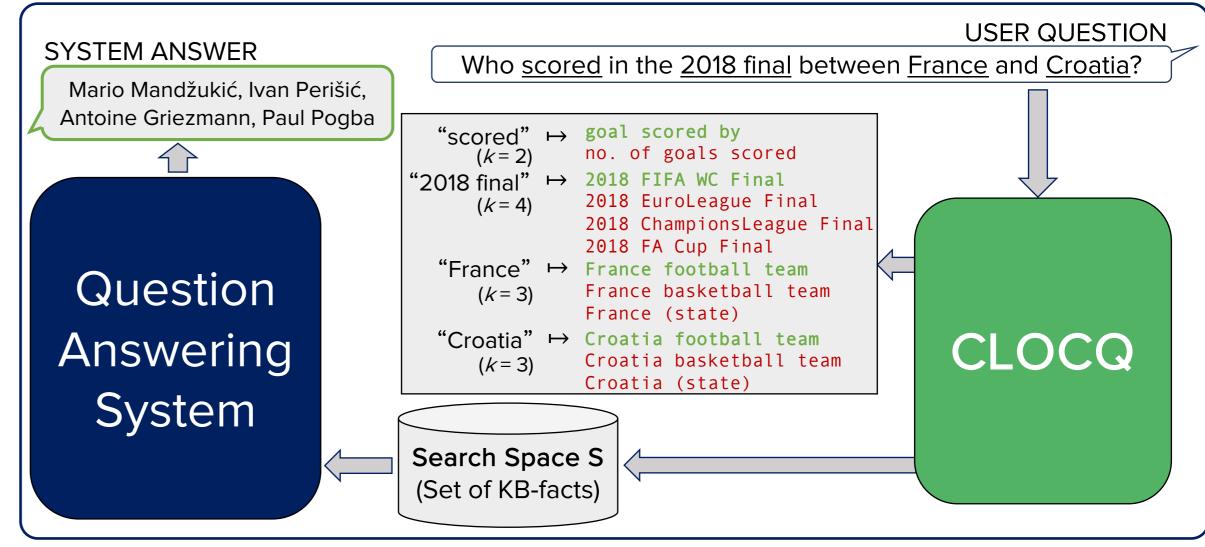
- Top-1 NED too restrictive for QA (single error can cause failure of full QA pipeline)
- \star Top-*k* NED can easily include noise, **blowing up** the search space
- **★** Important cues within the question (e.g. scored in example on the right) are not considered, given that most NED methods disambiguate entity mentions only

APPROACH: GOING BEYOND NED

- \star Adaptive top-k: choose **appropriate** k, for each question word **individually**
- **T** Disambiguate all KB-items (entities, types, concepts, predicates)
- * Consider proximity of disambiguated KB-items within graph underlying the KB
- **★** Establish a novel **fact-centric KB-index** for more **efficient** access to the KB

PIPELINE: SEARCH SPACE REDUCTION FOR COMPLEX KB-QA

- **KB-item retrieval:** Retrieve *d* KB-item candidates per question word
- Auto-k: Choose k automatically for each question word based on its ambiguity
- Candidate scoring: Identify top-k candidates using signals for pair-wise KB-proximity, 3. pair-wise semantic coherence, question relevance, lexical matching
- 4. Search space retrieval: Retrieve facts for disambiguated KB-items from knowledge base using our fact-centric KB-index; consider only salient facts, pruning potentially



KB-QA pipeline with CLOCQ: All KB-items are disambiguated, with an adaptive k per question word.

Graph-based definition of KB-distance	Triple-centric KB-index		
goal 2018 FIFA	France football team —		
Paul Pogba goal 2018 FIFA location Moscow	France football te	eam country	France
2-hop 3-hop 4-hop 5-hop 7			
France 1-hop	fact-id	for team	France football tear
football for team	fact-id —		
	fact-id	for team	France football tea
KB-distance(France football team, 2018 FIFA World Cup Final) = 3	fact-id	goal scored by	Paul Pogba
KB-distance (France football team, Moscow) = 5	2018 FIFA WC Final	goal scored by	fact-id
Paul Pogba goal scored by France football team	France football team, country, France] [France football team, home venue, Stade de France [France football team, coach, Didier Deschamps, start time, 8 July 2012]		
In 1-hop of "France football team" In 2-hop of "France football team" KB-distance (France football team, 2018 FIFA World Cup Final) = 1 KB-distance (France football team, Moscow) = 2	[2018 FIFA WC Fina for team, France f		i by, Paul Pogba,
Legend Entity node Predicate node	→ Data in index	for KB-item	[•] KB-fact

noisier facts based on a parameter *p* (e.g. facts with US as object or qualifier-object)

CLOCQ makes use of a fact-centric KB-index, which stores n-ary facts in a way that allows for efficient retrieval.

CLOCQ enhances answer presence in search space for QA systems

ConvQuestions	LC-QuAD 2.0	RESULTS ON LC-QuAD 2.0	
Question: "How is the main soccer club of the german city Düsseldorf called?"	Question: "Who is the composer of All We Know?"		
<u>CLOCQ</u> "main" ↦ (Frankfurt am Main, Main(river), Offenbach	CLOCQ "composer" → (composer, film score composer); "All We Know" → (All We Know (Paramore), For All We	CLOCQ 82.6 ★	CLOCQ +GRAFT-Net 19.7
<pre>am Main >; "soccer" → {football, football team >; "club" → {Nightclub, Torina F.C. >;</pre>	Know (album), All We Know (Chainsmokers), For All We Know (Carpenters), For All We Know (1934 song));	CLOCQ (<i>k</i> =1) 80.0	TagME+HDT +GRAFT-Net17.1
"german" → (German, German Empire); "city" → (Manchester City F.C., Birmingham City F.C., Stoke City F.C., Cardiff City F.C.);	TagME "Who" ↦ (The Who); "composer" ↦ (composer); "All We Know" ↦ (For All We Know (Carpenters));	TagME+HDT 76.8	ELQ+HDT 16.8 +GRAFT-Net
<pre>"Düsseldorf" → (Düsseldorf, Fortuna Düsseldorf); <u>TagME</u> "main" → (Main (river));</pre>	ELQ "All We Know" ↦ (All We Know (Chainsmokers));	ELQ+HDT 76.7	Precision at 1 for QA task when feeding outputs of search space reduction into QA system GRAFT-Net
<pre>"soccer" → 〈football〉; "club" → 〈sports club〉; "german" → 〈Germany〉; "city" → 〈City of London〉;</pre>	Question: "Who is the son of the brother of Queenie Padilla?" CLOCQ	EARL+HDT 65.9	<i>k</i> =5 79.7 83.8 87.2 92.4 90.0 87.5 <i>k</i> =5 0.6k 2.7k 15.1k 2.0M 2.00M
<pre>ELQ "german" → 〈Germany〉; "soccer" → 〈football〉;</pre>	<pre>"son" → (Son en Breugel, nephew, Mae Hong Son, Porto de Son); "brother" → (sibling); "Queenie Padilla" → (Queenie Padilla);</pre>	AIDA+HDT 60.5	k=3 76.8 80.9 84.6 91.0 85.0 0.4k 1.7k 9.7k 1.6M 1.50M k=2 74.9 79.2 82.9 88.9 77.5 k=2 0.3k 1.2k 6.7k 1.1M
Question: "Who was the screenwriter for Crazy Rich Asians?" <u>CLOCQ</u>	TagME"Who" ↦ (World Health Organization);"brother" ↦ (Brother);	REL+HDT 55.8	$k=1 70.5 74.5 78.2 82.9 75.0 \\ 72.5 \\ 70.0 k=1 0.2k 0.6k 3.7k 0.5M \\ p=100 p=1k p=10k all all p=10k all al$
<pre>"screenwriter" → (screenwriter); "Crazy Rich Asians" → (Crazy Rich Asians (film)); TagME</pre>	<u>ELQ</u> "Padilla" → {Zsa Zsa Padilla};		Answer presence Search space size
<pre>"Crazy Rich Asians" → (Crazy Rich Asians (book)); ELQ "Crazy Rich Asians" → (Crazy Rich Asians (book));</pre>	Anecdotal examples for which only CLOCQ had an answer in the search space.	Answer presence after search space reduction on LC-QuAD 2.0 dataset.	Trade-off between answer presence and search space size.







