A Reasoning-Based Approach to Cryptic Crossword Clue Solving

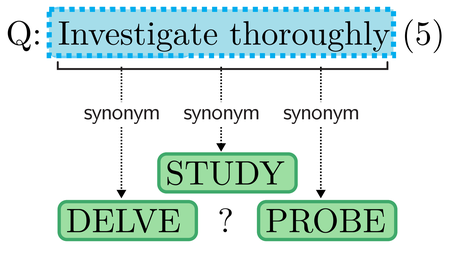
Martin Andrews
martin@RedDragon.ai

Sam Witteveen
sam@RedDragon.ai

RED DRAGON AI

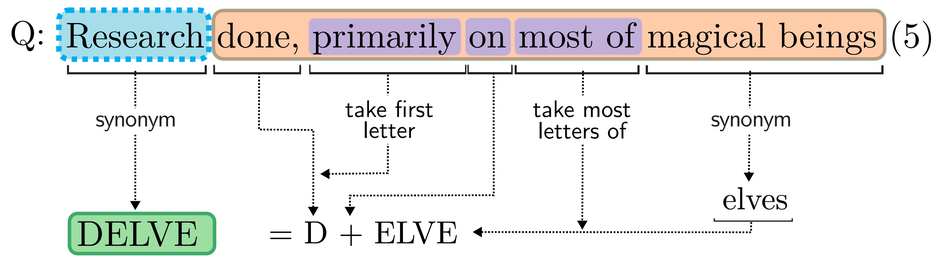
ICML International Conference On Machine Learning

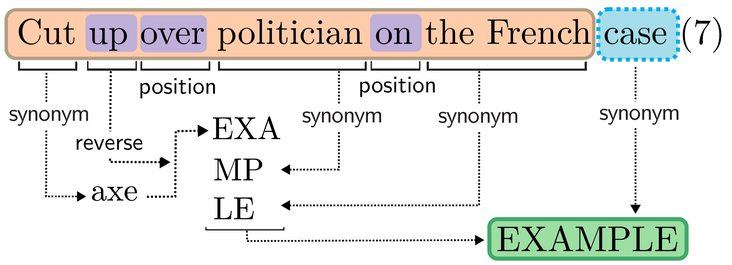
Regular Crossword Clue



▲ Regular clue answers are not unique

Cryptic Crossword Clues : Hard-for-Humans NLP Reasoning Task, with Verifiable Answers





▲ Cryptic clue answers are proved two ways : Definition must agree with the Wordplay

Reasoning Formalisation in Python

Gemma2 answer candidate

```
def proof(answer="DELVE",
          clue="research done, primarily on most of magical beings",
          pattern='5'):
```

Gemma2 wordplay

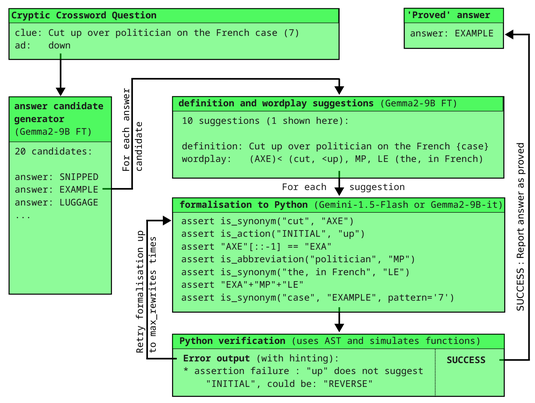
```
  """
  definition: research done, primarily, on most of magical beings
  wordplay: D[one] (primarily) ELVE[s] (magical beings, most of)
  """
```

LLM formalisation

```
  assert action_type("primarily", Action.INITIALS)
  assert "DONE"[:1] == "D"
  assert is_synonym("magical beings", "ELVES")
  assert action_type("most of", Action.REMOVE_LAST)
  assert "ELVES"[:-1] == "ELVE"
  assert "D"+"ELVE" == "DELVE"
  assert is_synonym("research", "DELVE", pattern='5')
proof()
```

▲ Finding: LLMs prefer to generate Python over new DSL

Test-Time Computation



▲ Verification enables more guesses to be analysed
▲ ... Leads to higher accuracy (since false positive rate is low)

DSL : Python In-Context

```
Action=Enum('Action',
            'ANAGRAM,REMOVE_FIRST,INITIALS,REMOVE_LAST,'+
            'GOES_INSIDE,GOES_OUTSIDE,REVERSE,SUBSTRING,HOMOPHONE')
# External definitions
def is_synonym(phrase:str, test_synonym:str, pattern:str='') -> bool:
  # True if 'test_synonym' is a reasonable synonym for 'phrase',
  # with letters optionally matching 'pattern'
def is_abbreviation(phrase:str, test_abbreviation:str) -> bool:
  # Determines whether 'test_abbreviation' is
  # a valid abbreviation or short form for 'phrase'
def action_type(phrase:str, action:Action) -> bool:
  # Determines whether 'phrase' might signify the given 'action'
def is_anagram(letters:str, word:str) -> bool:
  # True if 'word' can be formed from 'letters' (i.e. an anagram)
def is_homophone(phrase:str, test_homophone:str) -> bool:
  # Determines whether 'test_homophone' sounds like 'phrase'
```

▲ DSL is 'declared' in the LLM context
▲ The functions return expected results,

Results

Model	samples	Test Overall	Test Quick	Test Hard
Rule-based (*)	26k	8.6%	13.5%	5.8%
T5-large (770M) FT (*)	26k	7.6%	12.8%	3.4%
Gemma2-9B-it 5-shot	1000	4.5%	10.5%	4.0%
Gemini-Flash 5-shot	1000	6.5%	11.8%	6.1%
GPT-4o 5-shot	1000	27.6%	47.4%	26.0%
Gemma2-9B FT	1000	15.9%	38.2%	14.1%
Gemma2-9B freq (#=20)	1000	25.5%	**55.3%**	23.1%
(AB) logprob answer	500	22.7%	55.3%	20.1%
(AB) logprob wordplay	200	20.5%	46.7%	18.4%
Gemini-Flash Formaliser	200	**32.5%**	46.7%	**31.4%**
Gemma2 9B-it Formaliser	200	29.0%	46.7%	27.6%
Gemma2 9B-FT Formaliser	200	29.5%	53.3%	27.6%

▲ SOTA results on the Cryptonite dataset
▲ Can apply test-time compute to improve results using Gemma2-9B pipeline

Summary

Domain :

▲ Cryptic Crossword clues combine:
- NLP + World knowledge
- Reasoning

Approach :

▲ Use code to prove answer correctness

Process :

▲ For every clue:
▲ ... Generate 20 candidate answers
▲ ... Guess 10 Wordplays for each
▲ ... Convert to Python to validate
▲ Creates interpretable reasoning

Cryptic Crosswords Globally

Characteristics :

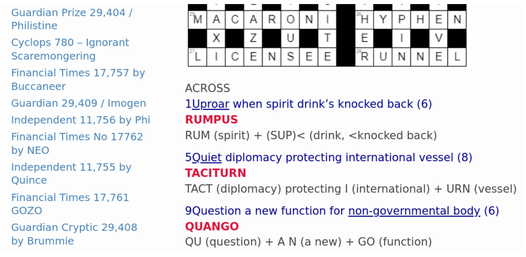
▲ Published In daily newspapers around world
- Cryptonite dataset = 570,000 Clues+Answers
- Verified to be of high quality
- New test sets being issued daily

▲ Challenging task :
- Not easy for native speakers
- Experts can solve entire puzzles at 100% rate

▲ Only correct answer is 'provable'
▲ Clues can be answered in independently

Wordplay Dataset

▲ Gathered from enthusiast websites
▲ Thousands of wordplay annotations
▲ Dataset splits aligned with Cryptonite



https://github.com/mdda/cryptic-wordplay

Key References

▲"Cryptonite: A cryptic crossword benchmark for extreme ambiguity in language" - Efrat et al. (2021)
▲"Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs" - Jiang et al. (2022)
▲"Code generation with AlphaCodium: From Prompt Engineering to Flow Engineering" - Ridnik et al. (2024)

Contact & Acknowledgements

martin@RedDragon.ai +65 8585 1750
http://RedDragon.ai/research

Support for this research was provided by the Google AI Developer Programs team, including access to the Gemini models and GPUs on Google Cloud Platform.