RiddleSense: Reasoning about Riddle Questions
Featuring Linguistic Creativity and Commonsense Knowledge

Bill Yuchen Lin  Ziyi Wu  Yichi Yang  Dong-Ho Lee  Xiang Ren
{yuchen.lin, ziyiwu, yichiyan, dongho.lee, xiangren}@usc.edu
Department of Computer Science and Information Sciences Institute,
University of Southern California

Abstract

Question: I have five fingers but I am not alive. What am I? Answer: a glove.

Answering such a riddle-style question is a challenging cognitive process, in that it requires complex commonsense reasoning abilities, an understanding of figurative language, and counterfactual reasoning skills, which are all important abilities for advanced natural language understanding (NLU). However, there is currently no dataset aiming to test these abilities. In this paper, we present RIDDLESENSE, a new multiple-choice question answering task, which comes with the first large dataset (5.7k examples) for answering riddle-style commonsense questions. We systematically evaluate a wide range of models over the RIDDLESENSE challenge, and point out that there is a large gap between the best-supervised model and human performance — suggesting intriguing future research in the direction of higher-order commonsense reasoning and linguistic creativity towards building advanced NLU systems.

1 Introduction

“The essence of a riddle is to express true facts under impossible combinations.”
— Aristotle, Poetics (350 BCE)

A riddle is a puzzling question about concepts in our everyday life. For example, a riddle might ask “My life can be measured in hours. I serve by being devoured. Thin, I am quick. Fat, I am slow. Wind is my foe. What am I?” The correct answer “candle,” is reached by considering a collection of commonsense knowledge: a candle can be lit and burns for a few hours; a candle’s life depends upon its diameter; wind can extinguish candles, etc.

It is believed that the riddle is one of the earliest forms of oral literature, which can be seen as a formulation of thoughts about common sense, a mode of association between everyday concepts, and a metaphor as higher-order use of natural language (Hirsch, 2014). Aristotle stated in his Rhetoric (335-330 BCE) that good riddles generally provide satisfactory metaphors for rethinking common concepts in our daily life. He also pointed out in the Poetics (350 BCE): “the essence of a riddle is to express true facts under impossible combinations,” which suggests that solving riddles is a nontrivial reasoning task.

Answering riddles is indeed a challenging cog-
nitive process as it requires complex commonsense reasoning skills. A riddle can describe multiple pieces of commonsense knowledge with figurative devices such as metaphor and personification (e.g., “wind is my foe → extinguish”). Moreover, counterfactual thinking is also necessary for answering many riddles such as “what can you hold in your left hand but not in your right hand? → your right elbow.” These riddles with ‘but-no’ cues require that models use counterfactual reasoning ability to consider possible solutions for situations or objects that are seemingly impossible at face value. This reporting bias (Gordon and Van Durme, 2013) makes riddles a more difficult type of commonsense question for pretrained language models to learn and reason. In contrast, superficial commonsense questions such as “What home entertainment equipment requires cable?” in CommonsenseQA (Talmor et al., 2019) are more straightforward and explicitly stated. We illustrate this comparison in Figure 1.

In this paper, we introduce the RIDDLESENSE challenge to study the task of answering riddle-style commonsense questions requiring creativity, counterfactual thinking and complex commonsense reasoning. RIDDLESENSE is presented as a multiple-choice question answering task where a model selects one of five answer choices to a given riddle question as its predicted answer, as shown in Fig. 1. We construct the dataset by first crawling from several free websites featuring large collections of human-written riddles and then aggregating, verifying, and correcting these examples using a combination of human rating and NLP tools to create a dataset consisting of 5.7k high-quality examples. Finally, we use Amazon Mechanical Turk to crowdsource quality distractors to create a challenging benchmark. We show that our riddle questions are more challenging than CommonsenseQA by analyzing graph-based statistics over ConceptNet (Speer et al., 2017), a large knowledge graph for common sense reasoning.

Recent studies have demonstrated that fine-tuning large pretrained language models, such as BERT (Devlin et al., 2019a), RoBERTa, and ALBERT (Lan et al., 2020), can achieve strong results on current commonsense reasoning benchmarks. Developed on top of these language models, graph-based language reasoning models such as KagNet (Lin et al., 2019) and MHGRN (Feng et al., 2020) show superior performance. Most recently, UnifiedQA (Khshabi et al., 2020) proposes to unify different QA tasks and train a text-to-text model for learning from all of them, which achieves state-of-the-art performance on many commonsense benchmarks.

To provide a comprehensive benchmarking analysis, we systematically compare the above methods. Our experiments reveal that while humans achieve 91.33% accuracy on RIDDLESENSE, the best language models can only achieve 68.80% accuracy, suggesting that there is still much room for improvement in the field of solutions to complex commonsense reasoning questions with language models. We believe the proposed RIDDLESENSE challenge suggests productive future directions for machine commonsense reasoning as well as the understanding of higher-order and creative use of natural language.

2 Construction of RIDDLESENSE

In this section, we first present our pipeline for collecting the RIDDLESENSE dataset, including the details of data cleaning. We introduce how we design a crowd-sourcing protocol for annotating quality distractors to turn riddle-solving into a multiple-choice question answering task.

2.1 Riddle Crawling and Cleaning

We write web crawlers for collecting a large number (approximately 10,000) of riddles and their answers from public riddle websites, such as brainzilla.com, riddlewot.com, etc. As the crawled data contain much noise such as inconsistent answer format and misspelled words, we process riddles through careful data cleaning as well as human verification. First, we use an open-source tool for detecting typos\(^3\) and then refine the sentences. Then we continuously sample (riddle, answer) pairs and recognize errors, for which we iteratively improve our program with a set of conditions to filter out noisy examples that are not readable or have ambiguous answers. Also, we merge the riddles from different sources while removing duplicate riddle questions with similar answers. For detecting duplicate riddles with minor word changes, we use SentenceBERT (Reimers and Gurevych, 2019) to find clusters with high cosine similarities.

\(^2\)We use “riddle” and “riddle-style commonsense question” interchangeably in this paper.

\(^3\)github.com/phatpiglet/autocorrect
2.2 Distractor Collection from AMT

We consider a multi-choice question answering format rather than the open-ended format, as it is easier to meaningfully compare the performance of different models in a more controlled manner — there is a limited range of options. For such a dataset, given a riddle-style question and 5 answer options, the model should select the best one as the predicted answer. This format offers a straightforward and fair evaluation metric — accuracy, which is the metric adopted by many popular commonsense reasoning benchmarks such as CommonsenseQA, ARC (Clark et al., 2018), and OpenbookQA (Mihaylov et al., 2018).

High-quality distractors are essential for multiple-choice question answering tasks as they can ensure that the dataset is both clean and challenging — the distractors are neither too similar nor too distant from the correct answer. We thus design a protocol to collect quality distractors from human annotators via Amazon Mechanical Turk based on a pool of candidate distractors.

Candidate Distractor Pool We use $Q$ to denote the concepts that are mentioned in the question, and $a$ to denote the concept in the answer. We then first get all two-hop neighbors in the ConceptNet of $a$ and one-hop neighbors of each $c \in Q$ respectively:

$$A = \{x| (x, r_i, y), (y, r_j, a)\}$$

$$B = \{x| (x, r_k, c), \forall c \in Q\}$$

$$D = A \cap B,$$

where $r_{i/j/k}$ is a binary relation in the ConceptNet such as HasProperty. The final intersection, $D$, is thus the pool of distractor candidates. We further use WordNet (Miller, 1992) to filter out concepts that have either too low or too high Wu-Palmer similarity. We argue that such sampled distractors are semantically relevant to both question concepts and answer concepts, forming a set of candidate distractors $D$ for annotators to choose from. Workers are required to select at least 5 concepts that they think are good distractors to the question. There are at least 3 different workers for each question and we take the candidates which are selected by at least two different workers to make sure the selected distractors are indeed meaningful.

AMT Crowd-sourcing We design a three-stage annotation protocol:

- **S1) Sanity Check.** We show a question and 3 choices where only 1 choice is correct and the other 2 are randomly sampled concepts from the full vocabulary of ConceptNet. Only when the workers pass this sanity check, their following annotations will be considered, so we can avoid noise from random workers.
- **S2) Candidate Selection.** As it is difficult to control and verify the quality of distractors from crowd workers, we first sample concepts from ConceptNet, which are relevant to both question concepts and answer concepts, forming a set of candidate distractors $D$ for annotators to choose from. Workers are required to select at least 5 concepts that they think are good distractors to the question. There are at least 3 different workers for each question and we take the candidates which are selected by at least two different workers to make sure the selected distractors are indeed meaningful.
- **S3) Open Distractor Collection.** We also ask master workers on AMT to write at least one more distractor based on the question context. This stage is important because sometimes the candidate pool contains fewer candidates of good quality and the human-written distractors are usually better than the ones in the candidate pool. We thus give extra bonus credits to encourage annotators to write more quality distractors.

<table>
<thead>
<tr>
<th></th>
<th>CSQA</th>
<th>RS</th>
</tr>
</thead>
<tbody>
<tr>
<td># All Examples</td>
<td>12,102</td>
<td>5,715</td>
</tr>
<tr>
<td># Train Examples</td>
<td>9,741</td>
<td>3,510</td>
</tr>
<tr>
<td># Validation Examples</td>
<td>1,221</td>
<td>1,021</td>
</tr>
<tr>
<td># Test Examples</td>
<td>1,140</td>
<td>1,184</td>
</tr>
<tr>
<td>Average Question Length</td>
<td>15.06</td>
<td>24.04</td>
</tr>
<tr>
<td>% Long Qs (&gt;20 tokens)</td>
<td>16.5%</td>
<td>47.3%</td>
</tr>
<tr>
<td>Distinct Question Words</td>
<td>6,822</td>
<td>7,110</td>
</tr>
<tr>
<td>Distinct Choice Words</td>
<td>7,044</td>
<td>9,912</td>
</tr>
<tr>
<td>Avg PLL of Qs</td>
<td>-34.41</td>
<td>-53.98</td>
</tr>
<tr>
<td>QA-NLI Conflict</td>
<td>12.7%</td>
<td>39.6%</td>
</tr>
<tr>
<td>QA-NLI Neutral</td>
<td>71.6%</td>
<td>44.9%</td>
</tr>
<tr>
<td>QA-NLI Entailment</td>
<td>15.7%</td>
<td>15.5%</td>
</tr>
</tbody>
</table>

Table 1: Key statistics of the RIDDLESENSE dataset (v1.1) vs the CommonsenseQA (CSQA) dataset.
(c) Min Length

Figure 2: The Q-A paths serve as estimation of underlying reasoning chains. Fig. (a) illustrates how to compute mean/min/max of the Q-A paths: \( \{q_1, q_2, q_3\} \) are three concepts mentioned in the question, and \( a \) is the answer concept. \( L_k \) is the length of the shortest path between \( q_k \) and \( a \) over ConceptNet; \( \text{min/max/mean} \) are computed over \( \{L_1, L_2, L_3\} \) as three aspects to measure the overall difficulty. Fig. (b), (c), and (d) show that generally RIDDLESENSE has a longer question-answer path than CommonsenseQA, thus being harder to reason.

Algorithm 1: Get statistics of QA paths.

Input: Knowledge graph \( KG = (V, E) \), riddle question \( Q \), riddle answer \( A \)

Output: \( \min\text{PathLength}, \max\text{PathLength}, \text{mean\text{PathLength}} \)

1. \( QC \leftarrow \text{extractConcept}(Q) \)
2. \( AC \leftarrow \text{extractConcept}(A) \)
3. \( ac \leftarrow v \in AC \) with smallest \( \deg(G, v) \)
4. \( l \leftarrow [] \)
5. foreach \( qc \in QC \) do
   6. \( \text{path} \leftarrow \text{shortestPathLen}(KG, qc, ac) \)
   7. if \( \text{path} \neq \text{None} \) then
      8. \( l.\text{append}(\text{path}) \)
9. \( \min\text{PathLength} \leftarrow \min(l) \)
10. \( \max\text{PathLength} \leftarrow \max(l) \)
11. \( \text{mean\text{PathLength}} \leftarrow \text{mean}(l) \)

3 Data Analysis of RIDDLESENSE

In this section, we first report the key statistics of the proposed RIDDLESENSE dataset, then we compare it to CommonsenseQA (Talmor et al., 2019) from two major angles: the distribution of the lengths of Q-A paths and the types of reasoning chains, which serve as an effective proxy to analyze the differences between the two datasets.

3.1 Key Statistics

Table 1 presents the key statistics of RIDDLESENSE (RS) and the comparisons with CommonsenseQA (CSQA) which is the most similar benchmark to ours. Although the size of RS is smaller than CSQA, we argue that RS is complementary to the CSQA dataset and introduces novel challenges for the commonsense reasoning community. As they share the same format, we can test different methods by training on either CSQA-only, RS-only, or the concatenation of CSQA and RS, as we show later in Section 4.

Moreover, there is a greater number of long questions (i.e., containing more than 20 words) in RS than in CSQA. Additionally, we find that RS questions have a lower normalized pseudo-likelihood (PLL) (Salazar et al., 2020), a proxy of estimating sentence probability, suggesting that RS questions are more puzzling (i.e., the words are less frequently co-occurring). We also use a RoBERTa model fine-tuned on MNLI (Williams et al., 2018) to perform natural language inference between CSQA/RS questions and their answers. There is a much greater proportion of questions in RS that have conflicting relations with their correct answers than compared to CSQA. This is indicative of RS’s complexity due to the self-contradictory and perplexing nature of riddles.

Interestingly, we also find that although there are about twice as many examples in CSQA as RS, there are more distinct words in the questions and answer choices of RS than CSQA, suggesting that RS covers more diverse topics than CSQA.

3.2 Distribution of the Lengths of Q-A Paths

Our main intuition is that the shortest paths between question concepts and the answer concepts can approximate the underlying reasoning chains, which are hidden and difficult to label. To understand the difference between CSQA and RS in
terms of their reasoning chains, we use Q-A paths over ConceptNet as a proxy. For a riddle question, a set of Q-A path lengths are the lengths of the shortest paths between every question concept and the answer concept, i.e., shortestPathLen(KG, qc, ac) in Alg. 1. For a question-answer pair, we first extract the concepts mentioned in the question and the answer respectively (extractConcept()) in Algorithm 1), following the steps of Lin et al. (2019) and Feng et al. (2020). If there are three question concepts \{q1, q2, q3\} and an answer concept a, we denote their shortest path lengths as \{L1, L2, L3\}. Finally, we compute the min/max/mean over them for a comprehensive understanding of the approximated difficulty of this riddle — a greater value indicates a more challenging example.

As shown in Figure 2 (b), we can see that RS has longer Q-A paths as underlying reasoning chains. In addition, we can see that RS generally has longer chains, particularly the min of CSQA is 1-hop for more than 80% of examples. On the other hand, only about 30% of RS examples have 1-hop minimum Q-A paths, while about 50% of the examples have 2-hop min Q-A paths. The distribution over the maximum in Figure 2 (d) also shows that RS tends to have longer maximum paths than CSQA. We also show the percentage of all Q-A paths of different length as part of Table 2, and we can see that RS has longer paths in general (e.g., CSQA = 14.0% vs. RS = 4.6% in 1-hop).

### 3.3 Relational Types of Reasoning Paths

In addition to the analysis on path length, we also show that the relation types of Q-A paths for RS and CSQA have clear differences, as shown in Table 2. The types of reasoning chains in RS rely more on a special relation in ConceptNet — Related, which is relatively more implicit and can not be grounded to a specific, explicit relation such as AtLoc (e.g., <wind, Related, air> vs. <lamp, AtLoc, table>). The most frequent relation between question concepts and answer concepts in CSQA is the AtLoc relation (4.8%), however, it is Related (3.1%) in RS. We define implicit-ratio for k-hop paths, \( \rho_k = \frac{E_k}{\%(\text{related} \times k)} \), where \( E_k \) is the most frequent type of chains with at least one explicit relation of length \( k \). In RS, \( \rho_k \) is around 4.1 ~ 7.8, while it is about 0.7 ~ 1.8 for CSQA. Thus, we conclude that the dominant reasoning chains in RS are much more implicit, and consequently RS is more challenging to reason with using commonsense knowledge resources like ConceptNet.

### 4 Experiments

We first introduce three types of popular baseline methods for commonsense reasoning (Section 4.1), then we present our main experimental results with analysis (Section 4.2), and finally show case studies for error analysis (Section 4.3).
4.1 Baseline Methods

Given a riddle question $q$, there are 5 different choices $\{c_1, \ldots, c_5\}$, where only one of them is the correct choice and the others are distractors. The model needs to rank all choices and select the best one as the final answer. There are three major types of models for commonsense reasoning tasks in this format: 1) fine-tuning pretrained language models, 2) incorporating relevant knowledge graphs for reasoning, 3) fine-tuning a unified text-to-text QA model, as shown in Figure 3.

Fine-tuning Pre-trained LMs As we seek to investigate how well current NLU models can perform in higher-order commonsense reasoning, we first experiment with a typical set of large pretrained language models such as BERT (Devlin et al., 2019b), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020). We concatenate the question with each choice, using [SEP] as the separator, thus forming a statement. Then, we fine-tune any pretrained LMs like BERT to use their [CLS] token embeddings to predict a score for each statement. Then, a set of five scores about an example will be fed to SoftMax to optimize for maximizing the score of the correct choice.

LMs + Graph Reasoning Modules KagNet (Lin et al., 2019) and MHGRN (Feng et al., 2020) are two typical graph-based language reasoning models. They both extract a schema graph from ConceptNet, i.e., a subgraph of ConceptNet consisting of Q-A paths in Figure 2, by incorporating them with a graph encoding module. They finally fuse the external commonsense knowledge with a text encoder (e.g., a pretrained LM). KagNet uses heuristics to prune irrelevant paths and then encode them with path-based LSTM and hierarchical attention to select the most important paths for improving commonsense reasoning. In contrast, the recent MHGRN explicitly encodes multi-hop paths at scale using graph networks with relational attention, improving efficiency and performance over KagNet and other models. A unique merit of such graph-based models is their interpretability due to the neural attention over the symbolic structures of KGs.

Fine-tuning a Text-to-Text QA Model UnifiedQA (Khashabi et al., 2020), the state-of-the-art multiple-choice QA model, simply concatenates the question with all answer candidates as a single input sequence to a T5 (Raffel et al., 2020) model for learning to generate the correct choice as extracting a span from the input. Apart from the multiple-choice QA format, it is also trained with other QA task formats so that it can benefit from many other QA datasets (including CSQA) via sharing the model parameters.

Human Evaluation We invite three native English speakers who study computer science to solve 100 riddle examples sampled from the test set. They achieved an average accuracy of 91.3%.

4.2 Results and Analysis

We show the main results of the experiments in Table 3. There are 3 settings according to the different training data options: 1) the training data of CSQA, 2) the training data of RS, and 3) the concatenation of both RS and CSQA, while all experiments are validated over the dev set of RS. However, as the public UnifiedQA checkpoints were already trained on CSQA (together with many other QA datasets), we directly use them for inference over RS in the first setting (i.e., “Train=CSQA”). This also suggests that the performance of UnifiedQA models in 2nd setting should be better than others although they all are fine-tuned on RS’s training data only.
Table 3: Benchmark performance over the dev and test set of RiddleSense.

<table>
<thead>
<tr>
<th>Models ↓ Training Data →</th>
<th>Train = CSQA</th>
<th>Train = RiddleSense</th>
<th>Train = RS+CSQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Guess</td>
<td>20.0 20.0</td>
<td>20.0 20.0</td>
<td>20.0 20.0</td>
</tr>
<tr>
<td>BERT-Base (Devlin et al., 2019a)</td>
<td>33.59 34.61</td>
<td>54.16 42.43</td>
<td>56.22 47.67</td>
</tr>
<tr>
<td>BERT-Large (Devlin et al., 2019a)</td>
<td>36.14 39.10</td>
<td>55.24 45.09</td>
<td>57.69 54.91</td>
</tr>
<tr>
<td>RoBERTa-Large (Liu et al., 2019)</td>
<td>43.68 47.42</td>
<td>60.72 52.58</td>
<td>66.11 59.82</td>
</tr>
<tr>
<td>ALBERT-XXL (Lan et al., 2020)</td>
<td><strong>51.03</strong> <strong>51.00</strong></td>
<td>66.99 60.65</td>
<td><strong>71.50</strong> 67.30</td>
</tr>
<tr>
<td>KagNet (RoBERTa-L) (Lin et al., 2019)</td>
<td>42.66 48.24</td>
<td>61.77 53.72</td>
<td>66.55 59.72</td>
</tr>
<tr>
<td>MHGRN (RoBERTa-L) (Feng et al., 2020)</td>
<td>46.83 49.65</td>
<td>63.27 54.49</td>
<td>66.90 63.73</td>
</tr>
<tr>
<td>MHGRN (ALBERT-XXL) (Feng et al., 2020)</td>
<td>50.89 50.21</td>
<td>66.27 59.93</td>
<td>70.81 66.81</td>
</tr>
<tr>
<td>UnifiedQA (T5-Large) (Khashabi et al., 2020)</td>
<td>28.50 37.27</td>
<td>56.21 56.40</td>
<td>58.17 56.57</td>
</tr>
<tr>
<td>UnifiedQA (T5-3B) (Khashabi et al., 2020)</td>
<td>37.32 50.25</td>
<td><strong>67.38</strong> <strong>66.06</strong></td>
<td>68.26 <strong>68.80</strong></td>
</tr>
</tbody>
</table>

| Human Performance | - 91.33 | - 91.33 | - 91.33 |

Figure 4: The curve of dev accuracy using different percentage of the RS-training data, respectively for RoBERTa-Large and ALBERT-XXL.

We can see that larger pretrained language understanding models always gain better performance, ranging from BERT-base to Albert-XXL, which gets the best performance in this group of baselines (67.30%). This matches their performance comparisons on CSQA and other benchmark datasets as well, suggesting that a better pretrained language model can be also identified by RiddleSense as well. Interestingly, we find that ALBERT-XXL is so powerful that it can generalize from training on CSQA only but achieve comparable results with RoBERTa-Large that is trained over RS (i.e., 51.0% vs. 52.6%). However, if we look at the curve of dev accuracy when using different percentage of the RS-train data (setting 2) in Figure 4, we can see that RoBERTa-Large can generally outperform ALBERTA-XXL when using less than 60% data for fine-tuning.

Moreover, we find that the KG-enhanced models, KagNet and MHGRN, using RoBERTa-Large (RB-L) as the encoder, perform better than vanilla RB-L. Although the Q-A paths over ConceptNet have more implicit paths (e.g., Related × k), some paths can still be beneficial. For example,

\[
\text{wind} \xrightarrow{\text{Related}} \text{blow} \xrightarrow{\text{Related}} \text{candle},
\]

can still help reason about the riddle “... Wind is my foe. What am I?” to the answer “candle.”

The fusion of ConceptNet also improves in the situation when only training with CSQA data using RoBERTa-Large. However, the improvement of KagNet is negative, which is unexpected. We conjecture that this is because the extracted subgraphs from the ConceptNet does not guarantee the reasoning path from question concepts to answer concepts, while the training phase forces models to learn to reason over those graphs, yielding a possibly harmful impact. Additionally, we find that MHGRN with ALBERT-XXL also results in a worse performance, unlike using RoBERTa-Large. We believe this may be related to the specific design of ALBERT, which reuses model parameters for multiple layers, and thus it could be a problem when fused with another learnable module (e.g., a graph network in MHGRN).

Fine-tuning UnifiedQA with T5-3B achieves the best performance, which is also the case for CSQA in their leaderboard. This is expected for two reasons: 1) UnifiedQA has been trained over multiple other QA datasets, which increases its...
generalization ability, 2) UnifiedQA considers all choices together at a time and thus can better compare different choices with self-attention mechanism of Transformer (Vaswani et al., 2017).

4.3 Error Analysis and Future Directions

We show a few examples that are mistakenly predicted by the UnifiedQA-3B model in Figure 5. From these concrete cases, we can see that even the best model cannot solve riddles that can be trivial to humans, especially when there are metaphors and/or counterfactual situations. We argue that future research should aim to address the creative use of language in commonsense reasoning and general understanding of language, as creativity is a critical feature of natural language. We list several promising directions as follows.

First of all, we should mine (semi-)structured knowledge of metaphors, so that concepts can connect via metaphorical links (e.g., “tail” → “thread”). Second, to prevent false inferences, we need more complete, precise commonsense knowledge of concepts. For example, in Figure 5, a model should know a chair only has exactly four legs instead of hundreds (Lin et al., 2020a); ink can be black or red, but it won’t change over time. However, current KGs only have (leg, PartOf, chair) and (ink, HasProperty, black/red). In addition, the reasoning methods should incorporate more symbolic logic rules, so that the multi-hop conditions and counterfactual “but-no” negations will be handled better. Finally, we think the graph-augmented methods should be improved to compare multiple options in a schema graph, e.g., QA-GNN (Yasunaga et al., 2021). Both KAGNET and MHGGRN consider only a single option at a time which prevents them from effectively reasoning about the subtle differences between options.

5 Related Work

Benchmarking Machine Common Sense

The prior works on building commonsense reasoning benchmarks touch different aspects of commonsense reasoning: SWAG (Zellers et al., 2018), HellaSWAG (Zellers et al., 2019), CODAH (Chen et al., 2019), aNLI (Bhagavatula et al., 2020) for situation-based reasoning; Physical IQA (Bisk et al., 2020) on physical knowledge; Social IQA (Sap et al., 2019) on social psychology knowledge; LocatedNearRE (Xu et al., 2018) on mining spatial commonsense knowledge; DoQ (Elazar et al., 2019) and NumerSense (Lin et al., 2020a) on numerical commonsense; CommonGen (Lin et al., 2020b) for generative commonsense reasoning, and many others; OpenCSR (Lin et al., 2021) and ProtoQA (Boratko et al., 2020) aim to test commonsense reasoning ability in an open-ended setting.

CommonsenseQA (Talmor et al., 2019) has the same format as our proposed RIDDLESENSE, and both target general commonsense knowledge via multiple-choice question answering. However, CSQA focuses more on straightforward questions where the description of the answer concept is easy to understand and retrieval over ConceptNet, while RS makes use of riddle questions to test higher-order commonsense reasoning ability. More detailed comparisons between them are in Section 3, which shows that the unique challenges of the RiddleSense on multiple dimensions.

Commonsense Reasoning Methods

Our experiments cover three major types of commonsense reasoning methods that are popular in many benchmarks: fine-tuning pretrained LMs (Devlin et al., 2019a; Liu et al., 2019; Lan et al., 2020), graph-based reasoning with external KGs (Lin et al., 2019; Feng et al., 2020), and fine-tuning unified text-to-text QA models (Khashabi et al., 2020). Apart from ConceptNet, There are also some methods (Lv et al., 2020; Xu et al., 2020) using additional knowledge resources such as Wikipedia and Wiktionary. A few recent methods also aim to generate relevant triples via language generation models so that the context graph is more beneficial for reasoning (Wang et al., 2020; Yan et al., 2020). Our experiments in this paper aim to compare the most typical and popular methods which have open-source implementations, which we believe are beneficial for understanding the limitation of these methods in higher-order commonsense reasoning — RIDDLESENSE.

Computational Creativity and NLP

Creativity has been seen as a central property of the human use of natural language (McDonald and Busa, 1994). Text should not be always taken at face value, however, higher-order use of language and figurative devices such as metaphor can communicate richer meanings and needs deeper reading and more complicated reasoning skills (Veale, 2011). Recent works on processing language with creative use focus on metaphor detection (Gao
et al., 2018), pun generation (He et al., 2019; Luo et al., 2019), creative story generation, and humor detection (Weller and Seppi, 2019, 2020), sarcasm generation (Chakrabarty et al., 2020), etc.

Riddling, as a way to use creative descriptions to query a common concept, are relatively underexplored. Previous works (Tan et al., 2016; Gonçalo Oliveira and Rodrigues, 2018) focus on the generation of riddles in specific languages and usually rely on language-specific features (e.g., decomposing a Chinese character into multiple smaller pieces). There is few datasets or public resources for studying riddles as a reasoning task, to the best of our knowledge. The proposed RIDDLESENSE is among the very first works connecting commonsense reasoning and computational creative, and provides a large dataset to train and evaluate models for answering riddle questions.

6 Conclusion

We propose a novel commonsense reasoning challenge, RIDDLESENSE, which requires complex commonsense skills for reasoning about creative and counterfactual questions, coming with a large multiple-choice QA dataset. We systematically evaluate recent commonsense reasoning methods over the proposed RIDDLESENSE dataset, and find that the best model is still far behind human performance, suggesting that there is still much space for commonsense reasoning methods to improve. We hope RIDDLESENSE can serve as a benchmark dataset for future research targeting complex commonsense reasoning and computational creativity.

Acknowledgements

This research is supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007, the DARPA MCS program under Contract No. N660011924033 with the United States Office Of Naval Research, the Defense Advanced Research Projects Agency with award W911NF-19-20271, and NSF SMA 18-29268. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. We would like to thank all the collaborators in USC INK research lab and the reviewers for their constructive feedback on the work.

Ethical Considerations

Copyright of Riddles. The RiddleSense dataset is consistent with the terms of use of the fan websites and the intellectual property and privacy rights of the original sources. All of our riddles and answers are from fan websites that can be accessed freely. The website owners state that we may print and download material from the sites solely for non-commercial use provided that we agree not to change or delete any copyright or proprietary notices from the materials. Therefore, in addition to the dataset itself, we also provide the according copyright statements of every website and an URL link to the original page for each riddle. The dataset users must sign an informed consent form that they will only use our dataset for research purposes before they can access the both the riddles and our annotations.
References


Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, A. Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. ArXiv, abs/1803.05457.


Edward Hirsch. 2014. A poet’s glossary. HMH.


