Answering Elementary Science Questions by Constructing Coherent Scenes using Background Knowledge

Yang Li*
UC Santa Barbara
Santa Barbara, CA, USA
yangli@cs.ucsb.edu

Peter Clark
Allen Institute for Artificial Intelligence
Seattle, WA, USA
peterc@allenai.org

Abstract

Much of what we understand from text is not explicitly stated. Rather, the reader uses his/her knowledge to fill in gaps and create a coherent, mental picture or "scene" depicting what text appears to convey. The scene constitutes an understanding of the text, and can be used to answer questions that go beyond the text.

Our goal is to answer elementary science questions, where this requirement is pervasive; A question will often give a partial description of a scene and ask the student about implicit information. We show that by using a simple "knowledge graph" representation of the question, we can leverage several large-scale linguistic resources to provide missing background knowledge, somewhat alleviating the knowledge bottleneck in previous approaches. The coherence of the best resulting scene, built from a question/answer-candidate pair, reflects the confidence that the answer candidate is correct, and thus can be used to answer multiple choice questions. Our experiments show that this approach outperforms competitive algorithms on several datasets tested. The significance of this work is thus to show that a simple "knowledge graph" representation allows a version of "interpretation as scene construction" to be made viable.

1 Introduction

Elementary grade science tests are challenging as they test a wide variety of commonsense knowledge that human beings largely take for granted, yet are very difficult for machines (Clark, 2015). For example, consider a question from a NY Regents 4th Grade science test:

Question 1 *"When a baby shakes a rattle, it makes a noise. Which form of energy was changed to sound energy?" [Answer: mechanical energy]*

Science questions are typically quite different from the entity-centric factoid questions extensively studied in the question answering (QA) community, e.g., "In which year was Bill Clinton born?" (Ferrucci et al., 2010; Yao and Van Durme, 2014). While factoid questions are usually answerable from text search or fact databases, science questions typically require deeper analysis. A full understanding of the above question involves not just parsing and semantic interpretation; it involves adding implicit information to create an overall picture of the "scene" that the text is intended to convey, including facts such as: noise is a kind of sound, the baby is holding the rattle, shaking involves movement, the rattle is making the noise, movement involves mechanical energy, etc. This mental ability to create a scene from partial information is at the heart of natural language understanding (NLU), which is essential for answering these kinds of question. It is also very difficult for a machine because it requires substantial world knowledge, and there are often many ways a scene can be elaborated.

We present a method for answering multiple-choice questions that implements a simple version of this. A scene is represented as a "knowledge graph" of nodes (words) and relations, and the scene is elaborated with (node,relation,node) tuples drawn from three large-scale linguistic knowledge resources: WordNet (Miller, 1995), DART (Clark and Harrison, 2009), and the Free-Association database (Nelson et al., 2004). These elaborations reflect the mental process of "filling in the gaps", and multiple choice questions can then be answered by finding which answer option creates the most coherent scene.

The notion of NLU as constructing a most coherent scene is not new, and has has been studied in several contexts including work on scripts (Schank and Abelson, 1977), interpretation as ab-

*Work was done while the author was an intern at Allen Institute for Artificial Intelligence.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2007–2012,
Lisbon, Portugal, 17-21 September 2015.

duction (Hobbs et al., 1988; Hobbs, 1979; Ovchinnikova et al., 2014), bridging anaphora (Asher and Lascarides, 1998; Fan et al., 2005), and paragraph understanding (Zadrozny and Jensen, 1991; Harabagiu and Moldovan, 1997). These methods are inspiring, but have previously been limited by the lack of background knowledge to supply implicit information, and with the complexity of their representations. To make progress, we have chosen to work with a simple "knowledge graph" representation of nodes (words) and edges (relations). Although we lose some subtlety of expression, we gain the ability to leverage several vast resources of world knowledge to supply implicit information. The significance of this work is thus to show that, by working with a simple "knowledge graph" representation, we can make a viable version of "interpretation as scene construction". Although the approach makes several simplifying assumptions, our experiments show that it outperforms competitive algorithms on several datasets of (real) elementary science questions.

2 Approach

The input to our question-answering system is a multiple choice question Q, a set of answer options a_k, and one or more background knowledge base(s) each containing a set of $(word_i, relation, word_j)$ tuples, each denoting that $word_1$ is plausibly related to $word_j$ by $relation$. The output is a ranked list of the K answer options.

We define a *scene* S as a "knowledge graph" of *nodes* (words) and *edges* (relations between words), where all $(word_i, relation, word_j)$ edges are sanctioned by (contained in) at least one of the background knowledge bases. Each scene node has an associated measure of $coherence$ (described shortly), denoting how well-connected it is. The question-answering objective is, for each answer option a_k, to find the most coherent scene containing (at least) the question keywords $kw_i \in Q$ and answer option a_k, and then return the answer option with the overall highest coherence score. Our implementation approximates this objective using a simple elaborate-and-prune algorithm, illustrated in Figure 1[1] and now described.

[1]The system constructs 4 alternative graphs, each contains only one answer option plus some additional related nodes. Figure 1 shows just one of these 4 graphs, namely the graph containing answer option "food".

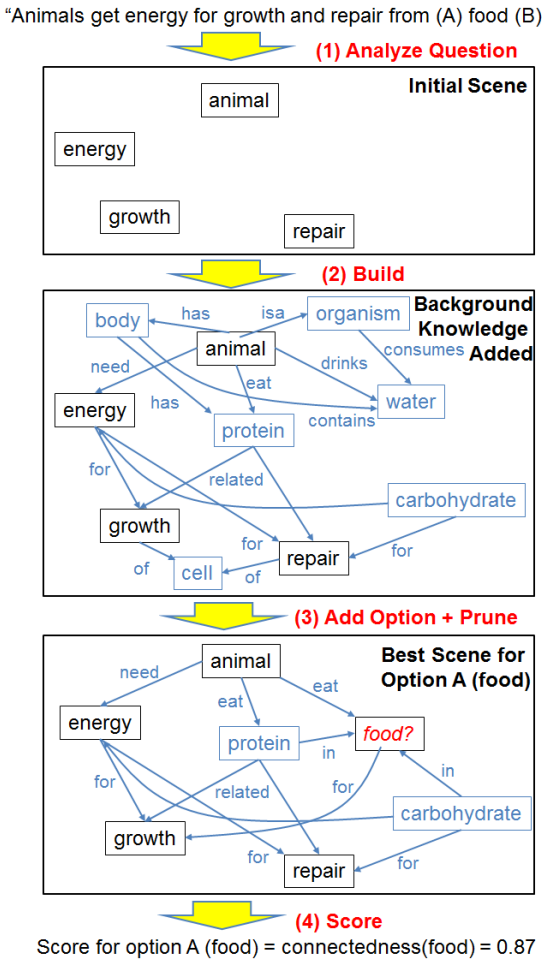

Figure 1: (1) Question keywords are extracted to form the initial scene. (2) The scene is elaborated with background knowledge to add plausible relationships. (3) For each answer option, it is added into the scene and connected with additional relationships. Then the scene is pruned. (4) A score is derived from the final scene, reflecting confidence that the answer option is correct.

2.1 Question Analyzer

The initial scene is simply the keywords (non-stop words) $KW = \{kw_i\}$ in the question Q, along with a measure of importance $IS(kw_i)$ for each word kw_i. For our purposes we compute importance by sending the question to Google, grouping the top 20 result snippets into a document d, and computing:

$$IS(kw) = \frac{tf_d(kw)}{df_Q(kw)}, \quad (1)$$

where $tf_d(kw)$ is the term frequency of kw in document d, and $df_Q(kw)$ is the document frequency of kw in question set Q containing all the available elementary science questions. The intuition here is

KB	Size (# tuples)	Examples
WordNet	235k	(dog,isa,animal) (sunlight,isa,energy)
DART	2.3M	(nutrients,in,food) (animal,eat,food)
FreeAssoc	64k	(car,relate_to,tire) (ice,relate_to,cold)

Table 1: Knowledge Bases Used

that the words frequently mentioned in variations of the question should be important (reflected by "tf"), while the descriptive words (e.g. "following", "example") which are widely used in many questions should be penalized (reflected by "idf"). Other methods could equally be used to compute importance.

2.2 Builder

In this step our goal is to inject implicit knowledge from the background KBs to form an elaborated knowledge graph. To do this, we first fetch all the triples $(kw, relation, w)$ that are directly connected with any keyword $kw \in KW$ from the background KBs, as well as all $(kw_i, relation, kw_j)$ triples between keywords. In our experiments we use three background knowledge bases to supply implicit knowledge, although in principle any triple store could be used: WordNet (Miller, 1995), DART (Clark and Harrison, 2009), and the FreeAssociation database (Nelson et al., 2004). Table 1 shows examples of each [2].

These triples introduce new nodes w into the graph. As we may get a large number of such nodes, we score them and retain only the top scoring ones (and all edges connecting to it). Informally, a new word w is preferred if it is connected to many important keywords kw with strong relationships. Formally, the scoring function is:

$$score(w) = \sum_{kw \in K} IS(kw) * rel(kw, w) \quad (2)$$

where $IS(kw)$ is the importance score of keyword kw and $rel(kw, w)$ is the relatedness score between kw and w. In this work we use the cosine similarity between word2vec (Mikolov et al., 2013) word vectors to measure two words' relatedness, as a rough proxy for the strength of relatedness in the KBs (the KBs themselves do not provide meaningful strengths of relationship). After the ranking, the top $N \times |KW|$ neighbor words w are retained[3], along with their edges to keywords kw and each other.

Note that at this point the elaboration process is independent of any answer option; rather, the graph depicts the question scenario.

2.3 Elaborate and Prune

To score the K different answer options, the system now builds K alternative elaborations of the scene so far, each one with answer option a_k added, and assesses the coherence of the addition. The answer option a_k that fits "most coherently" with the scene is returned as the answer to the question.

To do this for a given option a_k, we add a_k to the graph along with all triples $(w_i, relation, a_k)$ in the KBs that relate any node w_i in the graph to a_k. Now that the focus a_k of the question is known, some of the earlier added nodes w in the graph may be only weakly relevant to the question and answer, and so we add a pruning step to remove these nodes. The goal of this pruning is to find a dense subgraph (i.e. the coherent scene) that would ideally contain all the question keywords kw, the answer option a_k, and extra words w_k that are highly connected with them.

Inspired by Sozio et al's work (Sozio and Gionis, 2010) on finding strongly interconnected subgroups in social networks, we have developed an iterative node removal algorithm for extracting this subgraph. We define the coherence of a node as the summed weight of its incident edges:

$$coherence(w) = \sum_{w' \in \{(w,r,w')\}} rel(w, w') \quad (3)$$

where $rel(w, w')$ is the weight of edge (w, r, w') in the graph, again computed using cosine similarity between w and w'. We then iteratively remove the non-keyword node (and attached edges) with least coherence until the answer option a_k is about to removed. The resulting graph is thus maximally pruned, subject to the constraint it must still describe the question plus answer option.

Finally, we use the coherence of the answer option a_k in this final scene as the confidence that a_k is the correct answer. The system repeats this

[2] WordNet: all relationships types are used. DART: The NVN and NPN databases with frequency counts > 10 are used. FreeAssoc: The top 3 associations per word were used.

[3] The optimal N (here 6) was selected using an independent set of training questions

for all K answer options and selects the a_k with highest confidence.

3 Evaluation

The system was developed using a dataset of natural (unedited) elementary science exam questions, and then tested on three similar, unseen (hidden) datasets. Its performance was compared with two other state-of-the-art systems for this task. As our system only fields questions where the answer options are all single words, we evaluate it, and the other systems, only on these subsets. These subsets are in general easier than other questions, but this advantage is the same for all systems being compared so it is still a fair test.

3.1 Evaluation Datasets

The datasets used are the non-diagram, multiple-choice questions with single-word answer options drawn from the following exams:

- **Dev (System Development):** New York Regents 4th Grade Science [4] (47 questions in 6 years)
- **Test1:** New York Regents 4th Grade Science (23 questions in 3 years)
- **Test2:** Additional 4th Grade Science (from multiple States) (26 questions)
- **Test3:** 5th Grade Science (from multiple States) (197 questions)

Although these datasets are small (real exam questions of this type are in limited supply), the numbers are large enough to draw conclusions.

3.2 Experiments

We compared our system (called **SceneQA**) with two other state-of-the-art systems for this task:

- **LSModel** (Lexical semantics model): SVM combination of several language models (likelihood of answer given question) and information retrieval scores (score of top retrieved sentence matching question plus answer), trained on a set of questions plus answers. (An expanded version of Section 4.3 of (Jansen et al., 2014))
- **A*Rules**: "Prove" the answer option from the question by applying lexical inference rules automatically extracted from science texts. Select the option with the strongest "proof". (Clark et al., 2014)

[4] http://www.nysedregents.org/Grade4/Science/home.html

	Dev	Test1	Test2	Test3
LSModel	65.96	58.70	28.85	30.08
A*Rules	65.96	67.00	47.00	29.22
SceneQA	**83.51**	66.30	**65.38**	**55.20**

Table 2: SceneQA outperforms two competitive systems on two of the three test sets. The highlighted improvements are statistically significant.

The results (% scores, Table 2) show SceneQA significantly outperforms the two other systems on two of the three test sets, including the largest (Test3, 197 questions), suggesting the approach has merit.

We also performed some case studies to identify what kinds of questions SceneQA does well on, relative to the baselines. In general, SceneQA performs well when the question words and the (correct) answer can be tightly related by background knowledge, including through intermediate nodes (words). For example, in Question 2 below:

Question 2 *Which type of energy does a person use to pedal a bicycle? (A) light (B) sound (C) mechanical (D) electrical*

the KB relates the correct answer "mechanical" to the question words "energy", "pedal", "bicycle", and the intermediate node "power" forming a tight graph. In contrast, the other algorithms select the wrong answer "light" due to frequent mentions of "bicycle lights" in their supporting text corpora that confuses their algorithms.

3.3 Ablations

We also performed ablations to assess which parts of our method are contributing the most:

- **-NewNodes**: Only add edges but no new nodes w during the Build step.
- **-Prune**: Do not prune nodes during the Elaborate and Prune step.
- **-Both**: No new nodes, no pruning

	Dev	Test1	Test2	Test3
SceneQA	**83.51**	66.30	**65.38**	**55.20**
-NewNodes	65.96	69.57	42.31	51.78
-Prune	70.74	57.61	47.12	50.13
-Both	59.57	65.22	42.31	50.25

Table 3: SceneQA outperforms all the ablations on two of the three test sets. The highlighted improvements are statistically significant.

The results (% scores, Table 3) suggest that the two most important algorithmic features - adding

concepts implied but not explicitly stated in the text (NewNodes), and later removing implied information that is of low relevance to the answer (Prune) - are important for answering the questions correctly. (The small gain without adding NewNodes on Test1 is not statistically significant).

3.4 Error Analysis

We also examined cases where SceneQA gave the wrong answer. Two problems were particularly common:

(1) There were two answer options with opposite meanings, and one of them was correct. For example:

Question 3 *An animal that has a backbone is called a(n) (A) invertebrate (B) vertebrate (C) exoskeleton (D) sponge*

Since the relatedness measure we use (i.e. word2vec) cannot distinguish words with similar distributional semantics (a common property of antonyms), our method cannot confidently identify which of the opposites (e.g., here, vertebrate vs. invertebrate) is correct.

(2) The word ordering in the question is particularly important, e.g., questions about processes or sequences. For example:

Question 4 *The process that changes a gas to liquid is called (A) condensation (B) melting (C) evaporation (D) vaporization*

Because our method ignores word order (the knowledge graph is initially populated with keywords in the question), the representation is inherently incapable of capturing sequential information (e.g., here, gas to liquid vs. liquid to gas). As a result, it struggles with such questions.

4 Discussion and Conclusion

Our goal is to answer simple science questions. Unlike entity-centric factoid QA tasks, science questions typically involve general concepts, and answering them requires identifying implicit relationships in the question. Our approach is to view question-answering as constructing a coherent scene. While the notion of scene construction is not new, our insight is that this can be done with a simple "knowledge graph" representation, allowing several massive background KBs to be applied, somewhat alleviating the knowledge bottleneck. Our contribution is to show this works well in the elementary science domain.

Despite this, there are clearly many limitations with our approach: we are largely ignoring syntactic structure in the questions; the KBs are noisy, contributing errors to the scenes; the graph representation has limited expressivity (e.g., no quantification or negation); the word2vec measure of relationship strength does not account for the question context; and contradictions are not detected within the scene. These all contributed to QA failures in the tests. However, the approach is appealing as it takes a step towards a richer picture of language understanding, the empirical results are encouraging, and there are many ways these initial limitations can be addressed going forward. We are confident that this is a rich and exciting space, worthy of further exploration.

Acknowledgments

The research was supported by the Allen Institute for Artificial Intelligence (AI2). We thank the Aristo team at AI2 for invaluable discussions, and the anonymous reviewers for helpful comments.

References

Nicholas Asher and Alex Lascarides. 1998. Bridging. *Journal of Semantics*, 15(1):83–113.

Peter Clark and Phil Harrison. 2009. Large-scale extraction and use of knowledge from text. In *Proceedings of the fifth international conference on Knowledge capture*, pages 153–160.

Peter Clark, Niranjan Balasubramanian, Sumithra Bhakthavatsalam, Kevin Humphreys, Jesse Kinkead, Ashish Sabharwal, and Oyvind Tafjord. 2014. Automatic construction of inference-supporting knowledge bases. In *Proceedings of AKBC*.

Peter Clark. 2015. Elementary school science and math tests as a driver for ai: Take the aristo challenge! In *Twenty-Seventh IAAI Conference*.

James Fan, Ken Barker, and Bruce Porter. 2005. Indirect anaphora resolution as semantic path search. In *Proceedings of the 3rd international conference on Knowledge capture*, pages 153–160. ACM.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.

Sanda M Harabagiu and Dan I Moldovan. 1997. Textnet–a text-based intelligent system. *Natural Language Engineering*, 3(02):171–190.

Jerry R Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. 1988. Interpretation as abduction. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, pages 95–103. Association for Computational Linguistics.

Jerry R Hobbs. 1979. Coherence and coreference. *Cognitive science*, 3(1):67–90.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of ACL*, pages 977–986.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407.

Ekaterina Ovchinnikova, Niloofar Montazeri, Theodore Alexandrov, Jerry R Hobbs, Michael C McCord, and Rutu Mulkar-Mehta. 2014. Abductive reasoning with a large knowledge base for discourse processing. In *Computing Meaning*, pages 107–127. Springer.

Roger C Schank and Robert P Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Erlbaum, Hillsdale, NJ.

Mauro Sozio and Aristides Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *Proceedings of SIGKDD*, pages 939–948.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*.

Wlodek Zadrozny and Karen Jensen. 1991. Semantics of paragraphs. *Computational Linguistics*, 17(2):171–209.