From ‘F’ to ‘A’ on the N.Y. Regents Science Exams: An Overview of the Aristo Project[*]

Peter Clark, Oren Etzioni, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin

Allen Institute for Artificial Intelligence, Seattle, WA, U.S.A.

Abstract

AI has achieved remarkable mastery over games such as Chess, Go, and Poker, and even *Jeopardy!*, but the rich variety of standardized exams has remained a landmark challenge. Even in 2016, the best AI system achieved merely 59.3% on an 8th Grade science exam challenge (Schoenick et al., 2016).

This paper reports unprecedented success on the Grade 8 New York Regents Science Exam, where for the first time a system scores more than 90% on the exam’s non-diagram, multiple choice (NDMC) questions. In addition, our Aristo system, building upon the success of recent language models, exceeded 83% on the corresponding Grade 12 Science Exam NDMC questions. The results, on unseen test questions, are robust across different test years and different variations of this kind of test. They demonstrate that modern NLP methods can result in mastery on this task. While not a full solution to general question-answering (the questions are multiple choice, and the domain is restricted to 8th Grade science), it represents a significant milestone for the field.

1 Introduction

This paper reports on the history, progress, and lessons from the Aristo project, a six-year quest to answer grade-school and high-school science exams. Aristo has recently surpassed 90% on multiple choice questions from the 8th Grade New York Regents Science Exam (see Figure 2).[1] We begin by offering several perspectives on why this achievement is significant for NLP and for AI more broadly.

1.1 The Turing Test versus Standardized Tests

In 1950, Alan Turing proposed the now well-known Turing Test as a possible test of machine intelligence: If a system can exhibit conversational behavior that is indistinguishable from that of a human during a conversation, that system could be considered intelligent (Turing, 1950). As the field of AI has grown, the test has become less meaningful as a challenge task for several reasons. First, its setup is not well defined (e.g., who is the person giving the test?). A computer scientist would likely know good distinguishing questions to ask, while a random member of the general public may not. What constraints are there on the interaction? What guidelines are provided to the judges? Second, recent Turing Test competitions have shown that, in certain formulations, the test itself is gameable; that is, people can be fooled by systems that simply retrieve sentences and make no claim of being intelligent (Aron, 2011; BBC, 2014). John Markoff of The New York Times wrote that the Turing Test is more a test of human gullibility than machine intelligence. Finally, the test, as originally conceived, is pass/fail rather than scored, thus providing no measure of progress toward a goal, something essential for any challenge problem.

Instead of a binary pass/fail, machine intelligence is more appropriately viewed as a diverse collection of capabilities associated with intelligent behavior. Finding appropriate benchmarks to test such capabilities is challenging; ideally, a benchmark should test a variety of capabilities in a natural and unconstrained way, while additionally being clearly measurable, understandable, accessible, and motivating.

Standardized tests, in particular science exams, are a rare example of a challenge that meets these requirements. While not a full test of machine intelligence, they do explore several capabilities strongly associated with intelligence, including language understanding, reasoning, and use of common-sense knowledge. One of the most interesting and appealing aspects of science exams is their graduated and multifaceted nature; different questions explore different types of knowledge, varying substantially in difficulty. For this reason, they have been used as a compelling—and challenging—task for the field for many years (Brachman et al., 2005; Clark and Etzioni, 2016).

1.2 Natural Language Processing

With the advent of contextualized word-embedding methods such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2018), and most recently RoBERTa (Liu et al., 2019b), the NLP community’s benchmarks are being felled at a remarkable rate. These are, however, internally-generated yardsticks, such as SQuAD (Rajpurkar et al., 2016), Glue (Wang et al., 2019), SWAG (Zellers et al., 2018), TriviaQA (Joshi et al., 2017), and many others.

In contrast, the 8th Grade science benchmark is an external, independently-generated benchmark where we can compare machine performance with human performance. Moreover, the breadth of the vocabulary and the depth of

[*] We gratefully acknowledge the late Paul Allen’s inspiration, passion, and support for research on this grand challenge.

[1] See Section 4.1 for the experimental methodology.

1. Which equipment will best separate a mixture of iron filings and black pepper? (1) magnet (2) filter paper (3) triple-beam balance (4) voltmeter
2. Which form of energy is produced when a rubber band vibrates? (1) chemical (2) light (3) electrical (4) sound
3. Because copper is a metal, it is (1) liquid at room temperature (2) nonreactive with other substances (3) a poor conductor of electricity (4) a good conductor of heat
4. Which process in an apple tree primarily results from cell division? (1) growth (2) photosynthesis (3) gas exchange (4) waste removal

Figure 1: Example questions from the NY Regents Exam (8th Grade), illustrating the need for both scientific and commonsense knowledge.

the questions is unprecedented. For example, in the ARC question corpus of science questions, the average question length is 22 words using a vocabulary of over 6300 distinct (stemmed) words (Clark et al., 2018). Finally, the questions often test scientific knowledge by applying it to everyday situations and thus require aspects of common sense. For example, consider the question: *Which equipment will best separate a mixture of iron filings and black pepper?* To answer this kind of question robustly, it is not sufficient to understand magnetism. Aristo also needs to have some model of "black pepper" and "mixture" because the answer would be different if the iron filings were submerged in a bottle of water. Aristo thus serves as a unique "poster child" for the remarkable and rapid advances achieved by leveraging contextual word-embedding models in, NLP.

1.3 Machine Understanding of Textbooks

Within NLP, machine understanding of textbooks is a grand AI challenge that dates back to the '70s, and was reinvigorated in Raj Reddy's 1988 AAAI Presidential Address and subsequent writing (Reddy, 1988, 2003). However, progress on this challenge has a checkered history. Early attempts side-stepped the natural language understanding (NLU) task, in the belief that the main challenge lay in problem-solving. For example, Larkin et al. (1980) manually encoded a physics textbook chapter as a set of rules that could then be used for question answering. Subsequent attempts to automate the reading task were unsuccessful, and the language task itself has emerged as a major challenge for AI.

In recent years there has been substantial progress in systems that can find factual answers in text, starting with IBM's Watson system (Ferrucci et al., 2010), and now with high-performing neural systems that can answer short questions provided they are given a text that contains the answer (e.g., Seo et al., 2016; Wang et al., 2018). The work presented here continues along this trajectory, but aims to also answer questions where the answer may not be written down explicitly. While not a full solution to the textbook grand challenge, this work is thus a further step along this path.

2 A Brief History of Aristo

Project Aristo emerged from the late Paul Allen's long-standing dream of a Digital Aristotle, an "easy-to-use, all-encompassing knowledge storehouse...to advance the field of AI." (Allen, 2012). Initially, a small pilot program in 2003 aimed to encode 70 pages of a chemistry textbook and answer the questions at the end of the chapter. The pilot was considered successful (Friedland et al., 2004), with the significant caveat that both text and questions were manually encoded, side-stepping the natural language task, similar to earlier efforts. A subsequent larger program, called Project Halo, developed tools allowing domain experts to rapidly enter knowledge into the system. However, despite substantial progress (Gunning et al., 2010; Chaudhri et al., 2013), the project was ultimately unable to scale to reliably acquire textbook knowledge, and was unable to handle questions expressed in full natural language.

In 2013, with the creation of the Allen Institute for Artificial Intelligence (AI2), the project was rethought and relaunched as Project Aristo (connoting Aristotle as a child), designed to avoid earlier mistakes. In particular: handling natural language became a central focus; Most knowledge was to be acquired automatically (not manually); Machine learning was to play a central role; questions were to be answered exactly as written; and the project restarted at elementary-level science (rather than college-level) (Clark et al., 2013).

The metric progress of the Aristo system on the Regents 8th Grade exams (non-diagram, multiple choice part, for a hidden, held-out test set) is shown in Figure 2. The figure shows the variety of techniques attempted, and mirrors the rapidly changing trajectory of the Natural Language Processing (NLP) field in general. Early work was dominated by information retrieval, statistical, and automated rule extraction and reasoning methods (Clark et al., 2014, 2016; Khashabi et al., 2016; Khot et al., 2017; Khashabi et al., 2018). Later work has harnessed state-of-the-art tools for large-scale language modeling and deep learning (Trivedi et al., 2019; Tandon et al., 2018), which have come to dominate the performance of the overall system and reflects the stunning progress of the field of NLP as a whole.

3 The Aristo System

We now describe the architecture of Aristo, and provide a brief summary of the solvers it uses.

3.1 Overview

The current configuration of Aristo comprises of eight solvers, described shortly, each of which attempts to answer a multiple choice question. To study particular phenomena and develop solvers, the project has created larger datasets to amplify and study different problems, resulting in 10 new

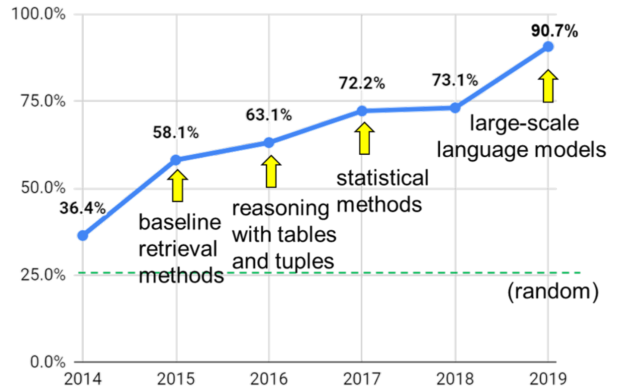


Figure 2: Aristo's scores on Regents 8th Grade Science (non-diagram, multiple choice) over time (held-out test set).

datasets[2] and 5 large knowledge resources[3] for the community.

The solvers can be loosely grouped into:

1. Statistical and information retrieval methods
2. Reasoning methods
3. Large-scale language model methods

Over the life of the project, the relative importance of the methods has shifted towards large-scale language methods.

Several methods make use of the Aristo Corpus, comprising a large Web-crawled corpus (5×10^{10} tokens (280GB)) originally from the University of Waterloo, combined with targeted science content from Wikipedia, SimpleWikipedia, and several smaller online science texts (Clark et al., 2016).

3.2 Information Retrieval and Statistics

Three solvers use information retrieval (IR) and statistical measures to select answers. These methods are particularly effective for "lookup" questions where an answer is explicitly stated in the Aristo corpus.

The **IR solver** searches to see if the question along with an answer option is explicitly stated in the corpus, and returns the confidence that such a statement was found. To do this, for each answer option a_i, it sends $q + a_i$ as a query to a search engine (we use ElasticSearch), and returns the search engines score for the top retrieved sentence s, where s also has at least one non-stopword overlap with q, and at least one with a_i. This ensures s has some relevance to both q and a_i. This is repeated for all options a_i to score them all, and the option with the highest score selected. Further details are available in (Clark et al., 2016).

The **PMI solver** uses pointwise mutual information (Church and Hanks, 1989) to measure the strength of the associations between parts of q and parts of a_i. Given a large corpus C, PMI for two n-grams x and y is defined as $\mathrm{PMI}(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$. Here $p(x, y)$ is the joint probability that x and y occur together in C, within a certain window of text (we use a 10 word window). The term $p(x)p(y)$, on the other hand, represents the probability with which x and y would occur together if they were statistically independent. The ratio of $p(x, y)$ to $p(x)p(y)$ is thus the ratio of the observed co-occurrence to the expected co-occurrence. The larger this ratio, the stronger the association between x and y. The solver extracts unigrams, bigrams, trigrams, and skip-bigrams from the question q and each answer option a_i. It outputs the answer with the largest average PMI, calculated over all pairs of question n-grams and answer option n-grams. Further details are available in (Clark et al., 2016).

Finally, **ACME** (Abstract-Concrete Mapping Engine) searches for a cohesive link between a question q and candidate answer a_i using a large knowledge base of *vector spaces* that relate words in language to a set of 5000 scientific terms enumerated in a *term bank*. ACME uses three types of vector spaces: terminology space, word space, and sentence space. Terminology space is designed for finding a term in the term bank that links a question to a candidate answer with strong lexical cohesion. Word space is designed to characterize a word by the context in which the word appears. Sentence space is designed to characterize a sentence by the words that it contains. The key insight in ACME is that we can better assess lexical cohesion of a question and answer by pivoting through scientific terminology, rather than by simple co-occurence frequencies of question and answer words. Further details are provided in (Turney, 2017).

These solvers together are particularly good at "lookup" questions where an answer is explicitly written down in the Aristo Corpus. For example, they correctly answer:

> *Infections may be caused by (1) mutations (2) microorganisms* **[correct]** *(3) toxic substances (4) climate changes*

as the corpus contains the sentence "Products contaminated with microorganisms may cause infection." (for the IR solver), as well as many other sentences mentioning both "infection" and "microorganisms" together (hence they are highly correlated, for the PMI solver), and both words are strongly correlated with the term "microorganism" (ACME).

3.3 Reasoning Methods

The **TupleInference solver** uses semi-structured knowledge in the form of *tuples*, extracted via Open Information Extraction (Open IE) (Banko et al., 2007). Two sources of tuples are used:

- A knowledge base of 263k tuples (T), extracted from the Aristo Corpus plus several domain-targeted sources, using training questions to retrieve science-relevant information.

[2] Datasets ARC, OBQA, SciTail, ProPara, QASC, WIQA, QuaRel, QuaRTz, PerturbedQns, and SciQ. Available at https://allenai.org/data/data-aristo-all.html

[3] The ARC Corpus, the AristoMini corpus, the TupleKB, the TupleInfKB, and Aristo's Tablestore. Available at https://allenai.org/data/data-aristo-all.html

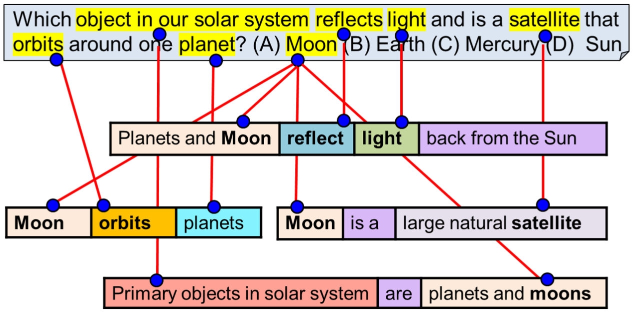

Which object in our solar system reflects light and is a satellite that orbits around one planet? (A) Moon (B) Earth (C) Mercury (D) Sun
Planets and Moon
reflect
light
back from the Sun
Moon
orbits
planets
Moon
is a
large natural satellite
Primary objects in solar system
are
planets and moons

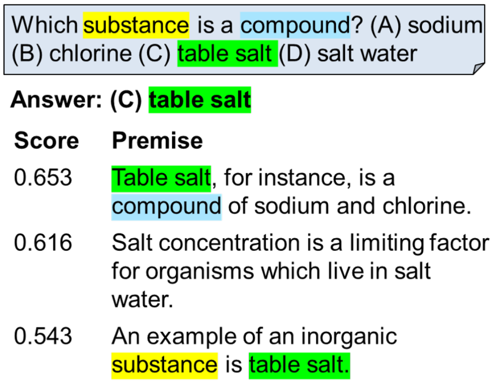
Figure 3: The Tuple Inference Solver retrieves tuples relevant to the question, and constructs a support graph for each answer option. Here, the support graph for the choice "(A) Moon" is shown. The tuple facts "...Moon reflect light...", "...Moon is a ...satellite", and "Moon orbits planets" all support this answer, addressing different parts of the question. This support graph is scored highest, hence option "(A) Moon" is chosen.

- On-the-fly tuples (T'), extracted at question-answering time from t¡he same corpus, to handle questions from new domains not covered by the training set.

TupleInference treats the reasoning task as searching for a graph that best connects the terms in the question (qterms) with an answer choice via the knowledge; see Figure 3 for a simple illustrative example. Unlike standard alignment models used for tasks such as Recognizing Textual Entailment (RTE) (Dagan et al., 2010), however, we must score alignments between the tuples retrieved from the two sources above, $T_{qa} \cup T'_{qa}$, and a (potentially multi-sentence) multiple choice question qa.

The qterms, answer choices, and tuples fields (i.e. subject, predicate, objects) form the set of possible vertices, $\mathcal{V}$, of the support graph. Edges connecting qterms to tuple fields and tuple fields to answer choices form the set of possible edges, $\mathcal{E}$. The support graph, $G(V, E)$, is a subgraph of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where V and E denote "active" nodes and edges, respectively. We define an ILP optimization model to search for the best support graph (i.e., the active nodes and edges), where a set of constraints define the structure of a valid support graph (e.g., an edge must connect an answer choice to a tuple) and the objective defines the preferred properties (e.g. active edges should have high word-overlap). Details of the constraints are given in (Khot et al., 2017). We then use the SCIP ILP optimization engine (Achterberg, 2009) to solve the ILP model. To obtain the score for each answer choice a_i, we force the node for that choice x_{a_i} to be active and use the objective function value of the ILP model as the score. The answer choice with the highest score is selected. Further details are available in (Khot et al., 2017).

Multee (Trivedi et al., 2019) is a solver that repurposes existing *textual entailment* tools for question answering. Textual entailment (TE) is the task of assessing if one text implies another, and there are several high-performing TE systems now available. However, question answering often requires reasoning over *multiple* texts, and so Multee learns to reason with multiple individual entailment decisions. Specifically, Multee contains two components: (i) a **sentence relevance model**, which learns to focus on the relevant sentences, and (ii) a **multi-layer aggregator**, which uses an entailment model to obtain multiple layers of question-relevant representations for the premises and then composes them using the sentence-level scores from the relevance model. Finding relevant sentences is a form of local entailment between each premise and the answer hypothesis, whereas aggregating question-relevant representations is a form of global entailment between all premises and the answer hypothesis. This means we can effectively repurpose the same pre-trained entailment function f_e for both components. Details of how this is done are given in (Trivedi et al., 2019). An example of a typical question and scored, retrieved evidence is shown in Figure 4. Further details are available in (Trivedi et al., 2019).



Which substance is a compound? (A) sodium (B) chlorine (C) table salt (D) salt water

Answer: (C) table salt

Score	**Premise**
0.653	Table salt, for instance, is a compound of sodium and chlorine.
0.616	Salt concentration is a limiting factor for organisms which live in salt water.
0.543	An example of an inorganic substance is table salt.

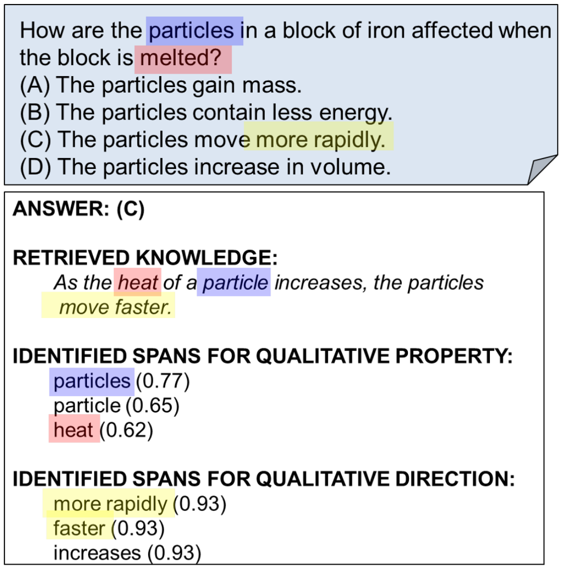
Figure 4: Multee retrieves potentially relevant sentences, then for each answer option in turn, assesses the degree to which each sentence entails that answer. A multi-layered aggregator then combines this (weighted) evidence from each sentence. In this case, the strongest overall support is found for option "(C) table salt", so it is selected.

The **QR (qualitative reasoning) solver** is designed to answer questions about qualitative influence, i.e., how more/less of one quantity affects another (see Figure 5). Unlike the other solvers in Aristo, it is a specialist solver that only fires for a small subset of questions that ask about qualitative change, identified using (regex) language patterns.

The solver uses a knowledge base K of 50,000 (textual) statements about qualitative influence, e.g., "A sunscreen with a higher SPF protects the skin longer.", extracted automatically from a large corpus. It has then been trained to apply such statements to qualitative questions, e.g.,

> *John was looking at sunscreen at the retail store. He noticed that sunscreens that had lower SPF would offer protection that is (A) Longer (B) Shorter* **[correct]**

In particular, the system learns through training to track the *polarity* of influences: For example, if we were to change "lower" to "higher" in the above example, the system will change its answer choice. Another example is shown in Figure 5. Again, if "melted" were changed to "cooled", the



How are the particles in a block of iron affected when the block is melted?
(A) The particles gain mass.
(B) The particles contain less energy.
(C) The particles move more rapidly.
(D) The particles increase in volume.

ANSWER: (C)

RETRIEVED KNOWLEDGE:
As the heat of a particle increases, the particles move faster.

IDENTIFIED SPANS FOR QUALITATIVE PROPERTY:
particles (0.77)
particle (0.65)
heat (0.62)

IDENTIFIED SPANS FOR QUALITATIVE DIRECTION:
more rapidly (0.93)
faster (0.93)
increases (0.93)

Figure 5: Given a question about a qualitative relationship (How does one increase/decrease affect another?), the qualitative reasoning solver retrieves a relevant qualitative rule from a large database. It then assesses which answer option is best implied by that rule. In this case, as the rule states more heat implies faster movement, option "(C)... move more rapidly" is scored highest and selected, including recognizing that "heat" and "melted", and "faster" and "more rapidly" align.

system would change its choice to "(B) less energy".

The QR solver learns to reason using the BERT language model (Devlin et al., 2018), using the approach described in Section 3.4 below. It is fine-tuned on 3800 crowdsourced qualitative questions illustrating the kinds of manipulation required, along with the associated qualitative knowledge sentence. The resulting system is able to answer questions that include significant linguistic and knowledge gaps between the question and retrieved knowledge (Table 1).

Because the number of qualitative questions is small in our dataset, the solver does not significantly change Aristo's performance, although it does provide an explanation for its answers. For this reason we omit it in the results later. Further details and a detailed separate evaluation is available in (Tafjord et al., 2019).

3.4 Large-Scale Language models

The field of NLP has advanced substantially with the advent of large-scale language models such as ELMo (Peters et al., 2018), ULMFit (Howard and Ruder, 2018), GPT (Radford et al., 2018), BERT (Devlin et al., 2018), and RoBERTa (Liu et al., 2019b). These models are trained to perform various language prediction tasks such as predicting a missing word or the next sentence, using large amounts of text (e.g., BERT was trained on Wikipedia + the Google Book Corpus of 10,000 books). They can also be fine-tuned to new language prediction tasks, such as question-answering, and have been remarkably successful in the few months that they have been available.

<u>Comparatives:</u>
"warmer" ↔ "increase temperature"
"more difficult" ↔ "slower"
"need more time" ↔ "have lesser amount"
"decreased distance" ↔ "hugged"
"cost increases" ↔ "more costly"
"increase mass" ↔ "add extra"
"more tightly packed" ↔ "add more"

<u>Commonsense Knowledge:</u>
"more land development" ↔ "city grow larger"
"not moving" ↔ "sits on the sidelines"
"caught early" ↔ 'sooner treated"
"lets more light in" ↔ "get a better picture"
"stronger electrostatic force" ↔ "hairs stand up more"
"less air pressure" ↔ "more difficult to breathe"
"more photosynthesis" ↔ "increase sunlight"

<u>Discrete Values:</u>
"stronger acid" ↔ "vinegar" vs. "tap water"
"more energy" ↔ "ripple" vs. "tidal wave"
"closer to Earth" ↔ "ball on Earth" vs. "ball in space"
"mass" ↔ "baseball" vs. "basketball"
"rougher" ↔ "notebook paper" vs. "sandpaper"
"heavier" ↔ "small wagon" vs. "eighteen wheeler"

Table 1: Examples of linguistic and semantic gaps between knowledge K_i (left) and question Q_i (right) that need to be bridged for answering qualitative questions.

We apply BERT to multiple choice questions by treating the task as classification: Given a question q with answer options a_i and optional background knowledge K_i, we provide it to BERT as:

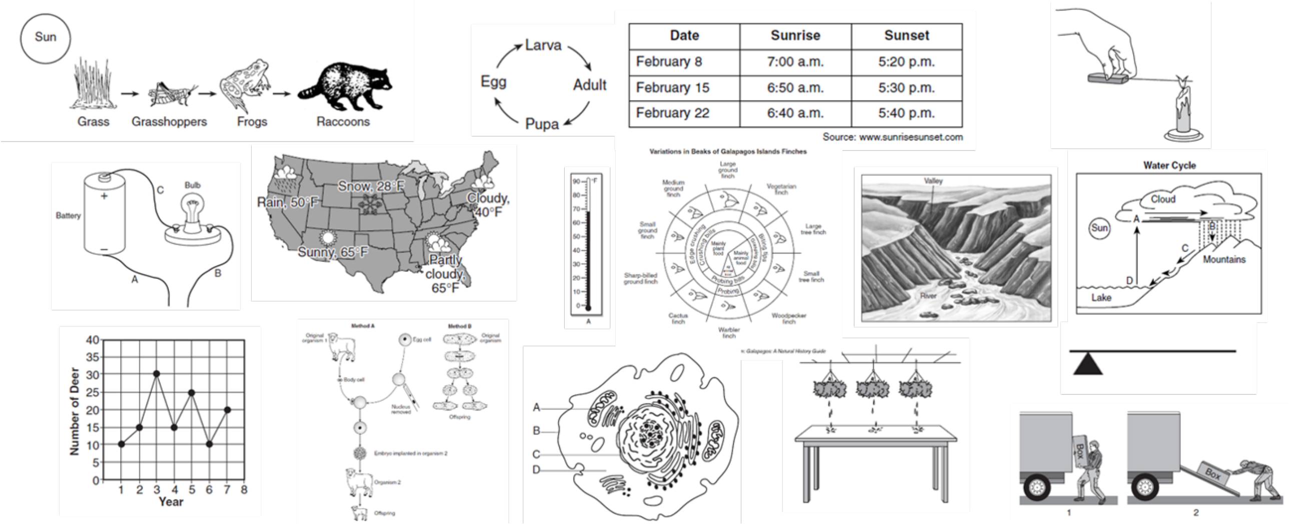
[CLS] K_i *[SEP]* q *[SEP]* a_i *[SEP]*

for each option (only the answer option is assigned as the second BERT "segment"). The [CLS] output token for each answer option is projected to a single logit and fed through a softmax layer, trained using cross-entropy loss against the correct answer.

The **AristoBERT solver** uses three methods to apply BERT more effectively. First, we retrieve and supply background knowledge along with the question when using BERT. This provides the potential for BERT to "read" that background knowledge and apply it to the question, although the exact nature of how it uses background knowledge is more complex and less interpretable. Second, we fine-tune BERT using a curriculum of several datasets, including some that are not science related. Finally, we ensemble different variants of BERT together.

1. **Background Knowledge** For background knowledge K_i we use up to 10 of the top sentences found by the IR solver, truncated to fit into the BERT *max tokens* setting (we use 256).

2. **Curriculum Fine-Tuning** Following earlier work on multi-step fine-tuning (Sun et al., 2019), we first fine-tune on the large (87866 qs) RACE training set (Lai et al.,



Date	Sunrise	Sunset
February 8	7:00 a.m.	5:20 p.m.
February 15	6:50 a.m.	5:30 p.m.
February 22	6:40 a.m.	5:40 p.m.

Figure 6: A sample of the wide variety of diagrams used in the Regents exams, including food chains, pictures, tables, graphs, circuits, maps, temporal processes, cross-sections, pie charts, and flow diagrams.

2017), a challenging set of English comprehension multiple choice exams given in Chinese middle and high schools.

We then further fine-tune on a collection of science multiple choice questions sets:

- OpenBookQA train (4957 qs) (Mihaylov et al., 2018)
- ARC-Easy train (2251 qs) (Clark et al., 2018)
- ARC-Challenge train (1119 qs) (Clark et al., 2018)
- 22 Regents Living Environment exams (665 qs).[4]

We optimize the final fine-tuning using scores on the development set, performing a small hyperparameter search as suggested in the original BERT paper (Devlin et al., 2018).

3. **Ensembling** We repeat the above using three variants of BERT, the original BERT-large-cased and BERT-large-uncased, as well as the later released BERT-large-cased-whole-word-masking.[5] We also add a model trained without background knowledge and ensemble them using the combination solver described below.

The **AristoRoBERTa solver** takes advantage of the recent release of Roberta (Liu et al., 2019b), a high-performing and optimized derivative of BERT trained on significantly more text. In AristoRoBERTa, we simply replace the BERT model in AristoBERT with RoBERTa, repeating similar fine-tuning steps. We ensemble two versions together, namely with and without the first fine-tuning step using RACE.

3.5 Ensembling

Each solver outputs a non-negative confidence score for each of the answer options along with other optional features. The Combiner then produces a combined confidence score (between 0 and 1) using the following two-step approach.

In the first step, each solver is "calibrated" on the training set by learning a logistic regression classifier from each answer option to a correct/incorrect label. The features for an answer option i include the raw confidence score s_i as well as the score normalized across the answer options for a given question. We include two types of normalizations:

$$normal_i = \frac{s_i}{\sum_j s_j} \qquad softmax_i = \frac{\exp(s_i)}{\sum_j \exp(s_j)}$$

Each solver can also provide other features capturing aspects of the question or the reasoning path. The output of this first step classifier is then a calibrated confidence for each solver s and answer option i: $calib_i^s = 1/(1+\exp(-\beta^s \cdot f^s))$ where f^s is the solver specific feature vector and β^s the associated feature weights.

The second step uses these calibrated confidences as (the only) features to a second logistic regression classifier from answer option to correct/incorrect, resulting in a final confidence in $[0, 1]$, which is used to rank the answers:

$$\mathit{confidence}_i = 1/\left(1 + \exp\left(-\beta^0 - \sum_{s \in \mathrm{Solvers}} \beta^s calib_i^s\right)\right)$$

Here, feature weights β^s indicate the contribution of each solver to the final confidence. Empirically, this two-step approach yields more robust predictions given limited training data compared to a one-step approach where all solver features are fed directly into a single classification step.

[4] https://www.nysedregents.org/livingenvironment, months 99/06, 01/06, 02/01, 02/08, 03/08, 04/01, 05/01, 05/08, 07/01, 08/06, 09/01, 09/08, 10/01, 11/01, 11/08, 12/06, 13/08, 15/01, 16/01, 17/06, 17/08, 18/06

[5] https://github.com/google-research/bert (5/31/2019 notes)

Test Set	Num Q	IR	PMI	ACME	TupInf	Multee	AristoBERT	AristoRoBERTa	ARISTO
Regents 4th	109	64.45	66.28	67.89	63.53	69.72	86.24	88.07	**89.91**
Regents 8th	119	66.60	69.12	67.65	61.41	68.91	86.55	88.24	**91.60**
Regents 12th	632	41.22	46.95	41.57	35.35	56.01	75.47	82.28	**83.54**
ARC-Easy	2376	74.48	77.76	66.60	57.73	64.69	81.78	82.88	**86.99**
ARC-Challenge	1172	n/a†	n/a†	20.44	23.73	37.36	57.59	**64.59**	64.33

†ARC-Challenge is defined using IR and PMI results, i.e., are questions that by definition both IR and PMI get wrong (Clark et al., 2018).

Table 2: This table shows the results of each of the Aristo solvers, as well as the overall Aristo system, on each of the test sets. Most notably, Aristo achieves 91.6% accuracy in 8th Grade, and exceeds 83% in 12th Grade. ("Num Q" refers to the number of questions in each test set.). Note that Aristo is a single system, run unchanged on each dataset (not retuned for each dataset).

	Partition			
Dataset	Train	Dev	Test	Total
Regents 4th	127	20	109	256
Regents 8th	125	25	119	269
Regents 12th	665	282	632	1579
ARC-Easy	2251	570	2376	5197
ARC-Challenge	1119	299	1172	2590
Totals†	4035	1151	4180	9366

†ARC (Easy + Challenge) includes Regents 4th and 8th as a subset.

Table 3: Dataset partition sizes (number of questions).

4 Experiments and Results

This section describes our precise experimental methodology followed by our results.

4.1 Experimental Methodology

Omitted Question Classes In the experimental results reported below, we omitted questions that utilized diagrams. While these questions are frequent in the test, they are outside of our focus on language and reasoning. Moreover, the diagrams are highly varied (see Figure 6) and despite work that tackled narrow diagram types, e.g, food chains (Krishnamurthy et al., 2016), overall progress has been quite limited (Choi et al., 2017).

We also omitted questions that require a direct answer (rather than selecting from multiple choices), for two reasons. First, after removing questions with diagrams, they are rare in the remainder. Of the 482 direct answer questions over 13 years of Regents 8th Grade Science exams, only 38 (<8%) do *not* involve a diagram. Second, they are complex, often requiring explanation and synthesis. Both diagram and direct-answer questions are natural topics for future work.

Dataset Formulation We evaluate Aristo using several datasets of independently-authored science questions taken from standardized tests. Each dataset is divided into train, development, and test partitions, the test partitions being "blind", i.e., hidden to both the researchers and the Aristo system during training. All questions are taken verbatim from the original sources, with no rewording or modification. As mentioned earlier, we use only the non-diagram, multiple choice (NDMC) questions. We exclude questions with an associated diagram that is required to interpret the question. In the occasional case where two questions share the same preamble, the preamble is repeated for each question so they are independent. The Aristo solvers are trained using questions in the training partition (each solver is trained independently, as described earlier), and then the combination is fine-tuned using the development set.

The Regents exam questions are taken verbatim from the New York Regents Examination board, using the 4th Grade Science, 8th Grade Science, and 12th Grade Living Environment examinations.[6] The questions are partitioned into train/dev/test by exam, i.e., each exam is either in train, dev, or test but not split up between them. The ARC dataset is a larger corpus of science questions drawn from public resources across the country, spanning grades 3 to 9, and also includes the Regents 4th and 8th questions (using the same train/dev/test split). Further details of the datasets are described in (Clark et al., 2018). The datasets are publicly available[7]. Dataset sizes are shown in Table 3. All but 39 of the 9366 questions are 4-way multiple choice, the remaining 39 (<0.5%) being 3- or 5-way. A random score over the entire dataset is 25.02%.

For each question, the answer option with the highest overall confidence from Aristo's combination module is selected, scoring 1 point if the answer is correct, 0 otherwise. In the (very rare) case of N options having the same confidence (an N-way tie) that includes the correct option, the system receives 1/N points (equivalent to the asymptote of random guessing between the N).

4.2 Main Results

The results are summarized in Table 2, showing the performance of the solvers individually, and their combination in the full Aristo system. Note that Aristo is a single system run on the five datasets (not retuned for each dataset in turn).

Most notably, Aristo's scores on the Regents Exams far exceed earlier performances (e.g., Schoenick et al., 2016; Clark et al., 2016), and represents a new high-point on science questions.

In addition, the results show the dramatic impact of new language modeling technology, embodied in AristoBERT and AristoRoBERTa, the scores for these two solvers dominating the performance of the overall system. Even on the ARC-Challenge questions, containing a wide variety of dif-

[6] See https://www.nysedregents.org/ for the original exams.

[7] http://data.allenai.org/arc/, and the 12th Grade Regents data is available on request

Test dataset	ARISTO
Regents 4th (benchmark, Table 2)	89.91
Regents 4th (years 2017-2019)	92.86
Regents 8th (benchmark, Table 2)	91.60
Regents 8th (years 2017-2019)	93.33

Table 4: Aristo's score on the three most recent years of Regents Science (2017-19), not part of the hidden benchmark.

ficult questions, the language modeling based solvers dominate. The general increasing trend of solver scores from left to right in the table loosely reflects the progression of the NLP field over the six years of the project.

To check that we have not overfit to our data, we also ran Aristo on the most recent years of the Regents Grade Exams (4th and 8th Grade), years 2017-19, that were unavailable at the start of the project and were not part of our datasets. The results are shown in Table 4, a showing score similar to those on our larger datasets, suggesting the system is not overfit.

On the entire exam, the NY State Education Department considers a score of 65% as "Meeting the Standards", and over 85% as "Meeting the Standards with Distinction"[8]. If this rubric applies equally to the NDMC subset we have studied, this would mean Aristo has met the standard with distinction in 8th Grade Science.

4.3 Answer Only Performance

Several authors have observed that for some multiple choice datasets, systems can still perform well even when ignoring the question body and looking only at the answer options (Gururangan et al., 2018; Poliak et al., 2018). This surprising result is particularly true for crowdsourced datasets, where workers may use stock words or phrases (e.g., "not") in incorrect answer options that gives them away. A dataset with this characteristic is clearly problematic, as systems can spot such cues and do well without even reading the question.

To measure this phenomenon on our datasets, we trained and tested a new AristoRoBERTa model giving it only the answer options (no question body nor retrieved knowledge). The results on the test partition are shown in Table 5. We find scores significantly above random (25%), in particular for the 12th Grade set which has longer answers. But the scores are sufficiently low to indicate the datasets are relatively free of annotation artifacts that would allow the system to often guess the answer independent of the question. This desirable feature is likely due to the fact these are natural science questions, carefully crafted by experts for inclusion in exams, rather than mass-produced through crowdsourcing.

4.4 Adversarial Answer Options

One way of testing robustness in multiple choice is to change or add incorrect answer options, and see if the system's performance degrades (Khashabi et al., 2016). If a system has mastery of the material, we would expect its score to be relatively unaffected by such modifications. To explore this, we investigated *adversarially* adding extra incorrect options, i.e., searching for answer options that might confuse the system, using AristoRoBERTa[9], and adding them as extra choices to the existing questions.

Test dataset	"Answer only" score	% Drop (relative)
Regents 4th	38.53	56.7
Regents 8th	37.82	56.3
Regents 12th	47.94	41.2
ARC-Easy	36.17	55.9
ARC-Challenge	35.92	44.7
All	37.11	48.5

Table 5: Scores when looking at the answer options only for (retrained) AristoRoBERTa (no ensembling), compared with using the full questions. The (desirably) low scores/large drops indicate it is hard to guess the answer without reading the question.

Test dataset	4-way MC	Adversarial 8-way MC	% drop (relative)
Regents 4th	87.1	76.1	12.6
Regents 8th	78.9	76.4	3.1
Regents 12th	75.3	58.0	22.9
ARC-Easy	74.1	65.7	11.3
ARC-Challenge	55.5	47.7	14.0
ALL	69.1	59.5	13.8

Table 6: Scores on the original 4-way multiple choice questions, and (after retraining) on adversarially generated 8-way multiple choice versions, for AristoRoBERTa (no ensembling).

To do this, for each question, we collect a large (≈ 100) number of candidate additional answer choices using the correct answers to *other* questions in the same dataset (and train/test split), where the top 100 are chosen by a superficial alignment score (features such as answer length and punctuation usage). We then re-rank these additional choices using AristoRoBERTa, take the top N, and add them to the original K (typically 4) choices for the question.

If we add N=4 extra choices to the normal 4-way questions, they become 8-way multiple choice, and performance drops dramatically (over 40 percentage points), albeit unfairly as we have by definition added choices that confuse the system. We then train the model further on this 8-way adversarial dataset, a process known as inoculation (Liu et al., 2019a). After further training, we still find a drop, but significantly less (around 10 percentage points absolute, 13.8% relative, Table 6), even though many of the new distractor choices would be easy for a human to rule out.

For example, while the solver gets the right answer to the

[8] https://www.nysedregents.org/grade8/science/618/home.html

[9] For computational tractability, we slightly modify the way background knowledge is retrieved for this experiment (only), namely using a search query of just the question body q (rather than question + answer option $q + a_i$).

following question:

> *The condition of the air outdoors at a certain time of day is known as (A) friction (B) light (C) force (D) weather* **[selected, correct]**

it fails for the 8-way variant:

> *The condition of the air outdoors at a certain time of day is known as (A) friction (B) light (C) force (D) weather* **[correct]** *(Q) joule (R) gradient* **[selected]** *(S) trench (T) add heat*

These results show that while Aristo performs well, it still has some blind spots that can be artificially uncovered through adversarial methods such as this.

5 Related Work

This section describes related work on answering standardized-test questions, and on math word problems in particular. It provides an overview rather than exhaustive citations.

5.1 Standardized Tests

Standardized tests have long been proposed as challenge problems for AI (e.g., Bringsjord and Schimanski, 2003; Brachman et al., 2005; Clark and Etzioni, 2016; Piatetsky-Shapiro et al., 2006), as they appear to require significant advances in AI technology while also being accessible, measurable, understandable, and motivating.

Earlier work on standardized tests focused on specialized tasks, for example, SAT word analogies (Turney, 2006), GRE word antonyms (Mohammad et al., 2013), and TOEFL synonyms (Landauer and Dumais, 1997). More recently, there have been attempts at building systems to pass university entrance exams. Under NII's Todai project, several systems were developed for parts of the University of Tokyo Entrance Exam, including maths, physics, English, and history (Strickland, 2013; NII, 2013; Fujita et al., 2014), although in some cases questions were modified or annotated before being given to the systems (e.g., Matsuzaki et al., 2014). Similarly, a smaller project worked on passing the Gaokao (China's college entrance exam) (e.g., Cheng et al., 2016; Guo et al., 2017). The Todai project was reported as ended in 2016, in part because of the challenges of building a machine that could "grasp meaning in a broad spectrum" (Mott, 2016).

5.2 Math Word Problems

Substantial progress has been achieved on math word problems. On plane geometry questions, (Seo et al., 2015) demonstrated an approach that achieve a 61% accuracy on SAT practice questions. The Euclid system (Hopkins et al., 2017) achieved a 43% recall and 91% precision on SAT "closed-vocabulary" algebra questions, a limited subset of questions that nonetheless constitutes approximately 45% of a typical math SAT exam. Closed-vocabulary questions are those that do not reference real-world situations (e.g., "what is the largest prime smaller than 100?" or "Twice the product of x and y is 8. What is the square of x times y?")

Work on open-world math questions has continued, but results on standardized tests have not been reported and thus it is difficult to benchmark the progress relative to human performance. See Amini et al. (2019) for a recent snapshot of the state of the art, and references to the literature on this problem.

6 Summary and Conclusion

Answering science questions is a long-standing AI grand challenge (Reddy, 1988; Friedland et al., 2004). This paper reports on Aristo—the first system to achieve a score of over 90% on the non-diagram, multiple choice part of the New York Regents 8th Grade Science Exam, demonstrating that modern NLP methods can result in mastery of this task. Although Aristo only answers multiple choice questions without diagrams, and operates only in the domain of science, it nevertheless represents an important milestone towards systems that can read and understand. The momentum on this task has been remarkable, with accuracy moving from roughly 60% to over 90% in just three years. Finally, the use of independently authored questions from a standardized test allows us to benchmark AI performance relative to human students.

Beyond the use of a broad vocabulary and scientific concepts, many of the benchmark questions intuitively appear to require reasoning to answer (e.g., Figure 5). To what extent is Aristo reasoning to answer questions? For many years in AI, reasoning was thought of as the discrete, symbolic manipulation of sentences expressed in a formally designed language (Brachman and Levesque, 1985; Genesereth and Nilsson, 2012). With the advent of deep learning, this notion of reasoning has shifted, with machines performing challenging tasks using neural architectures rather than explicit representation languages. Today, we do not have a sufficiently fine-grained notion of reasoning to answer this question precisely, but we can observe surprising performance on answering science questions. This suggests that the machine has indeed learned something about language and the world, and how to manipulate that knowledge, albeit neither symbolically nor discretely.

Although an important milestone, this work is only a step on the long road toward a machine that has a deep understanding of science and achieves Paul Allen's original dream of a Digital Aristotle. A machine that has fully understood a textbook should not only be able to answer the multiple choice questions at the end of the chapter—it should also be able to generate both short and long answers to direct questions; it should be able to perform constructive tasks, e.g., designing an experiment for a particular hypothesis; it should be able to explain its answers in natural language and discuss them with a user; and it should be able to learn directly from an expert who can identify and correct the machine's misunderstandings. These are all ambitious tasks still largely beyond the current technology, but with the rapid progress happening in NLP and AI, solutions may arrive sooner than we expect.

Acknowledgements

We gratefully acknowledge the many other contributors to this work, including Niranjan Balasubramanian, Matt Gardner, Peter Jansen, Daniel Khashabi, Jayant Krishnamurthy, Souvik Kundu, Todor Mihaylov, Michael Schmitz, Harsh Trivedi, Peter Turney, and the Beaker team at AI2.

References

T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1): 1–41, 2009.

P. Allen. *Idea Man: A memoir by the cofounder of Microsoft*. Penguin, 2012.

A. Amini, S. Gabriel, P. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *NAACL-HLT*, 2019.

J. Aron. Software tricks people into thinking it is human. *New Scientist*, (Issue 2829), Sept 2011.

M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, 2007.

BBC. Computer AI passes Turing Test in 'world first'. *BBC News*, 2014. http://www.bbc.com/news/technology-27762088.

R. Brachman, D. Gunning, S. Bringsjord, M. Genesereth, L. Hirschman, and L. Ferro. Selected grand challenges in cognitive science. Technical report, MITRE Technical Report 05-1218. Bedford MA: The MITRE Corporation, 2005.

R. J. Brachman and H. J. Levesque. *Readings in knowledge representation*. Morgan Kaufmann Publishers Inc., 1985.

S. Bringsjord and B. Schimanski. What is artificial intelligence? Psychometric AI as an answer. In *IJCAI*, pp. 887–893. Citeseer, 2003.

V. K. Chaudhri, B. Cheng, A. Overtholtzer, J. Roschelle, A. Spaulding, P. Clark, M. Greaves, and D. Gunning. Inquire Biology: A textbook that answers questions. *AI Magazine*, 34(3):55–72, 2013.

G. Cheng, W. Zhu, Z. Wang, J. Chen, and Y. Qu. Taking up the gaokao challenge: An information retrieval approach. In *IJCAI*, pp. 2479–2485, 2016.

J. Choi, J. Krishnamurthy, A. Kembhavi, and A. Farhadi. Structured set matching networks for one-shot part labeling. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3627–3636, 2017.

K. W. Church and P. Hanks. Word association norms, mutual information and lexicography. In *27th ACL*, pp. 76–83, 1989.

P. Clark, N. Balasubramanian, S. Bhakthavatsalam, K. Humphreys, J. Kinkead, A. Sabharwal, and O. Tafjord. Automatic construction of inference-supporting knowledge bases. In *4th Workshop on Automated Knowledge Base Construction (AKBC)*, Montreal, Canada, Dec. 2014.

P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? Try ARC, the AI2 Reasoning Challenge. *ArXiv*, abs/1803.05457, 2018.

P. Clark and O. Etzioni. My computer is an honor student - But how intelligent is it? Standardized tests as a measure of AI. *AI Magazine*, 37(1):5–12, 2016.

P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. D. Turney, and D. Khashabi. Combining retrieval, statistics, and inference to answer elementary science questions. In *AAAI*, pp. 2580–2586, 2016.

P. Clark, P. Harrison, and N. Balasubramanian. A study of the knowledge base requirements for passing an elementary science test. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pp. 37–42. ACM, 2013.

I. Dagan, B. Dolan, B. Magnini, and D. Roth. Recognizing textual entailment: Rational, evaluation and approaches–erratum. *Natural Language Engineering*, 16(01):105–105, 2010.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018.

D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3):59–79, 2010.

N. S. Friedland, P. G. Allen, G. Matthews, M. Witbrock, D. Baxter, J. Curtis, B. Shepard, P. Miraglia, J. Angele, S. Staab, et al. Project Halo: Towards a digital Aristotle. *AI magazine*, 25(4):29–29, 2004.

A. Fujita, A. Kameda, A. Kawazoe, and Y. Miyao. Overview of Todai robot project and evaluation framework of its NLP-based problem solving. In *LREC*, 2014.

M. R. Genesereth and N. J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann, 2012.

D. Gunning, V. K. Chaudhri, P. E. Clark, K. Barker, S.-Y. Chaw, M. Greaves, B. Grosof, A. Leung, D. D. McDonald, S. Mishra, et al. Project Halo update – Progress toward digital Aristotle. *AI Magazine*, 31(3):33–58, 2010.

S. Guo, X. Zeng, S. He, K. Liu, and J. Zhao. Which is the effective way for gaokao: Information retrieval or neural networks? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 111–120, 2017.

S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith. Annotation artifacts in natural language inference data. In *NAACL*, 2018.

M. Hopkins, C. Petrescu-Prahova, R. Levin, R. L. Bras, A. Herrasti, and V. Joshi. Beyond sentential semantic parsing: Tackling the math SAT with a cascade of tree transducers. In *EMNLP*, 2017.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *ACL*, 2018.

M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

D. Khashabi, T. Khot, A. Sabharwal, P. Clark, O. Etzioni, and D. Roth. Question answering via integer programming over semi-structured knowledge. In *IJCAI*, 2016.

D. Khashabi, T. Khot, A. Sabharwal, and D. Roth. Question answering as global reasoning over semantic abstractions. In *AAAI*, 2018.

T. Khot, A. Sabharwal, and P. F. Clark. Answering complex questions using open information extraction. In *ACL*, 2017.

J. Krishnamurthy, O. Tafjord, and A. Kembhavi. Semantic parsing to probabilistic programs for situated question answering. In *EMNLP*, 2016.

G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*, 2017.

T. K. Landauer and S. T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

J. H. Larkin, J. McDermott, D. P. Simon, and H. A. Simon. Models of competence in solving physics problems. *Cognitive Science*, 4:317–345, 1980.

N. F. Liu, R. Schwartz, and N. A. Smith. Inoculation by fine-tuning: A method for analyzing challenge datasets. *ArXiv*, abs/1904.02668, 2019a.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

T. Matsuzaki, H. Iwane, H. Anai, and N. H. Arai. The most uncreative examinee: a first step toward wide coverage natural language math problem solving. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

T. Mihaylov, P. F. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

S. M. Mohammad, B. J. Dorr, G. Hirst, and P. D. Turney. Computing lexical contrast. *Computational Linguistics*, 39(3):555–590, 2013.

N. Mott. Todai robot gives up on getting into the university of tokyo. *Inverse*, 2016. (https://www.inverse.com/article/23761-todai-robot-gives-up-university-tokyo).

NII. The Todai robot project. *NII Today*, 46, July 2013. (http://www.nii.ac.jp/userdata/results/pr_data/NII_Today/60_en/all.pdf).

M. E. Peters, M. Neumann, M. Iyyer, M. P. Gardner, C. Clark, K. Lee, and L. S. Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

G. Piatetsky-Shapiro, C. Djeraba, L. Getoor, R. Grossman, R. Feldman, and M. Zaki. What are the grand challenges for data mining?: Kdd-2006 panel report. *ACM SIGKDD Explorations Newsletter*, 8(2):70–77, 2006.

A. Poliak, J. Naradowsky, A. Haldar, R. Rudinger, and B. Van Durme. Hypothesis only baselines in natural language inference. In *StarSem*, 2018.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pretraining. *Technical report, OpenAI*, 2018.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

R. Reddy. Foundations and grand challenges of artificial intelligence: AAAI presidential address. *AI Magazine*, 9 (4), 1988.

R. Reddy. Three open problems in AI. *J. ACM*, 50:83–86, 2003.

C. Schoenick, P. F. Clark, O. Tafjord, P. D. Turney, and O. Etzioni. Moving beyond the Turing Test with the Allen AI Science Challenge. *CACM*, 2016.

M. J. Seo, H. Hajishirzi, A. Farhadi, O. Etzioni, and C. Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*, 2015.

M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603, 2016.

E. Strickland. Can an ai get into the university of tokyo? *IEEE Spectrum*, 50(9):13–14, 2013.

K. Sun, D. Yu, D. Yu, and C. Cardie. Improving machine reading comprehension with general reading strategies. In *NAACL-HLT*, 2019.

O. Tafjord, M. Gardner, K. Lin, and P. Clark. QuaRTz: An open-domain dataset of qualitative relationship questions. In *EMNLP*, 2019. (to appear).

N. Tandon, B. D. Mishra, J. Grus, W.-t. Yih, A. Bosselut, and P. Clark. Reasoning about actions and state changes by injecting commonsense knowledge. *arXiv preprint arXiv:1808.10012*, 2018.

H. Trivedi, H. Kwon, T. Khot, A. Sabharwal, and N. Balasubramanian. Repurposing entailment for multi-hop question answering tasks. In *NAACL*, 2019.

A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236), 1950.

P. D. Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, 2006.

P. D. Turney. Leveraging term banks for answering complex questions: A case for sparse vectors. *arXiv preprint arXiv:1704.03543*, 2017.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

W. Wang, M. Yan, and C. Wu. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In *ACL*, 2018.

R. Zellers, Y. Bisk, R. Schwartz, and Y. Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *ArXiv*, abs/1808.05326, 2018.