MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text

Matthew Richardson
Microsoft Research
One Microsoft Way
Redmond, WA 98052
mattri@microsoft.com

Christopher J.C. Burges
Microsoft Research
One Microsoft Way
Redmond, WA 98052
cburges@microsoft.com

Erin Renshaw
Microsoft Research
One Microsoft Way
Redmond, WA 98052
erinren@microsoft.com

Abstract

We present MCTest, a freely available set of stories and associated questions intended for research on the machine comprehension of text. Previous work on machine comprehension (e.g., semantic modeling) has made great strides, but primarily focuses either on limited-domain datasets, or on solving a more restricted goal (e.g., open-domain relation extraction). In contrast, MCTest requires machines to answer multiple-choice reading comprehension questions about fictional stories, directly tackling the high-level goal of open-domain machine comprehension. Reading comprehension can test advanced abilities such as causal reasoning and understanding the world, yet, by being multiple-choice, still provide a clear metric. By being fictional, the answer typically can be found only in the story itself. The stories and questions are also carefully limited to those a young child would understand, reducing the world knowledge that is required for the task. We present the scalable crowd-sourcing methods that allow us to cheaply construct a dataset of 500 stories and 2000 questions. By screening workers (with grammar tests) and stories (with grading), we have ensured that the data is the same quality as another set that we manually edited, but at one tenth the editing cost. By being open-domain, yet carefully restricted, we hope MCTest will serve to encourage research and provide a clear metric for advancement on the machine comprehension of text.

1 Reading Comprehension

A major goal for NLP is for machines to be able to understand text as well as people. Several research disciplines are focused on this problem: for example, information extraction, relation extraction, semantic role labeling, and recognizing textual entailment. Yet these techniques are necessarily evaluated individually, rather than by how much they advance us towards the end goal. On the other hand, the goal of semantic parsing is the machine comprehension of text (MCT), yet its evaluation requires adherence to a specific knowledge representation, and it is currently unclear what the best representation is, for open-domain text.

We believe that it is useful to directly tackle the top-level task of MCT. For this, we need a way to measure progress. One common method for evaluating someone's understanding of text is by giving them a multiple-choice reading comprehension test. This has the advantage that it is objectively gradable (vs. essays) yet may test a range of abilities such as causal or counterfactual reasoning, inference among relations, or just basic understanding of the world in which the passage is set.

Therefore, we propose a multiple-choice reading comprehension task as a way to evaluate progress on MCT. We have built a reading comprehension dataset containing 500 fictional stories, with 4 multiple choice questions per story. It was built using methods which can easily scale to at least 5000 stories, since the stories were created, and the curation was done, using crowd sourcing almost entirely, at a total of $4.00 per story. We plan to periodically update the dataset to ensure that methods are not overfitting to the existing data. The dataset is open-domain, yet restricted to concepts and words that a 7 year old is expected to understand. This task is still beyond the capability of today's computers and algorithms.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 193–203,
Seattle, Washington, USA, 18-21 October 2013.

By restricting the concept space, we gain the difficulty of being an open-domain problem, without the full complexity of the real world (for example, there will be no need for the machine to understand politics, technology, or to have any domain specific expertise). The multiple choice task avoids ambiguities (such as when the task is to find a sentence that best matches a question, as in some early reading comprehension tasks: see Section 2), and also avoids the need for additional grading, such as is needed in some TREC tasks. The stories were chosen to be fictional to focus work on finding the answer in the story itself, rather than in knowledge repositories such as Wikipedia; the goal is to build technology that actually understands stories and paragraphs on a deep level (as opposed to using information retrieval methods and the redundancy of the web to find the answers).

We chose to use crowd sourcing, as opposed to, for example, contracting teachers or paying for existing standardized tests, for three reasons, namely: (1) scalability, both for the sizes of datasets we can provide, and also for the ease of regularly refreshing the data; (2) for the variety in story-telling that having many different authors brings; and (3) for the free availability that can only result from providing non-copyrighted data. The content is freely available at http://research.microsoft.com/mct, and we plan to use that site to track published results and provide other resources, such as labels of various kinds.

2 Previous Work

The research goal of mapping text to meaning representations in order to solve particular tasks has a long history. DARPA introduced the Airline Travel Information System (ATIS) in the early 90's: there the task was to slot-fill flight-related information by modeling the intent of spoken language (see Tur et al., 2010, for a review). This data continues to be a used in the semantic modeling community (see, for example, Zettlemoyer and Collins, 2009). The Geoquery database contains 880 geographical facts about the US and has played a similar role for written (as opposed to spoken) natural language queries against a database (Zelle and Mooney, 1996) and it also continues to spur research (see for example Goldwasser et al., 2011), as does the similar Jobs database, which provides mappings of 640 sentences to a listing of jobs (Tang and Mooney, 2001). More recently, Zweig and Burges (2012) provided a set of 1040 sentences that comprise an SAT-style multiple choice sentence completion task.

The idea of using story-based reading comprehension questions to evaluate methods for machine reading itself goes back over a decade, when Hirschmann et al. (1999) showed that a bag of words approach, together with some heuristic linguistic modeling, could achieve 40% accuracy for the task of picking the sentence that best matches the query for "who / what / when / where / why" questions, on a small reading comprehension dataset from Remedia. This dataset spurred several research efforts, for example using reinforcement learning (Grois and Wilkins, 2005), named entity resolution (Harabagiu et al., 2003) and mapping questions and answers to logical form (Wellner et al., 2006). Work on story understanding itself goes back much further, to 1972, when Charniak proposed using a background model to answer questions about children's stories. Similarly, the TREC (and TAC) Question Answering tracks (e.g., Voorhees and Tice, 1999) aim to evaluate systems on their ability to answer factual questions such as "Where is the Taj Mahal". The QA4MRE task also aims to evaluate machine reading systems through question answering (e.g., Clark et al., 2012). Earlier work has also aimed at controlling the scope by limiting the text to children's stories: Breck et al. (2001) collected 75 stories from the Canadian Broadcasting Corporation's web site for children, and generated 650 questions for them manually, where each question was answered by a sentence in the text. Leidner et al. (2003) both enriched the CBC4kids data by adding several layers of annotation (such as semantic and POS tags), and measured QA performance as a function of question difficulty. For a further compendium of resources related to the story comprehension task, see Mueller (2010).

The task proposed here differs from the above work in several ways. Most importantly, the data collection is scalable: if the dataset proves sufficiently useful to others, it would be straightforward to gather an order of magnitude more. Even the dataset size presented here is an order of magnitude larger than the Remedia or the CBC4kids data and many times larger than QA4MRE. Second, the multiple choice task presents less ambiguity (and is consequently easier to collect data for) than the

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.
One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.
His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.
After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
A) Fries
B) Pudding
C) James
D) Jane

2) What did James pull off of the shelves in the grocery store?
A) pudding
B) fries
C) food
D) splinters

3) Where did James go after he went to the grocery store?
A) his deck
B) his freezer
C) a fast food restaurant
D) his room

4) What did James do after he ordered the fries?
A) went to the grocery store
B) went home without paying
C) ate them
D) made up his mind to be a better turtle

Figure 1. Sample Story and Questions (chosen randomly from MC500 train set).

task of finding the most appropriate sentence, and may be automatically evaluated. Further, our stories are fictional, which means that the information to answer the question is contained only in the story itself (as opposed to being able to directly leverage knowledge repositories such as Wikipedia). This design was chosen to focus the task on the machine understanding of short passages, rather than the ability to match against an existing knowledge base. In addition, while in the CBC4kids data each answer was a sentence from the story, here we required that approximately half of the questions require at least two sentences from the text to answer; being able to control complexity in this way is a further benefit of using multiple choice answers. Finally, as explained in Section 1, the use of free-form input makes the problem open domain (as opposed to the ATIS, Geoquery and Jobs data), leading to the hope that solutions to the task presented here will be easier to apply to novel, unrelated tasks.

3 Generating the Stories and Questions

Our aim was to generate a corpus of fictional story sets[1] that could be scaled with as little expert input as possible. Thus, we designed the process to be gated by cost, and keeping the costs low was a high priority. Crowd-sourcing seemed particularly appropriate, given the nature of the task, so we opted to use Amazon Mechanical Turk[2] (AMT). With over 500,000 workers[3], it provides the work force required to both achieve scalability and, equally importantly, to provide diversity in the stories and types of questions. We restricted our task to AMT workers (*workers*) residing in the United States. The average worker is 36 years old, more educated than the United States population in general (Paolacci et al., 2010), and the majority of workers are female.

3.1 The Story and Questions

Workers were instructed to write a short (150-300 words) fictional story, and to write as if for a child in grade school. The choice of 150-300 was made to keep the task an appropriate size for workers while still allowing for complex stories and questions. The workers were free to write about any topic they desired (as long as it was appropriate for a young child), and so there is a wide range, including vacations, animals, school, cars, eating, gardening, fairy tales, spaceships, and cowboys.

[1] We use the term "story set" to denote the fictional story together with its multiple choice questions, hypothetical answers, and correct answer labels.

[2] http://www.mturk.com

[3] https://requester.mturk.com/tour

Workers were also asked to provide four reading comprehension questions pertaining to their story and, for each, four multiple-choice answers. Coming up with incorrect alternatives (*distractors*) is a difficult task (see, e.g., Agarwal, 2011) but workers were requested to provide "reasonable" incorrect answers that at least include words from the story so that their solution is not trivial. For example, for the question "*What is the name of the dog?*", if only one of the four answers occurs in the story, then that answer must be the correct one.

Finally, workers were asked to design their questions and answers such that at least two of the four questions required multiple sentences from the story to answer them. That is, for those questions it should not be possible to find the answer in any individual sentence. The motivation for this was to ensure that the task could not be fully solved using lexical techniques, such as word matching, alone. Whilst it is still possible that a sophisticated lexical analysis could completely solve the task, requiring that answers be constructed from at least two different sentences in the story makes this much less likely; our hope is that the solution will instead require some inference and some form of limited reasoning. This hope rests in part upon the observation that standardized reading comprehension tests, whose goal after all is to test comprehension, generally avoid questions that can be answered by reading a single sentence.

3.2 Automatic Validation

Besides verifying that the story and all of the questions and answers were provided, we performed the following automatic validation before allowing the worker to complete the task:

Limited vocabulary: The lowercase words in the story, questions, and answers were stemmed and checked against a vocabulary list of approximately 8000 words that a 7-year old is likely to know (Kuperman et al., 2012). Any words not on the list were highlighted in red as the worker typed, and the task could not be submitted unless all of the words satisfied this vocabulary criterion. To allow the use of arbitrary proper nouns, capitalized words were not checked against the vocabulary list.

Multiple-sentence questions: As described earlier, we required that at least two of the questions need multiple sentences to answer. Workers were simply asked to mark whether a question needs one or multiple sentences and we required that at least two are marked as *multiple*.

3.3 The Workers

Workers were required to reside in the United States and to have completed 100 HITs with an over 95% approval rate[4]. The median worker took 22 minutes to complete the task. We paid workers $2.50 per story set and allowed each to do a maximum of 8 tasks (5 in MC500). We did not experiment with paying less, but this rate amounts to $6.82/hour, which is approximately the rate paid by other writing tasks on AMT at the time, though is also significantly higher than the median wage of $1.38 found in 2010 (Horton and Chilton, 2010). Workers could optionally leave feedback on the task, which was overwhelmingly positive – the most frequent non-stopword in the comments was "*fun*" and the most frequent phrase was "*thank you*". The only negative comments (in <1% of submissions) were when the worker felt that a particular word should have been on the allowed vocabulary list. Given the positive feedback, it may be possible to pay less if we collect more data in the future. We did not enforce story length constraints, but some workers interpreted our suggestion that the story be 150-300 words as a hard constraint, and some asked to be able to write a longer story.

The MCTest corpus contains two sets of stories, named MC160 and MC500, and containing 160 and 500 stories respectively. MC160 was gathered first, then some improvements were made before gathering MC500. We give details on the differences between these two sets below.

3.4 MC160: Manually Curated for Quality

In addition to the details described above, MC160 workers were given a target elementary grade school level (1-4) and a sample story matching that grade level[5]. The intent was to produce a set of stories and questions that varied in difficulty so that research work can progress grade-by-grade if needed. However, we found little difference between grades in the corpus..

After gathering the stories, we manually curated the MC160 corpus by reading each story set and

[4] The latter two are the default AMT requirements.

[5] From http://www.englishforeveryone.org/.

1. We went to visit the Smith's at their house.
2. I altered their suits for them.
3. You're car is very old.
4. Jim likes to run, hike, and going kayaking.
5. He should of come to work on time.
6. I think its best to wash lots of apples.
7. Are people who write "ping" thinking of submarines?
8. Smoke filled the room, making it hard to breathe.
9. Alert yet aloof - that's you.
10. They wanted they're money back.
11. Hawks and eagles like to fly high in the sky.
12. Don't let her wear them down.
13. The cat particularly liked the greasy plate.
14. The company is less successful because we have less employees.
15. The hamster belongs to Sam and I.
16. No one landed on the air strip today.
17. He was very effected by her tears.
18. You are a tired piece of toast, metaphorically speaking.
19. Anne plays bass and sings.
20. Him and me met at the park.

Figure 2. Grammar test for qualifying workers.

correcting errors. The most common mistakes were grammatical, though occasionally questions and/or answers needed to be fixed. 66% of the stories have at least one correction. We provide both the curated and original corpuses in order to allow research on reading comprehension in the presence of grammar, spelling, and other mistakes.

3.5 MC500: Adding a Grammar Test

Though the construction of MC160 was successful, it requires a costly curation process which will not scale to larger data sets (although the curation was useful, both for improving the design of MC500, and for assessing the effectiveness of automated curation techniques). To more fully automate the process, we added two more stages: (1) A grammar test that automatically pre-screens workers for writing ability, and (2) a second Mechanical Turk task whereby new workers take the reading comprehension tests and rate their quality. We will discuss stage (2) in the next section.

The grammar test consisted of 20 sentences, half of which had one grammatical error (see Figure 2). The incorrect sentences were written using common errors such as *you're* vs. *your*, using *'s* to indicate plurality, incorrect use of tense, *it's* vs. *its*, *less* vs. *fewer*, *I* vs. *me*, etc. Workers were required to indicate for each sentence whether it was grammatically correct or not, and had to pass with at least 80% accuracy in order to qualify for the task. The 80% threshold was chosen to trade off worker quality with the rate at which the tasks would be completed; initial experiments using a threshold of 90% indicated that collecting 500 stories would take many weeks instead of days. Note that each worker is allowed to write at most 5 stores, so we required at least 100 workers to pass the qualification test.

	Quality (1-5)	About animals
No Grammar Test	3.2	73%
Grammar Test	**4.3**	**30%**

Table 1. Pre-screening workers using a grammar test improves both quality and diversity of stories. Both differences are significant using the two-tailed t-test ($p<0.05$ for quality and $p<0.01$ for animals).

To validate the use of the qualification test, we gathered 30 stories requiring the test (*qual*) and 30 stories without. We selected a random set of 20 stories (10 from each), hid their origin, and then graded the overall quality of the story and questions from 1-5, meaning *do not attempt to fix*, *bad but rescuable*, *has non-minor problems*, *has only minor problems*, and *has no problems*, respectively. Results are shown in Table 1. The difference is statistically significant ($p<0.05$, using the two-tailed t-test). The *qual* stories were also more diverse, with fewer of them about animals (the most common topic).

Additional Modifications: Based on our experience curating MC160, we also made the following modifications to the task. In order to eliminate trivially-answerable questions, we required that each answer be unique, and that either the correct answer did not appear in the story or, if it did appear, that at least two of the incorrect answers also appeared in the story. This is to prevent questions that are trivially answered by checking which answer appears in the story. The condition on whether the correct answer appears is to allow questions such as "*How many candies did Susan eat?*", where the total may never appear in the story, even though the information needed to derive it does. An answer is considered to appear in the story if at least half (rounded down) of its non-stopword

terms appear in the story (ignoring word endings). This check is done automatically and must be satisfied before the worker is able to complete the task. Workers could also bypass the check if they felt it was incorrect, by adding a special term to their answer.

We were also concerned that the sample story might bias the workers when writing the story set, particularly when designing questions that require multiple sentences to answer. So, we removed the sample story and grade level from the task.

Finally, in order to encourage more diversity of stories, we added *creativity terms*, a set of 15 nouns chosen at random from the allowed vocabulary set. Workers were asked to "*please consider*" using one or more of the terms in their story, but use of the words was strictly optional. On average, workers used 3.9 of the creativity terms in their stories.

4 Rating the Stories and Questions

In this section we discuss the crowd-sourced rating of story sets. We wished to ensure story set quality despite the fact that MC500 was only minimally manually curated (see below). Pre-qualifying workers with a grammar test was one step of this process. The second step was to have additional workers on Mechanical Turk both evaluate each story and take its corresponding test. Each story was evaluated in this way by 10 workers, each of whom provided scores for each of age-appropriateness (*yes/maybe/no*), grammaticality (*few/some/many* errors), and story clarity (*excellent/reasonable/poor*). When answering the four reading comprehension questions, workers could also mark a question as "unclear". Each story set was rated by 10 workers who were each paid $0.15 per set.

Since we know the purportedly correct answer, we can estimate worker quality by measuring what fraction of questions that worker got right. Workers with less than 80% accuracy (ignoring those questions marked as unclear) were removed from the set. This constituted just 4.1% of the raters and 4.2% of the judgments (see Figure 3). Only one rater appeared to be an intentional spammer, answering 1056 questions with only 29% accuracy. The others primarily judged only one story. Only one worker fell between, answering 336 questions with just 75% accuracy.

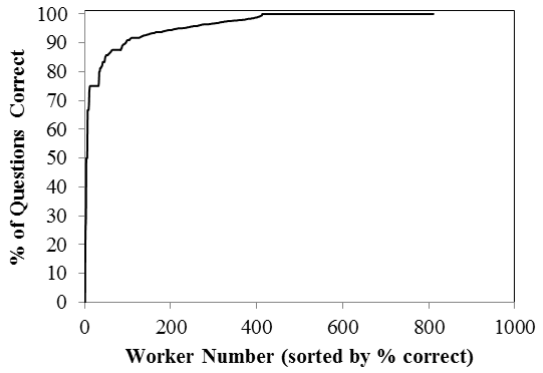

Figure 3. Just 4.1% of raters had an accuracy below 80% (constituting 4.2% of the judgments).

For the remaining workers (those who achieved at least 80% accuracy), we measured median story appropriateness, grammar, and clarity. For each category, stories for which less than half of the ratings were the best possible (e.g., *excellent* story clarity) were inspected and optionally removed from the data set. This required inspecting 40 (<10%) of the stories, only 2 of which were deemed poor enough to be removed (both of which had over half of the ratings all the way at the bottom end of the scale, indicating we could potentially have inspected many fewer stories with the same results). We also inspected questions for which at least 5 workers answered incorrectly, or answered "unclear". In total, 29 questions (<2%) were inspected. 5 were fixed by changing the question, 8 by changing the answers, 2 by changing both, 6 by changing the story, and 8 were left unmodified.

Note that while not fully automated, this process of inspecting stories and repairing questions took one person one day, so is still scalable to at least an order of magnitude more stories.

5 Dataset Analysis

In Table 2, we present results demonstrating the value of the grammar test and curation process. As expected, manually curating MC160 resulted in increased grammar quality and percent of questions answered correctly by raters. The goal of MC500 was to find a more scalable method to achieve the same quality as the curated MC160. As Table 2 shows, the grammar test improved story grammar quality from 1.70 to 1.77 (both uncurated). The rating and one-day curation process in-

Set	AgeAp	Clarity	Grammar	Correct
160	1.88	1.63	1.70	95.3
500	1.92	1.65	1.77	95.3
500 curated	1.94	**1.71**	1.79	**96.9**
160 curated	1.91	1.67	**1.84**‡	**97.7**

Table 2. Average age appropriateness, story clarity, grammar quality (0-2, with 2 being best), and percent of questions answered correctly by raters, for the original and curated versions of the data. Bold indicates statistical significance vs. the original version of the same set, using the two-sample t-test with unequal variance. The ‡ indicates the only statistical difference between 500 curated and 160 curated.

Corpus	Stories	Median writing time	Average Words Per:		
			Story	Question	Answer
MC160	160	26 min	204	8.0	3.4
MC500	500	20 min	212	7.7	3.4

Table 3. Corpus statistics for MC160 and MC500.

creases this to 1.79, whereas a fully manual curation results in a score of 1.84. Curation also improved the percent of questions answered correctly for both MC160 and MC500, but, unlike with grammar, there is no significant difference between the two curated sets. Indeed, the only statistically significant difference between the two is in grammar. So, the MC500 grammar test and curation process is a very scalable method for collecting stories of nearly the quality of the costly manual curation of MC160.

We also computed correlations between these measures of quality and various factors such as story length and time spent writing the story. On MC500, there is a mild correlation between a worker's grammar test score and the judged grammar quality of that worker's story (correlation of 0.24). Interestingly, this relation disappeared once MC500 was curated, likely due to repairing the stories with the worst grammar. On MC160, there is a mild correlation between the clarity and the number of words in the question and answer (0.20 and 0.18). All other correlations were below 0.15. These factors could be integrated into an estimate for age-appropriateness, clarity, and grammar, potentially reducing the need for raters.

Table 3 provides statistics on each corpus. MC160 and MC500 are similar in average number of words per story, question, and answer, as well as the median writing time. The most commonly used

Baseline Algorithms

Require: Passage P, set of passage words PW, i^{th} word in passage P_i, set of words in question Q, set of words in hypothesized answers $A_{1..4}$, and set of stop words U,
Define: $C(w) \coloneqq \sum_i \mathbb{I}(P_i = w)$;
Define: $IC(w) \coloneqq \log\left(1 + \frac{1}{C(w)}\right)$.

Algorithm 1 Sliding Window

for i = 1 to 4 **do**
$S = A_i \cup Q$
$$sw_i = \max_{j=1..|P|} \sum_{w=1..|S|} \begin{cases} IC(P_{j+w}) & if\ P_{j+w} \in S \\ 0 & otherwise \end{cases}$$
end for
return $sw_{1..4}$

Algorithm 2 Distance Based

for i = 1 to 4 **do**
$S_Q = (Q \cap PW) \setminus U$
$S_{A_i} = \left((A_i \cap PW) \setminus Q\right) \setminus U$
if $|S_Q| = 0$ or $|S_{A_i}| = 0$
$d_i = 1$
else
$d_i = \frac{1}{|P|-1} \min_{q \in S_Q, a \in S_{A_i}} d_P(q, a)$,
where $d_P(q, a)$ is the minimum number of words between an occurrence of q and an occurrence of a in P, plus one.
end if
end for
return $d_{1..4}$

Algorithm SW

Return $\arg\max_i sw_{1..4}$

Algorithm SW+D

Return $\arg\max_i sw_{1..4} - d_{1..4}$

Figure 4. The two lexical-based algorithms used for the baselines.

nouns in MC500 are: *day*, *friend*, *time*, *home*, *house*, *mother*, *dog*, *mom*, *school*, *dad*, *cat*, *tree*, and *boy*. The stories vary widely in theme. The first 10 stories of the randomly-ordered MC500 set are about: travelling to Miami to visit friends, waking up and saying hello to pets, a bully on a schoolyard, visiting a farm, collecting insects at Grandpa's house, planning a friend's birthday party, selecting clothes for a school dance, keeping animals from eating your ice cream, animals ordering food, and adventures of a boy and his dog.

MC160	Train and Dev: 400 Q's		Test: 240 Q's	
	SW	SW+D	SW	SW+D
Single	59.46	68.11	64.29	75.89
Multi	59.53	67.44	48.44	57.81
All	*59.50*	*67.75*	*55.83*	*66.25*

Table 4. Percent correct for the multiple choice questions for MC160. SW: sliding window algorithm. SW+D: combined results with sliding window and distance based algorithms. Single/Multi: questions marked by worker as requiring a single/multiple sentence(s) to answer. All differences between SW and SW+D are significant ($p<0.01$ using the two-tailed paired t-test).

MC500	Train and Dev: 1400 Q's		Test: 600 Q's		All
	SW	SW+D	SW	SW+D	SW+D
Single	55.13	61.77	51.10	57.35	60.44
Multi	49.80	55.28	51.83	56.10	55.53
All	*52.21*	*58.21*	*51.50*	*56.67*	*57.75*

Table 5. Percent correct for the multiple choice questions for MC500, notation as above. All differences between SW and SW+D are significant ($p<0.01$, tested as above).

We randomly divided MC160 and MC500 into train, development, and test sets of 70, 30, and 60 stories and 300, 50, and 150 stories, respectively.

6 Baseline System and Results

We wrote two baseline systems, both using only simple lexical features. The first system used a sliding window, matching a bag of words constructed from the question and hypothesized answer to the text. Since this ignored long range dependencies, we added a second, word-distance based algorithm. The distance-based score was simply subtracted from the window-based score to arrive at the final score (we tried scaling the distance score before subtraction but this did not improve results on the MC160 train set). The algorithms are summarized in Figure 4. A coin flip is used to break ties. The use of inverse word counts was inspired by TF-IDF.

Results for MC160 and MC500 are shown in Table 4 and Table 5. The MC160 train and development sets were used for tuning. The baseline algorithm was authored without seeing any portion of MC500, so both the MC160 test set and all of MC500 were used for testing (although we nevertheless report results on the train/test split). Note that adding the distance based algorithm improved accuracy by approximately 10% absolute on MC160 and approximately 6% on MC500. Overall, error rates on MC500 are higher than on MC160, which agrees with human performance (see Table 2), suggesting that MC500's questions are more difficult.

	MC160 Test	MC500 Test
Baseline (SW+D)	66.25	56.67
RTE	59.79‡	53.52
Combined	67.60	60.83‡

Table 6. Percent correct for MC160 and MC500 test sets. The ‡ indicates statistical significance vs. baseline ($p<0.01$ using the two-tailed paired t-test). MC160 combined vs. baseline has p-value 0.063.

7 Recognizing Textual Entailment Results

We also tried using a "recognizing textual entailment" (RTE) system to answer MCTest questions. The goal of RTE (Dagan et al., 2005) is to determine whether a given statement can be inferred from a particular text. We can cast MCTest as an RTE task by converting each question-answer pair into a statement, and then selecting the answer whose statement has the highest likelihood of being entailed by the story. For example, in the sample story given in Figure 1, the second question can be converted into four statements (one for each answer), and the RTE system should select the statement "James pulled pudding off of the shelves in the grocery store" as the most likely one.

For converting question-answer pairs to statements, we used the rules employed in a web-based question answering system (Cucerzan and Agichtein, 2005). For RTE, we used BIUTEE (Stern and Dagan, 2011), which performs better than the median system in the past four RTE competitions. We ran BIUTEE both in its default configuration, as well as with its optional additional data sources (FrameNet, ReVerb, DIRT, and others as found on the BIUTEE home page). The default configuration performed better so we present its results here. The results in Table 6 show that the RTE method performed worse than the baseline.

We also combined the baseline and RTE system by training BIUTEE on the train set and using the development set to optimize a linear combination of BIUTEE with the baseline; the combined system outperforms either component system on MC500.

It is possible that with some tuning, an RTE system will outperform our baseline system. Nevertheless, these RTE results, and the performance of the baseline system, both suggest that the reading comprehension task described here will not be trivially solved by off-the-shelf techniques.

8 Making Data and Results an Ongoing Resource

Our goal in constructing this data is to encourage research and innovation in the machine comprehension of text. Thus, we have made both MC160 and MC500 freely available for download at http://research.microsoft.com/mct. To our knowledge, these are the largest copyright-free reading comprehension data sets publicly available. To further encourage research on these data, we will be continually updating the webpage with the best-known published results to date, along with pointers to those publications.

One of the difficulties in making progress on a particular task is implementing previous work in order to apply improvements to it. To mitigate this difficulty, we are encouraging researchers who use the data to (optionally) provide per-answer scores from their system. Doing so has three benefits: (a) a new system can be measured in the context of the errors made by the previous systems, allowing each research effort to incrementally add useful functionality without needing to also re-implement the current state-of-the-art; (b) it allows system performance to be measured using paired statistical testing, which will substantially increase the ability to determine whether small improvements are significant; and (c) it enables researchers to perform error analysis on any of the existing systems, simplifying the process of identifying and tackling common sources of error. We will also periodically ensemble the known systems using standard machine learning techniques and make those results available as well (unless the existing state-of-the-art already does such ensembling).

The released data contains the stories and questions, as well as the results from workers who rated the stories and took the tests. The latter may be used, for example, to measure machine performance vs. human performance on a per-question basis (i.e., does your algorithm make similar mistakes to humans?), or vs. the judged clarity of each story. The ratings, as well as whether a question needs multiple sentences to answer, should typically only be used in evaluation, since such information is not generally available for most text. We will also provide an anonymized author id for each story, which could allow additional research such as using other works by the same author when understanding a story, or research on authorship attribution (e.g., Stamatatos, 2009).

9 Future Work

We plan to use this dataset to evaluate approaches for machine comprehension, but are making it available now so that others may do the same. If MCTest is used we will collect more story sets and will continue to refine the collection process. One interesting research direction is ensuring that the questions are difficult enough to challenge state-of-the-art techniques as they develop. One idea for this is to apply existing techniques automatically during story set creation to see whether a question is too easily answered by a machine. By requiring authors to create difficult questions, each data set will be made more and more difficult (but still answerable by humans) as the state-of-the-art methods advance. We will also experiment with timing the raters as they answer questions to see if we can find those that are too easy for people to answer. Removing such questions may increase the difficulty for machines as well. Additionally, any divergence between how easily a person answers a question vs. how easily a machine does may point toward new techniques for improving machine comprehension; we plan to conduct research in this direction as well as make any such data available for others.

10 Conclusion

We present the MCTest dataset in the hope that it will help spur research into the machine comprehension of text. The metric (the accuracy on the question sets) is clearly defined, and on that metric, lexical baseline algorithms only attain approximately 58% correct on test data (the MC500 set) as

opposed to the 100% correct that the majority of crowd-sourced judges attain. A key component of MCTest is the scalable design: we have shown that data whose quality approaches that of expertly curated data can be generated using crowd sourcing coupled with expert correction of worker-identified errors. Should MCTest prove useful to the community, we will continue to gather data, both to increase the corpus size, and to keep the test sets fresh. The data is available at http://research.microsoft.com/mct and any submitted results will be posted there too. Because submissions will be requested to include the score for each test item, researchers will easily be able to compare their systems with those of others, and investigation of ensembles comprised of components from several different teams will be straightforward. MCTest also contains supplementary material that researchers may find useful, such as worker accuracies on a grammar test and crowd-sourced measures of the quality of their stories.

Acknowledgments

We would like to thank Silviu Cucerzan and Lucy Vanderwende for their help with converting questions to statements and other useful discussions.

References

M. Agarwal and P. Mannem. 2011. Automatic Gap-fill Question Generation from Text Books. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*, 56–64.

E. Breck, M. Light, G.S.Mann, E. Riloff, B. Brown, P. Anand, M. Rooth M. Thelen. 2001. Looking under the hood: Tools for diagnosing your question answering engine. In *Proceedings of the workshop on Open-domain question answering*, 12, 1-8.

E. Charniak. 1972. Toward a Model of Children's Story Comprehension. *Technical Report*, 266, MIT Artificial Intelligence Laboratory, Cambridge, MA.

P. Clark, P. Harrison, and X. Yao. An Entailment-Based Approach to the QA4MRE Challenge. 2012. In *Proceedings of the Conference and Labs of the Evaluation Forum (CLEF) 2012*.

S. Cucerzan and E. Agichtein. 2005. Factoid Question Answering over Unstructured and Structured Content on the Web. In *Proceedings of the Fourteenth Text Retrieval Conference (TREC)*.

I. Dagan, O. Glickman, and B. Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In J. Quiñonero-Candela, I. Dagan, B. Magnini, F. d'Alché-Buc (Eds.), Machine Learning Challenges. Lecture Notes in Computer Science, Vol. 3944, pp. 177-190, Springer.

D. Goldwasser, R. Reichart, J. Clarke, D. Roth. 2011. Confidence Driven Unsupervised Semantic Parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 1486-1495.

E. Grois and D.C. Wilkins. 2005. Learning Strategies for Story Comprehension: A Reinforcement Learning Approach. In *Proceedings of the Twenty Second International Conference on Machine Learning*, 257-264.

S.M. Harabagiu, S.J. Maiorano, and M.A. Pasca. 2003. Open-Domain Textual Question Answering Techniques. *Natural Language Engineering*, 9(3):1-38. Cambridge University Press, Cambridge, UK.

L. Hirschman, M. Light, E. Breck, and J.D. Burger. 1999. Deep Read: A Reading Comprehension System. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, 325-332.

J. Horton and L. Chilton. 2010. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 209-218.

V. Kuperman, H. Stadthagen-Gonzalez, M. Brysbaert. 2012. Age-of-acquisition ratings for 30,000 English words. *Behavior Research Methods*, 44(4):978-990.

J.L. Leidner, T. Dalmas, B. Webber, J. Bos, C. Grover. 2003. Automatic Multi-Layer Corpus Annotation for Evaluating Question Answering Methods: CBC4Kids. In *Proceedings of the 3rd International Workshop on Linguistically Interpreted Corpora*.

E.T. Mueller. 2010. Story Understanding Resources. *http://xenia.media.mit.edu/~mueller/storyund/storyres.html*.

G. Paolacci, J. Chandler, and P. Iperirotis. 2010. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*. 5(5):411-419.

E. Stamatatos. 2009. A survey of modern authorship attribution methods. *J. Am. Soc. Inf. Sci.*, 60:538–556.

A. Stern and I. Dagan. 2011. A Confidence Model for Syntactically-Motivated Entailment Proofs. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*.

L.R. Tang and R.J. Mooney. 2001. Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*, 466-477.

G. Tur, D. Hakkani-Tur, and L.Heck. 2010. What is left to be understood in ATIS? *Spoken Language Technology Workshop*, 19-24.

E.M. Voorhees and D.M. Tice. 1999. The TREC-8 Question Answering Track Evaluation. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*.

B. Wellner, L. Ferro, W. Greiff, and L. Hirschman. 2005. Reading comprehension tests for computer-based understand evaluation. *Natural Language Engineering*, 12(4):305-334. Cambridge University Press, Cambridge, UK.

J.M. Zelle and R.J. Mooney. 1996. Learning to Parse Database Queries using Inductive Logic Programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*, 1050-1055.

L.S. Zettlemoyer and M. Collins. 2009. Learning Context-Dependent Mappings from Sentences to Logical Form. In *Proceedings of the 47th Annual Meeting of the Association for Computation Linguistics (ACL)*, 976-984.

G. Zweig and C.J.C. Burges. 2012. A Challenge Set for Advancing Language Modeling. In *Proceedings of the Workshop on the Future of Language Modeling for HLT, NAACL-HLT*.