A Strong Lexical Matching Method for the Machine Comprehension Test

Ellery Smith, Nicola Greco, Matko Bošnjak, Andreas Vlachos
Department of Computer Science
University College London
{e.smith,n.greco,m.bosnjak,a.vlachos}@cs.ucl.ac.uk

Abstract

Machine comprehension of text is the overarching goal of a great deal of research in natural language processing. The Machine Comprehension Test (Richardson et al., 2013) was recently proposed to assess methods on an open-domain, extensible, and easy-to-evaluate task consisting of two datasets. In this paper we develop a lexical matching method that takes into account multiple context windows, question types and coreference resolution. We show that the proposed method outperforms the baseline of Richardson et al. (2013), and despite its relative simplicity, is comparable to recent work using machine learning. We hope that our approach will inform future work on this task. Furthermore, we argue that MC500 is harder than MC160 due to the way question answer pairs were created.

1 Introduction

Machine comprehension of text is the central goal in NLP. The academic community has proposed a variety of tasks, such as information extraction (Sarawagi, 2008), semantic parsing (Mooney, 2007) and textual entailment (Androutsopoulos and Malakasiotis, 2010). However, these tasks assess performance on each task individually, rather than on overall progress towards machine comprehension of text.

To this end, Richardson et al. (2013) proposed the Machine Comprehension Test (MCTest), a new challenge that aims at evaluating machine comprehension. It does so through an open-domain multiple-choice question answering task on fictional stories requiring the common sense reasoning typical of a 7-year-old child. It is easy to evaluate as it consists of multiple choice questions. Richardson et al. (2013) also showed how the creation of stories and questions can be crowdsourced efficiently, constructing two datasets for the task, namely MC160 and MC500. In addition, the authors presented a lexical matching baseline which is combined with the textual entailment recognition system BIUTEE (Stern and Dagan, 2011).

In this paper we develop an approach based on lexical matching which we extend by taking into account the type of the question and coreference resolution. These components improve the performance on questions that are difficult to handle with pure lexical matching. When combined with BIUTEE, we achieved 74.27% accuracy on MC160 and 65.96% on MC500, which are significantly better than those reported by Richardson et al. (2013). Despite the simplicity of our approach, these results are comparable with the recent machine learning-based approaches proposed by Narasimhan and Barzilay (2015), Wang et al. (2015) and Sachan et al. (2015).

Furthermore, we examine the types of questions and answers in the two datasets. We argue that some types are relatively simple to answer, partly due to the limited vocabulary used, which explains why simple lexical matching methods can perform well. On the other hand, some questions require understanding of higher level concepts such as those of the story and its characters, and/or require inference. This is still beyond the scope of current NLP systems. However, we believe our analysis will be useful in developing new methods and datasets for the task. To that extent, we will make our code and analysis publicly available.[1]

2 Task description

MCTest is an open-domain multiple-choice question answering task on fictional stories consisting of two datasets, MC160 and MC500. The

[1]http://github.com/elleryjsmith/UCLMCTest

It was a terrible day to live in the zoo again for Pauly. It wasn't a terrible day for Zip, the monkey next to him, or Garth, the giraffe down the sidewalk, or Pat, the alligator in the pond, or for Bam the prairie dog, but it was a terrible day in the monkey cage for Pauly. Pauly didn't feel he belonged in the monkey cage because he wasn't a monkey. He was a sailor who had visited the zoo on vacation and fallen asleep on a bench right before closing time. The zoo worker saw how hairy he was and thought he was a monkey that had escaped from his cage, so they put him in a cage.

1. Where was Pauly when the zoo worker saw him?
A) Looking at the monkeys
B) Sailing on a boat
C) Asleep on a bench
D) Walking down a path

2. Why did Pauly feel he didn't belong in the monkey cage?
A) Because he wasn't a monkey
B) Because he was a zoo worker
C) Because he didn't fall asleep
D) Because he was hairy

Figure 1: An excerpt from story *mc500.train.44*

two datasets contain 160 and 500 stories respectively, with 4 questions per story, and 4 candidate answers per question (Figure 1). All stories and questions were crowd-sourced using Amazon Mechanical Turk.[2] MC160 was manually curated by Richardson et al., while MC500 was curated by crowdworkers. Both datasets are divided into training, development, and test sets. All development was conducted on the training and development sets; the test sets were used only to report the final results.

3 Scoring function

Richardson et al. (2013) proposed a sliding window algorithm that ranks the answers by forming the bag-of-words vector of each answer paired with the question text and then scoring them according to their overlap with the story text. We propose a modified version of this algorithm, which combines the scores across a range of window sizes.

More concretely, the algorithm of Richardson et al. (2013) passes a sliding window over the story, size of which is equal to the number of words in the question-answer pair. The highest overlap score between a story text window and the question-answer pair is taken as the score for the answer. Therefore, their algorithm makes a single pass over the story text per answer. In comparison, our system scores each answer by making multiple passes and summing the obtained scores. Concretely, on the first pass, we set the sliding window size to 2 tokens, and increment this size on each subsequent pass, up to a length of 30 tokens. We then combine this score with the overall number of matches of the question-answer pair across the story as a whole. This enables our algorithm to catch long-distance relations in the story. Similar to Richardson et al. (2013), we use a linear combination of this score with their distance-based scoring function, and we weigh tokens with their inverse document frequencies in each individual story.

By itself, this simple enhancement gives substantial improvements over the MSR baseline as shown in Table 1 (*Enhanced SW+D*), as it measures the overlap of the question-answer pair with multiple portions of the story text.

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

4 Incorporating linguistic analyses

We build upon our enhanced scoring function using stemming, rules taking into account the type of the question, and coreference. The improvements due to each of these components are presented in Table 1, and we discuss the application of coreference and the rules used in more detail in the following subsections.

	MC160	**MC500**
MSR SW+D	69.43%	63.01%
Enhanced SW+D	72.65%	63.57%
+*Coreference Rules*	+2.38%	+1.36%
+*Negation*	+0.75%	+0.36%
+*Stemming*	+2.04%	+0.50%
Final system	**75.77%**	**65.43%**

Table 1: Performance improvements on combined train and dev sets.

4.1 Coreference resolution

The entities mentioned in MCTest stories are frequently referred to by anaphoric expressions throughout the story and the questions, which is ignored by the described scoring function. Therefore, we substituted each mention in a coreference chain with its representative mention, applied the scoring function on the processed text and added the score to the original one. The chains and their represenatative mentions were obtained using the Stanford CoreNLP toolkit (Manning et al., 2014). We found that coreference improved performance on some question types, but decreased performance on others. Thus we developed a set of question rules in order to apply it selectively, which we discuss in the next section.

4.2 Rules

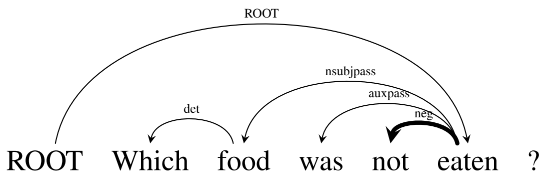

Figure 2: Dependency tree to detect negation

To account for the variety of questions in MCTest, we developed a set of rules to handle certain question types differently. To this purpose, we created rules which detect numerical, temporal, narrative and negation-based questions, and additionally partition questions by their *wh*-word.

By partitioning questions in different types, we found that *who* questions, which primarily deal with identifying a character in the story, benefit from the use of coreference chains described in the previous section. In addition, performance in questions aimed at selecting an appropriate noun, such as *which*, *where* or numerical questions, also improved with coreference. However, other question types, such as *why* questions, or questions concerning the story narrative, did not register any consistent improvement, and we opted not to use co-reference for them. This selective application of co-reference resulted in improvements on both datasets (Table 1).

We also identified negation questions as requiring special treatment. Some negation questions are trivially solvable by selecting an answer which does not appear in the text. However, our proposed function that scores answers according to the lexical overlap with the story text is unlikely to score answers not appearing in text highly. Motivated by this observation we invert the score of each word when a question with a negated root verb was detected, e.g. *"What did James not eat for dinner?"*, using Stanford Typed Dependencies (De Marneffe and Manning, 2008), as depicted in Figure 2. Due to this inversion a higher lexical overlap results in a lower score, improving accuracy on both MC160 and MC500 (+*negation* in Table 1)

In a similar fashion we detected numerical questions based on the presence of a POS tag for a cardinal number in either the question or any of the answers choices. Questions concerning the story's narrative (e.g. *"Which is the first character mentioned in the story"*) were detected using keywords (e.g. *character*, *book*, etc.). Additionally, we detected temporal questions such as *"What did Jane do before she went home?"* by the presence of a temporal modifier or temporal prepositions (e.g. *before*, *while*, etc.). Then we attempted to account for them by searching the text for the sentence indicating that she had gone home and reducing the weight for all subsequent sentences. However, since the improvements due to these rules were negligible, we did not include them in our final system. Nevertheless, these rules were helpful in analyzing problem areas in the datasets, as discussed in Section 6.

5 Results

We evaluated our system on MC160 and MC500 test sets and the results are shown in Table 2. Our proposed baseline outperforms the baseline of Richardson et al. (2013) by 4 and 3 points in accuracy on MC160 and MC500 respectively.[3] Our system is comparable to the MSR baseline with the RTE system BIUTEE (Stern and Dagan, 2011). If we linearly combine the RTE scores used in the MSR baseline with our method, we achieve 5 and 2.5 accuracy points higher than the best results achieved by Richardson et al. (2013).

Concurrently with ours, three other approaches to solving MCTest were developed and subsequently published a few months before our method. Narasimhan and Barzilay (2015) presented a discourse-level approach, which chooses an answer by utilising relations between sentences chosen as important. Despite is simplicity, our method is comparable in performance, suggesting that better lexical matching could help improve their model. Sachan et al. (2015) treated MCTest as a structured prediction problem, searching for a latent structure connecting the question, answer and the text, dubbed the answer-entailing structure. Their model performs better on MC500 (was

[3]We consider the updated MSR algorithms and results, together with partial credit accuracies, provided at `http://research.microsoft.com/en-us/um/redmond/projects/mctest/results.html`

	MC160	**MC500**
MSR SW+D	68.02%	59.93%
Final system	72.19%	62.67%*
MSR SW+D + RTE	69.27%	63.33%
Final system + RTE	74.27%*	65.96%*
Narasimhan & Barzilay	73.23%	63.75%
Sachan et al.	-	67.83%
Wang et al.	75.27%	69.94%

Table 2: Performance on the MC160 and MC500 test sets, including the results of all previous work. * denotes statistically significant ($p < 0.05$) improvement using McNemar's test, with respect to the MSR baseline (SW+D)

Category	**MC160**	**MC500**
What	76.98% (126)	68.77% (317)
Who	71.43% (28)	58.44% (77)
When	80.00% (5)	100.00% (7)
How	72.62% (21)	50.58% (43)
Which	66.67% (6)	40.00% (25)
Where	91.67% (12)	68.97% (58)
Why	72.97% (37)	80.65% (62)
Whose	-	66.67% (3)
Other	0.00% (5)	25.00% (8)
Negation	53.33% (15)	34.48% (29)
Temporal	58.82% (17)	56.41% (39)
Numerical	69.32% (22)	48.26% (43)
Narrative	81.82% (11)	58.41% (26)
Quantifiers	70.00% (20)	53.38% (37)
Single	78.79% (112)	69.12% (272)
Multi	70.31% (128)	63.34% (328)

Table 3: Performance of our final system + RTE per question type on the test sets. The number of relevant questions is in parentheses.

not tested on MC160), however the strength of our model is obtaining comparable results with a much simpler model. The work of Wang et al. (2015) is the most similar to ours, in the sense that they combine a baseline feature set with more advanced linguistic analyses, namely syntax, frame semantics, coreference, and word embeddings. Instead of a rule-based approach, they combine them through a latent-variable classifier achieving the current state-of-the-art performance on MCTest.

6 Discussion

Using the question-filtering rules mentioned in Section 4.2, we obtained individual accuracy scores per question type for the final system combined with RTE (Table 3). Note that these types are in three groups: i) wh-word questions (disjoint, questions without an wh-word are in *Other*), ii) classes of questions requiring non-trivial linguistic analysis and reasoning (not disjoint, not all questions considered), and iii) questions originally classified by crowdworkers, classifying whether the question can be answered by a single or multiple sentences in the story (disjoint).

Compared to the wh-word question type results of Narasimhan and Barzilay (2015), our approach performs better primarily on *why* questions (72.97% and 80.65% vs. 59.45% and 69.35% on MC160 and MC500 respectively) and slightly better on *how*, *when* and *what* questions on MC500. Additionally, our system is more successful in questions requiring multiple sentences to be answered correctly (70.31% and 63.3%vs. 65.23% and 59.9% on MC160 and MC500 respectively).

If we remove the RTE component from our system, the performance on relatively simple question types such as *what*, *who* and *where* remains practically the same, thus confirming that our approach can handle simple questions well. On the other hand, the performance on *why* questions drops without RTE, thus stressing the need for deeper text understanding.

There are several clear deficiencies in certain question types, particularly in handling negation. These errors provide a broad overview of the cases in which simple lexical techniques are not sufficient to determine the correct answer.

Many numerical questions, particularly in MC500, require the use of simple algebra over story elements, including counting characters and objects, and understanding temporal order. One question even requires calculating the probability of an event occurring, while another one calls for complex volumetric calculation. Answering questions such as these is beyond the capabilities of a lexical algorithm, and accuracies in this category are worse than on all questions. Additionally, lexical algorithms such as ours, which ignore predicate-argument structure, perform worse in the presence of quantifiers.

In MC500, the performance of our system on more abstract questions, concerning the overall narrative of the story, also demonstrates a significant inadequacy of lexical-based algorithms. Questions such as "*What was the first character mentioned in the story?*", which relate to the overall narrative flow of the passage, or questions concerning the state of the story environment, such as "*Where is the story set?*", are difficult to solve without a system which understands the concept of a story. Typical question-answering methods would also struggle here.

Another type of challenging question are those which require an implicit temporal understanding of the text, i.e. questions concerning time without using a temporal modifier. For example, given a story which states that "*John is at the beach*", then later "*John went home*", a question such as "*What did John do at home?*" would prove itself difficult for traditional methods to answer. These questions are difficult to identify automatically by the form of the question alone, thus we cannot provide accuracies for them.

Our results confirm that it is easier to achieve better performance on MC160 with simple lexical techniques, while the MC500 has proved more resilient to the same improvements. We also observed that the MC500 registers smaller improvements in accuracy when adding components such as co-reference. This is a consequence of the design and curation process of the MC500 dataset, which stipulated that answers must not be contained directly within the story text, or if they are, that two or more misleading choices included.

Richardson et al. (2013) demonstrate that the MC160 and MC500 have similar ratings for clarity and grammar, and that humans perform equally well on both. However, in many cases MC500 appears to be designed in such a way to confuse lexical algorithms and encourage the use of more sophisticated techniques necessary to deal with phenomena such as elimination questions, negation, and common knowledge not explicitly written in the story.

7 Related work

The use of shallow methods for machine comprehension has been explored in previous work, for example Hirschman et al. (1999) used a bag-of-words to match question-answer pairs to sentences in the text, and choose the best pair with the best matching sentence. As discussed in our analysis, such systems cannot handle well questions involving negation and quantification. Numerical questions, which we found to be particularly challenging, have been the focus of recent work on algebra word problems (Kushman et al., 2014) for which dedicated systems have been developed.

MacCartney et al. (2006) demonstrated that a large set of rules can be used to recognize valid textual entailments. These consider phenomena such as polarity and quantification, similar to those we used in our analysis of the MCTest datasets. More complex methods, which attempt deeper modeling of text include Natural Logic (Angeli and Manning, 2014) and Combinatorial Categorial Grammars (Lewis and Steedman, 2013) combined with distributional models. While promising, these approaches have been developed primarily on sentence-level tasks, thus the stories in MCTest are likely to present additional challenges.

The recently proposed class of methods called Memory Network (Weston et al., 2014), uses neural networks and external memory to answer a simpler comprehension task. Though quite successful on toy tasks, those methods cannot yet be applied to MCTest as they require much larger training datasets than the ones available for this task.

A recent approach by Hermann et al. (2015) uses attention-based recurrent neural networks to attack the problem of machine comprehension. In this work, the authors show how to generate large amounts of data for machine comprehension exploiting news websites, and how to use novel architectures in deep learning to solve the task. However, due to the need for a large dataset for training, and the focus only on questions that take entities as answers, this approach has not been applied to MCTest.

8 Conclusion

In this paper we developed an approach to MCTest that combines lexical matching with simple linguistic analysis. We evaluated it on the two MCTest datasets, MC160 and MC500, and we showed that it improves upon the original baseline by 4 and 3 percentage points respectively, while being comparable to more complex approaches. In addition, our analysis highlighted the challenges involved and in particular in the MC500 dataset.

Acknowledgments

We would like to thank Tom Brown for his contributions in the early stages of this work.

References

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, pages 135–187.

Gabor Angeli and Christopher D Manning. 2014. Naturalli: Natural logic inference for common sense reasoning. In *Proceedings of EMNLP*.

Marie-Catherine De Marneffe and Christopher D Manning. 2008. Stanford typed dependencies manual. *URL http://nlp. stanford. edu/software/dependencies manual. pdf*.

Karl M. Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. *ArXiv e-prints*.

Lynette Hirschman, Marc Light, Eric Breck, and John D Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of ACL*, pages 325–332.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of ACL*, pages 271–281.

Mike Lewis and Mark Steedman. 2013. Combined distributional and logical semantics. *TACL*, 1:179–192.

Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of the HLT-NAACL*, pages 41–48.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL: System Demonstrations*, pages 55–60.

Raymond J Mooney. 2007. Learning for semantic parsing. In *Computational Linguistics and Intelligent Text Processing*, pages 311–324.

Karthik Narasimhan and Regina Barzilay. 2015. Machine comprehension with discourse relations. In *Proceedings of ACL*.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of EMNLP*, pages 193–203.

Mrinmaya Sachan, Avinava Dubey, Eric P Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *Proceedings of ACL*.

Sunita Sarawagi. 2008. Information extraction. *Foundations and trends in databases*, 1(3):261–377.

Asher Stern and Ido Dagan. 2011. A Confidence Model for Syntactically-Motivated Entailment Proofs. In *Proceedings of Recent Advances in Natural Language Processing, (RANLP)*, pages 455–462.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In *Proceedings of ACL: Short papers*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.