MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, Manfred Pinkal
Saarland University
Saarbrücken, Germany
{simono|ashutosh|mroth|stth|pinkal}@coli.uni-saarland.de

Abstract

We introduce a large dataset of narrative texts and questions about these texts, intended to be used in a machine comprehension task that requires reasoning using commonsense knowledge. Our dataset complements similar datasets in that we focus on stories about everyday activities, such as going to the movies or working in the garden, and that the questions require commonsense knowledge, or more specifically, *script knowledge*, to be answered. We show that our mode of data collection via crowdsourcing results in a substantial amount of such inference questions. The dataset forms the basis of a shared task on commonsense and script knowledge organized at SemEval 2018 and provides challenging test cases for the broader natural language understanding community.

Keywords: machine comprehension, reading comprehension, commonsense knowledge, script knowledge

1. Introduction

Ambiguity and implicitness are inherent properties of natural language that cause challenges for computational models of language understanding. In everyday communication, people assume a shared common ground which forms a basis for efficiently resolving ambiguities and for inferring implicit information. Thus, recoverable information is often left unmentioned or underspecified. Such information may include encyclopedic and commonsense knowledge. This work focuses on commonsense knowledge about everyday activities, so-called *scripts*.

This paper introduces a dataset to evaluate natural language understanding approaches with a focus on interpretation processes requiring inference based on commonsense knowledge. In particular, we present *MCScript*, a dataset for assessing the contribution of script knowledge to machine comprehension. Scripts are sequences of events describing stereotypical human activities (also called scenarios), for example baking a cake or taking a bus (Schank and Abelson, 1975). To illustrate the importance of script knowledge, consider Example (1):

(1) The waitress brought Rachel's order. She ate the food with great pleasure.

Without using commonsense knowledge, it may be difficult to tell who ate the food: Rachel or the waitress. In contrast, if we utilize commonsense knowledge, in particular, script knowledge about the EATING IN A RESTAURANT scenario, we can make the following inferences: Rachel is most likely a customer, since she received an order. It is usually the customer, and not the waitress, who eats the ordered food. So *She* most likely refers to Rachel.

Various approaches for script knowledge extraction and processing have been proposed in recent years. However, systems have been evaluated for specific aspects of script knowledge only, such as event ordering (Modi and Titov, 2014a; Modi and Titov, 2014b), event paraphrasing (Regneri et al., 2010; Wanzare et al., 2017) or event prediction (namely, the narrative cloze task (Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2009; Pichotta and Mooney, 2014; Pichotta and Mooney, 2016; Modi, 2016)). These evaluation methods lack a clear connection to real-world tasks. Our MCScript dataset provides an extrinsic evaluation framework, based on text comprehension involving commonsense knowledge. This framework makes it possible to assess system performance in a multiple-choice question answering setting, without imposing any specific structural or methodical requirements.

T I wanted to plant a tree. I went to the home and garden store and picked a nice oak. Afterwards, I planted it in my garden.

Q1 What was used to dig the hole?
a. a shovel b. his bare hands

Q2 When did he plant the tree?
a. after watering it b. after taking it home

Figure 1: An example for a text snippet with two reading comprehension questions.

MCScript is a collection of (1) narrative texts, (2) questions of various types referring to these texts, and (3) pairs of answer candidates for each question. It comprises approx. 2,100 texts and a total of approx. 14,000 questions. Answering a substantial subset of questions requires knowledge beyond the facts mentioned in the text, i.e. it requires inference using commonsense knowledge about everyday activities. An example is given in Figure 1. For both questions, the correct choice for an answer requires commonsense knowledge about the activity of planting a tree, which goes beyond what is mentioned in the text. Texts, questions, and answers were obtained through crowdsourcing. In order to ensure high quality, we manually validated and filtered the dataset. Due to our design of the data ac-

quisition process, we ended up with a substantial subset of questions that require commonsense inference (27.4%).

2. Corpus

Machine comprehension datasets consist of three main components: texts, questions and answers. In this section, we describe our data collection for these 3 components. We first describe a series of pilot studies that we conducted in order to collect commonsense inference questions (Section 2.1.). In Section 2.2., we discuss the resulting data collection of questions, texts and answers via crowdsourcing on Amazon Mechanical Turk[1] (henceforth *MTurk*). Section 2.3. gives information about some necessary postprocessing steps and the dataset validation. Lastly, Section 2.4. gives statistics about the final dataset.

2.1. Pilot Study

As a starting point for our pilots, we made use of texts from the *InScript* corpus (Modi et al., 2016), which provides stories centered around everyday situations (see Section 2.2.2.). We conducted three different pilot studies to determine the best way of collecting questions that require inference over commonsense knowledge:

The most intuitive way of collecting reading comprehension questions is to show texts to workers and let them formulate questions and answers on the texts, which is what we tried internally in a *first pilot*. Since our focus is to provide an evaluation framework for inference over commonsense knowledge, we manually assessed the number of questions that indeed require common sense knowledge. We found too many questions and answers collected in this manner to be lexically close to the text.

In a *second pilot*, we investigated the option to take the questions collected for one text and show them as questions for another text of the same scenario. While this method resulted in a larger number of questions that required inference, we found the majority of questions to not make sense at all when paired with another text. Many questions were specific to a text (and not to a scenario), requiring details that could not be answered from other texts.

Since the two previous pilot setups resulted in questions that centered around the texts themselves, we decided for a *third pilot* to not show workers any specific texts at all. Instead, we asked for questions that centered around a specific script scenario (e.g. EATING IN A RESTAURANT). We found this mode of collection to result in questions that have the right level of specificity for our purposes: namely, questions that are related to a scenario and that can be answered from different texts (about that scenario), but for which a text does not need to provide the answer explicitly. The next section will describe the mode of collection chosen for the final dataset, based on the third pilot, in more detail.

2.2. Data Collection

2.2.1. Scenario Selection

As mentioned in the previous section, we decided to base the question collection on script scenarios rather than specific texts. As a starting point for our data collection, we use scenarios from three script data collections (Regneri et al., 2010; Singh et al., 2002; Wanzare et al., 2016). Together, these resources contain more than 200 scenarios. To make sure that scenarios have different complexity and content, we selected 80 of them and came up with 20 new scenarios. Together with the 10 scenarios from InScript, we end up with a total of 110 scenarios.

2.2.2. Texts

For the collection of texts, we followed Modi et al. (2016), where workers were asked to write a story about a given activity "as if explaining it to a child". This results in elaborate and explicit texts that are centered around a single scenario. Consequently, the texts are syntactically simple, facilitating machine comprehension models to focus on semantic challenges and inference. We collected 20 texts for each scenario. Each participant was allowed to write only one story per scenario, but work on as many scenarios as they liked. For each of the 10 scenarios from InScript, we randomly selected 20 existing texts from that resource.

2.2.3. Questions

For collecting questions, workers were instructed to "imagine they told a story about a certain scenario to a child and want to test if the child understood everything correctly". This instruction also ensured that questions are linguistically simple, elaborate and explicit. Workers were asked to formulate questions about details of such a situation, i.e. independent of a concrete narrative. This resulted in questions, the answer to which is not literally mentioned in the text.

To cover a broad range of question types, we asked participants to write 3 temporal questions (asking about time points and event order), 3 content questions (asking about persons or details in the scenario) and 3 reasoning questions (asking how or why something happened). They were also asked to formulate 6 free questions, which resulted in a total of 15 questions. Asking each worker for a high number of questions enforced that more creative questions were formulated, which go beyond obvious questions for a scenario.

Since participants were not shown a concrete story, we asked them to use the neutral pronoun "they" to address the protagonist of the story. We permitted participants to work on as many scenarios as desired and we collected questions from 10 participants per scenario.

2.2.4. Answers

Our mode of question collection results in questions that are not associated with specific texts. For each text, we collected answers for 15 questions that were randomly selected from the same scenario. Since questions and texts were collected independently, answering a random question is not always possible for a given text. Therefore, we carried out answer collection in two steps. In the first step, we asked participants to assign a category to each text–question pair. We distinguish two categories of answerable questions: The category *text-based* was assigned to questions that can be answered from the text directly. If the answer could only be inferred by using commonsense knowledge, the category

[1] www.mturk.com

answerable		not answerable	
text-based	**script-based**	**unknown**	**unfitting**
10,160	3,914	9,974	3,172
14,074		13,246	

Table 1: Distribution of question categories

script-based was assigned. Making this distinction is interesting for evaluation purposes, since it enables us to estimate the number of commonsense inference questions. For questions that did not make sense at all given a text, *unfitting* was assigned. If a question made sense for a text, but it was impossible to find an answer, the label *unknown* was used.

In a second step, we told participants to formulate a plausible correct and a plausible incorrect answer candidate to answerable questions (*text-based* or *script-based*). To level out the effort between answerable and non-answerable questions, participants had to write a new question when selecting *unknown* or *unfitting*.

In order to get reliable judgments about whether or not a question can be answered, we collected data from 5 participants for each question and decided on the final category via majority vote (at least 3 out of 5). Consequently, for each question with a majority vote on either *text-based* or *script-based*, there are 3 to 5 correct and incorrect answer candidates, one from each participant who agreed on the category. Questions without a clear majority vote or with ties were not included in the dataset.

2.2.5. Data Post-Processing

We performed four post-processing steps on the collected data.

- We manually filtered out texts that were instructional rather than narrative.
- All texts, questions and answers were spellchecked by running aSpell[2] and manually inspecting all corrections proposed by the spellchecker.
- We found that some participants did not use "they" when referring to the protagonist. We identified "I", "you", "he", "she", "my", "your", "his", "her" and "the person" as most common alternatives and replaced each appearance manually with "they" or "their", if appropriate.
- We manually filtered out invalid questions, e.g. questions that are suggestive ("Should you ask an adult before using a knife?") or that ask for the personal opinion of the reader ("Do you think going to the museum was a good idea?").

2.3. Answer Selection and Validation

We finalized the dataset by selecting one correct and one incorrect answer for each question–text pair. To increase the proportion of non-trivial inference cases, we chose the candidate with the *lowest* lexical overlap with the text from the set of correct answer candidates as *correct* answer. Using this principle also for incorrect answers leads to problems. We found that many incorrect candidates were not plausible answers to a given question. Instead of selecting a candidate based on overlap, we hence decided to rely on majority vote and selected the candidate from the set of incorrect answers that was most often mentioned.

For this step, we normalized each candidate by lowercasing, deleting punctuation and stop words (articles, *and*, *to* and *or*), and transforming all number words into digits, using *text2num*[3]. We merged all answers that were string-identical, contained another answer, or had a Levenshtein distance (Levenshtein, 1966) of 3 or less to another answer. The "most frequent answer" was then selected based on how many other answers it was merged with. Only if there was no majority, we selected the candidate with the highest overlap with the text as a fallback.

Due to annotation mistakes, we found a small number of chosen correct and incorrect answers to be inappropriate, that is, some "correct" answers were actually incorrect and vice versa. Therefore, we manually validated the complete dataset in a final step. We asked annotators to read all texts, questions, and answers, and to mark for each question whether the correct and incorrect answers were appropriate. If an answer was inappropriate or contained any errors, they selected a different answer from the set of collected candidates. For approximately 11.5% of the questions, at least one answer was replaced. 135 questions (approx. 1%) were excluded from the dataset because no appropriate correct or incorrect answer could be found.

2.4. Data Statistics

For all experiments, we admitted only experienced MTurk workers who are based in the US. One HIT[4] consisted of writing one text for the text collection, formulating 15 questions for the question collection, or finding 15 pairs of answers for the answer collection. We paid $0.50 per HIT for the text and question collection, and $0.60 per HIT for the answer collection.

More than 2,100 texts were paired with 15 questions each, resulting in a total number of approx. 32,000 annotated questions. For 13% of the questions, the workers did not agree on one of the 4 categories with a 3 out of 5 majority, so we did not include these questions in our dataset.

The distribution of category labels on the remaining 87% is shown in Table 1. 14,074 (52%) questions could be answered. Out of the answerable questions, 10,160 could be answered from the text directly (*text-based*) and 3,914 questions required the use of commonsense knowledge (*script-based*). After removing 135 questions during the validation, the final dataset comprises 13,939 questions, 3,827 of which require commonsense knowledge (i.e. 27.4%). This ratio was manually verified based on a random sample of questions.

We split the dataset into training (9,731 questions on 1,470 texts), development (1,411 questions on 219 texts), and test set (2,797 questions on 430 texts). Each text appears only

[2] `http://aspell.net/`

[3] `https://github.com/ghewgill/text2num`

[4] A *Human Intelligence Task* (HIT) is one single experimental item in MTurk.

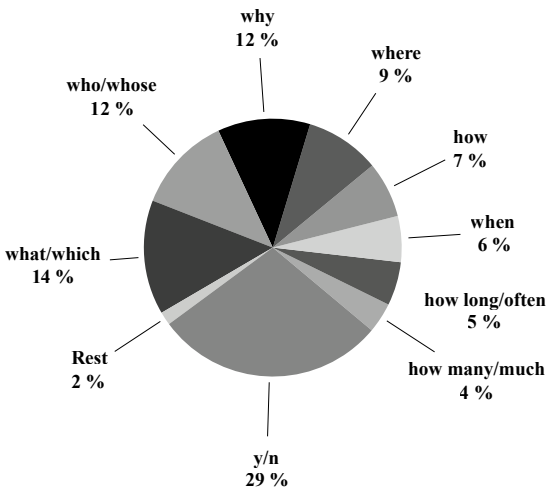

Figure 2: Distribution of question types in the data.

in one of the three sets. The complete set of texts for 5 scenarios was held out for the test set.

The average text, question, and answer length is 196.0 words, 7.8 words, and 3.6 words, respectively. On average, there are 6.7 questions per text.

Figure 2 shows the distribution of question types in the dataset, which we identified using simple heuristics based on the first words of a question: Yes/no questions were identified as questions starting with an auxiliary or modal verb, all other question types were determined based on the question word.

We found that 29% of all questions are yes/no questions. Questions about details of a situation (such as *what/ which* and *who*) form the second most frequent question category. Temporal questions (*when* and *how long/often*) form approx. 11% of all questions. We leave a more detailed analysis of question types for future work.

3. Data Analysis

As can be seen from the data statistics, our mode of collection leads to a substantial proportion of questions that require inference using commonsense knowledge. Still, the dataset contains a large number of questions in which the answer is explicitly contained or implied by the text: Figure 3 shows passages from an example text of the dataset together with two such questions. For question Q1, the answer is given literally in the text. Answering question Q2 is not as simple; it can be solved, however, via standard semantic relatedness information (chicken and hotdogs are meat; water, soda and juice are drinks).

The following cases require commonsense inference to be decided. In all these cases, the answers are not overtly contained nor easily derivable from the respective texts. We do not show the full texts, but only the scenario names for each question.

(2) BORROWING A BOOK FROM THE LIBRARY
Did they have to pay anything to borrow the book?
a. yes
b. no

(3) CHANGING A BABY DIAPER
Did they throw away the old diaper?
a. Yes, they put it into the bin.
b. No, they kept it for a while.

(4) CLEANING THE TABLE
When did they clean the table?
a. After a meal
b. Before they ate

(5) PREPARING A PICNIC
Who is packing the picnic?
a. the children
b. the parents

(6) TAKING A SHOWER
How long did the shower take?
a. a few hours
b. a few minutes

T It was time to prepare for the picnic that we had plans for the last couple weeks. ...I needed to set up the cooler, which included bottles of water, soda and juice to keep everyone hydrated. Then I needed to ensure that we had all the food we intended to bring or cook. So at home, I prepared baked beans, green beans and macaroni and cheese. ...But in a cooler, I packed chicken, hotdogs, hamburgers and rots that were to be cooked on the grill once we were at the picnic location.

Q1 What did they bring to drink?
a. Water, soda and juice.
b. Water, wine coolers and sports drinks.

Q2 What type of food did they pack?
a. Meat, drinks and side dishes.
b. Pasta salad only.

Figure 3: An example text with 2 questions from MCScript

Example 2 refers to a library setting. Script knowledge helps in assessing that usually, *paying* is not an event when borrowing a book, which answers the question. Similarly, event information helps in answering the questions in Examples 3 and 4. In Example 5, knowledge about the typical role of parents in the preparation of a picnic will enable a plausibility decision. Similarly, in Example 6, it is commonsense knowledge that showers usually take a few minutes rather than hours.

(7) MAKING BREAKFAST
What time of the day is breakfast eaten?
a. at night
b. in the morning

There are also cases in which the answer can be inferred from the text, but where commonsense knowledge is still beneficial: The text for example 7 does not contain the information that breakfast is eaten in the morning, but it could still be inferred from many pointers in the text (e.g. phrases such as *I woke up*), or from commonsense knowledge.
These few examples illustrate that our dataset covers questions with a wide spectrum of difficulty, from rather simple questions that can be answered from the text to challenging inference problems.

4. Experiments

In this section, we assess the performance of baseline models on MCScript, using accuracy as the evaluation measure. We employ models of differing complexity: two unsupervised models using only word information and distributional information, respectively, and two supervised neural models. We assess performance on two dimensions: One, we show how well the models perform on text-based questions as compared to questions that require common sense for finding the correct answer. Two, we evaluate each model for each different question type.

4.1. Models

Word Matching Baseline
We first use a simple word matching baseline, by selecting the answer that has the highest literal overlap with the text. In case of a tie, we randomly select one of the answers.

Sliding Window
The second baseline is a sliding window approach that looks at windows of w tokens on the text. Each text and each answer are represented as a sequence of word embeddings. The embeddings for each window of size w and each answer are then averaged to derive window and answer representations, respectively. The answer with the lowest cosine distance to one of the windows of the text is then selected as correct.

Bilinear Model
We employ a simple neural model as a third baseline. In this model, each text, question, and answer is represented by a vector. For a given sequence of words $w_1 \dots w_n$, we compute this representation by averaging over the components of the word embeddings $\mathbf{w}_i$ that correspond to a word w_i, and then apply a linear transformation using a weight matrix. This procedure is applied to each answer a to derive an answer representation $\mathbf{a}$. The representation of a text $\mathbf{t}$ and of a question $\mathbf{q}$ are computed in the same way. We use different weight matrices for $\mathbf{a}$, $\mathbf{t}$ and $\mathbf{q}$, respectively. A combined representation $\mathbf{p}$ for the text–question pair is then constructed using a bilinear transformation matrix $\mathbf{W}$:

$$\mathbf{p} = \mathbf{t}^\top \mathbf{W} \mathbf{q} \tag{1}$$

We compute a score for each answer by using the dot product and pass the scores for both answers through a softmax layer for prediction. The probability p for an answer a to be correct is thus defined as:

$$p(a|t,q) = softmax(\mathbf{p}^\top \mathbf{a}) \tag{2}$$

Attentive Reader
The attentive reader is a well-established machine comprehension model that reaches good performance e.g. on the *CNN/Daily Mail* corpus (Hermann et al., 2015; Chen et al., 2016). We use the model formulation by Chen et al. (2016) and Lai et al. (2017), who employ bilinear weight functions to compute both attention and answer-text fit. Bidirectional GRUs are used to encode questions, texts and answers into hidden representations. For a question q and an answer a, the last state of the GRUs, $\mathbf{q}$ and $\mathbf{a}$, are used as representations, while the text is encoded as a sequence of hidden states $\mathbf{t}_1 ... \mathbf{t}_n$. We then compute an attention score s_j for each hidden state $\mathbf{t}_j$ using the question representation $\mathbf{q}$, a weight matrix $\mathbf{W}_a$, and an attention bias b. Last, a text representation $\mathbf{t}$ is computed as a weighted average of the hidden representations:

$$\begin{aligned} s_j &= softmax_j(\mathbf{t}_j^\top \mathbf{W}_a \mathbf{q} + b) \\ \mathbf{t} &= \sum_j s_j \mathbf{t}_j \end{aligned} \tag{3}$$

The probability p of answer a being correct is then predicted using another bilinear weight matrix $\mathbf{W}_s$, followed by an application of the softmax function over both answer options for the question:

$$p(a|t,q) = softmax(\mathbf{t}^\top \mathbf{W}_s \mathbf{a}) \tag{4}$$

4.2. Implementation Details

Texts, questions and answers were tokenized using NLTK[5] and lowercased. We used 100-dimensional GloVe vectors[6] (Pennington et al., 2014) to embed each token. For the neural models, the embeddings are used to initialize the token representations, and are refined during training. For the sliding similarity window approach, we set $w = 10$.
The vocabulary of the neural models was extracted from training and development data. For optimizing the bilinear model and the attentive reader, we used vanilla stochastic gradient descent with gradient clipping, if the norm of gradients exceeds 10. The size of the hidden layers was tuned to 64, with a learning rate of 0.2, for both models. We apply a dropout of 0.5 to the word embeddings. Batch size was set to 25 and all models were trained for 150 epochs. During training, we measured performance on the development set, and we selected the model from the best performing epoch for testing.

4.3. Results and Evaluation

Human Upper Bound
As an upper bound for model performance, we assess how well humans can solve our task. Two trained annotators labeled the correct answer on all instances of the test set. They agreed with the gold standard in 98.2 % of cases. This result shows that humans have no difficulty in finding the correct answer, irrespective of the question type.

[5] http://www.nltk.org/
[6] https://nlp.stanford.edu/projects/glove/

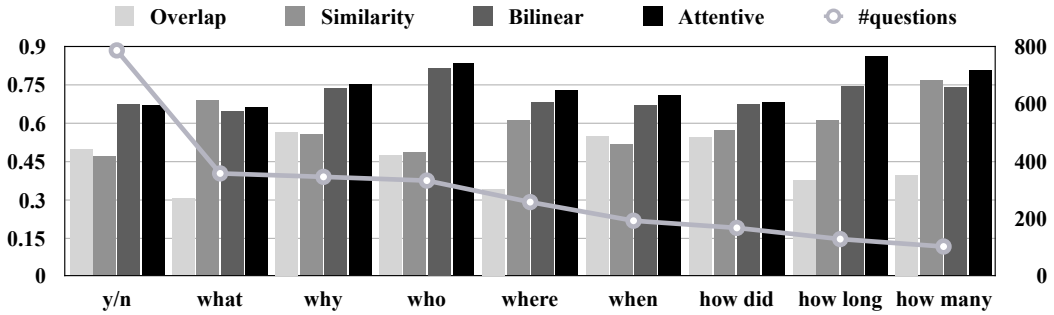

Figure 4: Accuracy values of the baseline models on question types appearing > 25 times.

Model	*Text*	*CS*	*Total*
Chance	50.0	50.0	50.0
Word Overlap	41.8	59.0	54.4
Sliding Window	55.7	53.1	55.0
Bilinear Model	69.8	71.4	70.2
Attentive Reader	**70.9**	**75.2**	**72.0**
Human Performance			98.2

Table 2: Accuracy of the baseline systems on text-based (*Text*), on commonsense-based questions (*CS*), and on the whole test set (*Total*). All numbers are percentages.

Performance of the Baseline Models

Table 2 shows the performance of the baseline models as compared to the human upper bound and a random baseline. As can be seen, neural models have a clear advantage over the pure word overlap baseline, which performs worst, with an accuracy of 54.4%.

The low accuracy is mostly due to the nature of correct answers in our data: Each correct answer has a low overlap with the text by design. Since the overlap model selects the answer with a high overlap to the text, it does not perform well. In particular, this also explains the very bad result on text-based questions. The sliding similarity window model does not outperform the simple word overlap model by a large margin: Distributional information alone is insufficient to handle complex questions in the dataset.

Both neural models outperform the unsupervised baselines by a large margin. When comparing the two models, the attentive reader is able to beat the bilinear model by only 1.8%. A possible explanation for this is that the attentive reader only attends to the text. Since many questions cannot be directly answered from the text, the attentive reader is not able to perform significantly better than a simpler neural model.

What is surprising is that the attentive reader works better on commonsense-based questions than on text questions. This can be explained by the fact that many commonsense questions do have prototypical answers within a scenario, irrespective of the text. The attentive reader is apparently able to just memorize these prototypical answers, thus achieving higher accuracy.

Inspecting attention values of the attentive reader, we found that in most cases, the model is unable to properly attend to the relevant parts of the text, even when the answer is literally given in the text. A possible explanation is that the model is confused by the large amount of questions that cannot be answered from the text directly, which might confound the computation of attention values.

Also, the attentive reader was originally constructed for reconstructing literal text spans as answers. Our mode of answer collection, however, results in many correct answers that cannot be found verbatim in the text. This presents difficulties for the attention mechanism.

The fact that an attention model outperforms a simple bilinear baseline only marginally shows that MCScript poses a new challenge to machine comprehension systems. Models concentrating solely on the text are insufficient to perform well on the data.

Performance on Question Types

Figure 4 gives accuracy values of all baseline systems on the most frequent question types (appearing $>$25 times in the test data), as determined based on the question words (see Section 2.4.). The numbers depicted on the left-hand side of the y-axis represent model accuracy. The right-hand side of the y-axis indicates the number of times a question type appears in the test data.

The neural models unsurprisingly outperform the other models in most cases, and the difference for *who* questions is largest. A large number of these questions ask for the narrator of the story, who is usually not mentioned literally in the text, since most stories are written in the first person.

It is also apparent that all models perform rather badly on *yes/no* questions. Each model basically compares the answer to some representation of the text. For yes/no questions, this makes sense for less than half of all cases. For the majority of yes/no questions, however, answers consist only of *yes* or *no*, without further content words.

5. Related Work

In recent years, a number of reading comprehension datasets have been proposed, including *MCTest* (Richardson et al., 2013), *BAbI* (Weston et al., 2015), the *Children's Book Test* (*CBT*, Hill et al. (2015)), *CNN/Daily Mail* (Hermann et al., 2015), the *Stanford Question Answering Dataset* (*SQuAD*, Rajpurkar et al. (2016)), and *RACE* (Lai et al., 2017). These datasets differ with respect to text type (Wikipedia texts, examination texts, etc.), mode of answer selection (span-based, multiple choice, etc.) and test systems regarding different aspects of language understand-

ing, but they do not explicitly address commonsense knowledge.

Two notable exceptions are the *NewsQA* and *TriviaQA* datasets. *NewsQA* (Trischler et al., 2017) is a dataset of newswire texts from CNN with questions and answers written by crowdsourcing workers. *NewsQA* closely resembles our own data collection with respect to the method of data acquisition. As for our data collection, full texts were not shown to workers as a basis for question formulation, but only the text's title and a short summary, to avoid literal repetitions and support the generation of non-trivial questions requiring background knowledge. The NewsQA text collection differs from ours in domain and genre (newswire texts vs. narrative stories about everyday events). Knowledge required to answer the questions is mostly factual knowledge and script knowledge is only marginally relevant. Also, the task is not exactly question answering, but identification of document passages containing the answer. *TriviaQA* (Joshi, Mandar and Choi, Eunsol and Weld, Daniel S. and Zettlemoyer, Luke, 2017) is a corpus that contains automatically collected question-answer pairs from 14 trivia and quiz-league websites, together with web-crawled evidence documents from *Wikipedia* and *Bing*. While a majority of questions require world knowledge for finding the correct answer, it is mostly factual knowledge.

6. Summary

We present a new dataset for the task of machine comprehension focussing on commonsense knowledge. Questions were collected based on script scenarios, rather than individual texts, which resulted in question–answer pairs that explicitly involve commonsense knowledge. In contrast to previous evaluation tasks, this setup allows us for the first time to assess the contribution of script knowledge for computational models of language understanding in a real-world evaluation scenario.

We expect our dataset to become a standard benchmark for testing models of commonsense and script knowledge. Human performance shows that the dataset is highly reliable. The results of several baselines, in contrast, illustrate that our task provides challenging test cases for the broader natural language processing community. MCScript forms the basis of a shared task organized at SemEval 2018. The dataset is available at `http://www.sfb1102.uni-saarland.de/?page_id=2582`.

Acknowledgements

We thank the reviewers for their helpful comments. We also thank Florian Pusse for the help with the MTurk experiments and our student assistants Christine Schäfer, Damyana Gateva, Leonie Harter, Sarah Mameche, Stefan Grünewald and Tatiana Anikina for help with the annotations. This research was funded by the German Research Foundation (DFG) as part of SFB 1102 'Information Density and Linguistic Encoding' and EXC 284 'Multimodal Computing and Interaction'.

7. Bibliographical References

Chambers, N. and Jurafsky, D. (2008). Unsupervised Learning of Narrative Event Chains. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 789–797, Columbus, Ohio, June. Association for Computational Linguistics.

Chambers, N. and Jurafsky, D. (2009). Unsupervised Learning of Narrative Schemas and their Participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Singapore, August. Association for Computational Linguistics.

Chen, D., Bolton, J., and Manning, C. D. (2016). A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2358–2367.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Hill, F., Bordes, A., Chopra, S., and Weston, J. (2015). The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv preprint arXiv:1511.02301*.

Joshi, Mandar and Choi, Eunsol and Weld, Daniel S. and Zettlemoyer, Luke. (2017). 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. Association for Computational Linguistics.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. (2017). RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710.

Modi, A. and Titov, I. (2014a). Inducing Neural Models of Script Knowledge. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, Baltimore, MD, USA.

Modi, A. and Titov, I. (2014b). Learning Semantic Script Knowledge with Event Embeddings. In *Proceedings of the 2nd International Conference on Learning Representations (Workshop track)*, Banff, Canada.

Modi, A., Anikina, T., Ostermann, S., and Pinkal, M. (2016). InScript: Narrative texts annotated with script information. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, may. European Language Resources Association (ELRA).

Modi, A. (2016). Event Embeddings for Semantic Script Modeling. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 75–83, Berlin, Germany, August. Association for

Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Pichotta, K. and Mooney, R. J. (2014). Statistical Script Learning with Multi-Argument Events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 220–229, Gothenburg, Sweden, April.

Pichotta, K. and Mooney, R. J. (2016). Using Sentence-Level LSTM Language Models for Script Inference. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-16)*, Berlin, Germany.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Regneri, M., Koller, A., and Pinkal, M. (2010). Learning Script Knowledge with Web Experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 979–988, Uppsala, Sweden, July. Association for Computational Linguistics.

Richardson, M., Burges, C. J., and Renshaw, E. (2013). MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA, October. Association for Computational Linguistics.

Schank, R. C. and Abelson, R. P. (1975). Scripts, Plans, and Knowledge. In *Proceedings of the 4th international joint conference on Artificial intelligence-Volume 1*, pages 151–157. Morgan Kaufmann Publishers Inc.

Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., and Zhu, W. L. (2002). Open Mind Common Sense: Knowledge Acquisition from the General Public. In *On the move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237. Springer.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., and Suleman, K. (2017). NewsQA: A Machine Comprehension Dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Wanzare, L. D. A., Zarcone, A., Thater, S., and Pinkal, M. (2016). DeScript: A Crowdsourced Database for the Acquisition of High-quality Script Knowledge. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, may. European Language Resources Association (ELRA).

Wanzare, L., Zarcone, A., Thater, S., and Pinkal, M. (2017). Inducing Script Structure from Crowdsourced Event Descriptions via Semi-Supervised Clustering. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 1–11, Valencia, Spain, April. Association for Computational Linguistics.

Weston, J., Bordes, A., Chopra, S., Rush, A. M., van Merriënboer, B., Joulin, A., and Mikolov, T. (2015). Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.