Comparing Attention-based Convolutional and Recurrent Neural Networks: Success and Limitations in Machine Reading Comprehension

Matthias Blohm, Glorianna Jagfeld, Ekta Sood, Xiang Yu, Ngoc Thang Vu
Institute for Natural Language Processing (IMS)
Universität Stuttgart, Germany
{blohmms,jagfelga,soodea,xiangyu,thangvu}
@ims.uni-stuttgart.de

Abstract

We propose a machine reading comprehension model based on the compare-aggregate framework with two-staged attention that achieves state-of-the-art results on the MovieQA question answering dataset. To investigate the limitations of our model as well as the behavioral difference between convolutional and recurrent neural networks, we generate adversarial examples to confuse the model and compare to human performance. Furthermore, we assess the generalizability of our model by analyzing its differences to human inference, drawing upon insights from cognitive science.

1 Introduction

Current state-of-the-art deep learning (DL) models outperform other techniques in many tasks including computer vision (Krizhevsky et al., 2012), speech recognition (Hinton et al., 2012) and more recently natural language processing (NLP) (Collobert et al., 2011). Neural-based NLP systems often use word embeddings (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2013) which are then fed into a convolutional neural network (CNN) (LeCun et al., 1990; Waibel et al., 1990) or a recurrent neural network (RNN) (Elman, 1990; Hochreiter and Schmidhuber, 1997) for further classification. These approaches proved to be successful for many NLP tasks (Mikolov et al., 2010; Kim, 2014; Hu et al., 2014; Bahdanau et al., 2014). Along with the success of DL in a wide range of applications, adversarial examples (Goodfellow et al., 2014) - that aim to confuse the system - have gained popularity in a wide range of research communities such as computer vision and NLP, since they can reveal the limitations in the generalizability of the models. As opposed to adversarial examples in computer vision, which are computed

on continuous data and can thus easily be imperceptible if desired, adversarial attacks in NLP entail the necessity to perform discrete and perceptible changes to the data. Thus, attack methods for computer vision such as the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) cannot be directly applied to NLP.

Machine comprehension has recently received increased interest in the NLP community (Yang et al., 2015; Tapaswi et al., 2016; Rajpurkar et al., 2016; Chen et al., 2016). Neural network models perform reasonably well on many data sets with different question answering setups, e.g. multiple choice or answer generation (Wang and Jiang, 2016; Liu et al., 2017; Yu et al., 2018).

Among others, Wang and Jiang (2016) proposed the compare-aggregate framework, which uses an attention mechanism (Luong et al., 2015) to compare the question and candidate answers, and a CNN to aggregate information. However, there is still an ongoing debate whether CNNs or RNNs are more suitable to NLP (Yin et al., 2017), and the behavioral differences between them are still under research. Many papers reported remarkable gains when combining these two models in ensembles (Deng and Platt, 2014; Zhou et al., 2015; Vu et al., 2016), since they process information in different ways and thus are complimentary to each other.

Despite the seemingly high accuracies of many models on machine comprehension tasks, Jia and Liang (2017) argued that many questions in such datasets are easily solvable by superficial cues. They showed with adversarial examples that most models can be easily tricked by modifications on the data which do not confuse humans. Similarly, Sanchez et al. (2018) performed controlled experiments on the robustness of several Natural Language Inference models by altering hypernym, hyponym, and antonym relations in the data. Both

studies revealed a major weakness of the models: They largely rely on pattern matching instead of human decision-making processes as required in the tasks, including heuristics (Gigerenzer and Gaissmaier, 2011) and elimination by aspects (Tversky, 1972).

In this paper, we implement two machine comprehension models based on the compare-aggregate framework with a hierarchical attention structure using CNNs and RNNs. First we show that we achieve state-of-the-art results on the MovieQA multiple choice question answering dataset (Tapaswi et al., 2016) outperforming other systems by a large margin.[1] Second, we investigate the different behavior of the two systems applying adversarial attacks in a systematic way. To our best knowledge, this is the first work exploring the difference between CNNs and RNNs by such an approach. Third, we present a detailed comparison between human and machine reading comprehension, giving insights when and why our systems fail. Therefore, these insights are important for future research towards enhancing machine comprehension systems loosely inspired by human processing. All code necessary to reproduce our experimental results is made available.[2]

2   Hierarchical Attention-based Compare-Aggregate Model

The basis for our model is the compare-aggregate model with attention (Wang and Jiang, 2016) that has been shown effective for reading comprehension. We extend the model in two aspects that lead to significant improvements.

Given a preprocessed matrix-representation of the question Q, a text (movie plot) P, and k answer candidates $A_1 \ldots A_k$, the main idea of Wang and Jiang (2016)'s compare-aggregate model is to compare P to Q and each A_j and then aggregate this information into a vector to derive a confidence score c_j for each answer candidate.

Wang and Jiang (2016) concatenate all plot sentences and do not leverage the inherent structuring of the text into sentences. Inspired by the recent success of hierarchical models in NLP (Sordoni et al., 2015; Yang et al., 2016; Liu et al., 2017) we extend the model to perform comparison and aggregation on the word and sentence

level separately (see Figure 1). Specifically, we first apply the compare-aggregate model to each plot sentence P_i individually to obtain question and answer-weighted representations $T_{q_i}^w, T_{a_{ij}}^w$ for each sentence. We then run the aggregation operation on each sentence representation individually to obtain sentence vector representations $r_{p_{ij}}$. The sentence representations are concatenated to obtain a plot representation r_{p_j}, which enters the sentence level of comparison and aggregation that mirrors the word level architecture.

As a second modification of the base model, we implement an RNN-based aggregation function to replace the CNN-based aggregation originally proposed by Wang and Jiang (2016). In the following we detail the building blocks of our hierarchical attention-based compare-aggregate model as depicted in Figure 1.

Preprocessing We represent the words in the question q, the plot sentences p_i and the answer candidates a_j by pretrained embeddings to obtain matrices $\overline{Q}, \overline{P}, \overline{A_j}$. We project them to lower dimensional Q, P, A_j via the following operation:

$$X = \sigma \left(W^i \overline{X} + b^i \right) \odot \tanh \left(W^u \overline{X} + b^u \right) \quad (1)$$

Attention The attention operation weights the plot regarding the question or a candidate answer.

$$G = \text{softmax} \left(X^T P \right) \quad (2)$$
$$H = XG, \quad (3)$$

where X on the word level represents Q or an answer candidate A_j and on the sentence level r_q or r_{a_j}.[3]

Comparison The comparison operation performs an element-wise comparison of each h_l in H with its counterparts q_l/a_{j_l} on the word level and r_q/r_{a_j} on the sentence level, respectively. Wang and Jiang (2016) compared many comparison functions. Here we use only the SUBMULT function since it performed best for MovieQA:

$$t_l = \text{ReLU}(W \left[\begin{array}{c} (x_l - h_l) \odot (x_l - h_l) \\ x_l \odot h_l \end{array} \right] + b)$$

where $\odot$ denotes element-wise multiplication and x_l corresponds to entries of Q/A_j or r_q/r_{a_j}.

---

[3]Different from Wang and Jiang (2016) we use dot-product attention instead of general attention (Luong et al., 2015) because we found no benefit of the additional parameters of general attention in preliminary experiments.
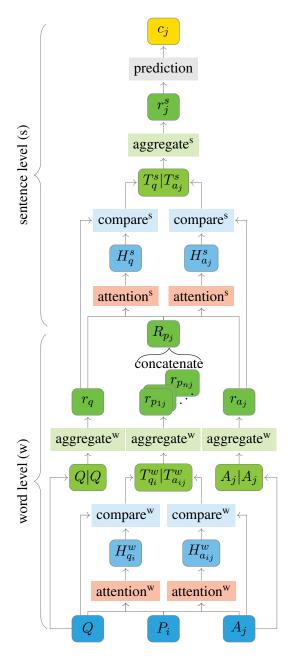
Figure 1: Hierarchical compare-aggregate model to compute the confidence score c_j of a preprocessed answer candidate A_j given question Q and plot $P = P_1 \ldots P_n$.

Aggregation The goal of the aggregation operation is to condense the information of a variable-length sequence into a single vector. Wang and Jiang (2016) implemented the aggregation operation as a single-layer CNN following Kim (2014). Specifically, they used a 1D convolution with filter sizes $\{1,3,5\}$, to capture unigrams, trigrams and 5-grams.

$$\text{aggregate}_{\text{CNN}} = \text{CNN}([z_1 \ldots z_m]) \quad (4)$$

where $[z_1 \ldots z_m]$ on the word level corresponds to the sequence of row vectors of $Q, T^w = T^w_{q_i} | T^w_{a_{ij}}, A_j$, and on the sentence level to that of $T^s = T^s_q | T^s_{a_j}$.

While CNNs are effective in modeling location-independent n-gram patterns, they cannot capture longer-range dependencies. Yet, we argue that it is important to also consider the context of the matched phrases. This motivates our proposed sequential aggregation function based on a single-layer unidirectional RNN with Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997).

$$\text{aggregate}_{\text{RNN-LSTM}} = \text{RNN}([z_1 \ldots z_m]) \quad (5)$$

By performing 1-max pooling over the outputs of $\text{aggregate}_{\text{CNN}}$ or $\text{aggregate}_{\text{RNN-LSTM}}$[4] we obtain a single vector r (representing $r_q, r_{p_{ij}}, r_{a_j}$ on the word level, or r^s_j on the sentence level):

$$r = \text{max\_pool}(\text{aggregate}([z_1 \ldots z_m])) \quad (6)$$

We share the weights between the comparison and aggregation operations within the word and sentence level but not across levels.

Prediction We map each aggregated answer-specific plot representation r^s_j to a confidence score c_j by two dense layers with shared weights for all answer candidates and of which the first uses $\tanh$ activation and the second one no activation function. The confidence scores are normalized to form a probability distribution $p_1 \ldots p_k$ by a softmax operation.

3 Experimental Set-Up

The hyperparameters for our models are provided in §A.1 in the appendix.

3.1 Data

We evaluate our models on the MovieQA dataset (Tapaswi et al., 2016) that contains 14,944 multiple-choice questions on 408 movies collected by human annotators. The questions vary from simple "*who*" or "*when*" to more complex "*why*" or "*how*" question types. Each question is provided along with five candidate answers of which only one is correct.

While the dataset contains multiple sources of information about the movie contents such as

---

[4]Using only the last RNN output for $\text{aggregate}_{\text{RNN-LSTM}}$ did not provide convincing results.

videos, subtitles, and movie scripts, here we focus on answering the questions only from *plot synopses*. Plot synopses are summaries of the movies collected from Wikipedia that mostly describe the actions happening in the story. They were used as references for the question collection and so far yield the best results on the dataset according to the MovieQA leaderboard. Figure 2 shows a sample question together with its candidate answers and an excerpt of the corresponding movie plot which contains the necessary information to answer the question. The dataset is split into 9,848 training, 1,958 development and 3,138 test questions, respectively. Note that the test set accuracies can only be evaluated by submitting the predictions to the server.

Plot: ... Aragorn is crowned King of Gondor and taking Arwen as his queen before all present at his coronation bowing before Frodo and the other Hobbits. The Hobbits return to **the Shire** where Sam marries Rosie Cotton. ...

Question: Where does Sam marry Rosie?

Candidate Answers: 0) Grey Havens 1) Gondor 2) **The Shire** 3) Erebor 4) Mordor

---

Figure 2: *MovieQA* example question (Wang and Jiang, 2016).

4 Results

We train 11 models with different random initializations for both the CNN and RNN-LSTM aggregation function and form majority-vote ensembles of the nine models with the highest validation accuracy. Table 1 shows the accuracies of ensembles of our proposed model variations in comparison to the published results on the MovieQA validation and test set. To the best of our knowledge, the results of Wang and Jiang (2016) and Dzendzik et al. (2017) were achieved by single models, while the results of Liu et al. (2017) corresponds to an ensemble of multiple models.

All our hierarchical single and ensemble models outperform the previous state of the art on both the validation and test set. With a test accuracy of 85.12, the RNN-LSTM ensemble achieves a new state of the art that is more than five percentage points above the previous best result.

The hierarchical structure is crucial for the model's success. Adding it to the CNN that oper-

Systems	Val.	Test
Wang and Jiang (2016)	72.10	72.90
Liu et al. (2017)	79.00	79.99
Dzendzik et al. (2017)	-	80.02
Proposed models		
CNN word level only	76.51	-
CNN	79.62	-
CNN ensemble	82.58	82.73
RNN-LSTM	83.14	-
RNN-LSTM ensemble	84.37	**85.12**
CNN RNN-LSTM ensemble	**84.78**	84.70

Table 1: MovieQA accuracies for previously published results and our proposed single models (best out of 11) and ensembles (nine best out of 11).

ates only at word level[5] causes a pronounced improvement on the validation set.

Furthermore, the RNN-LSTM aggregation function is superior to aggregation via CNNs, improving the validation accuracy by 1.5 percentage points. While this improvement is statistically significant,[6] combining both aggregation functions by ensembling the nine best CNN and RNN-LSTM models each, yields a small but statistically insignificant improvement of 0.41 percentage points over the RNN-LSTM ensemble on the validation set. This might explain why the RNN-LSTM ensemble even outperforms the CNN RNN-LSTM ensemble on the test set by a small margin. The difference in test set performance between these two ensembles is likely not significant. We cannot test this as the test set is not released and only accuracy values can be obtained for model evaluation on the test set.

4.1 Impact of Sentence Attention

The sentence attention allows us to get more insight into the models' inner state. For example, it allows us to check whether the model actually focuses on relevant sentences in order to answer the questions. The MovieQA dataset provides human annotations of the minimal set of plot sentences required to answer a question. In average, 1.15/1.11 sentences in the training/validation set are marked as containing the clue to the answer. We leverage

---

[5]The CNN word level only model essentially corresponds to our reimplementation of Wang and Jiang (2016). The performance gain on the validation set might be due to using consistent random initializations for unknown words.

[6]McNemar test (McNemar, 1947), $p < 0.05$.

Systems	CNN	RNN-LSTM
All questions	71.45	71.31
- Correctly solved	80.86	79.35
- Incorrectly solved	35.73	34.49

Table 2: Percentage of questions in which the plot sentences containing the clues for the answer are ranked highest according to the model's sentence attention distribution (relative to its selected answer) on the validation set (averaged results of nine models).

Systems	Average	Ensemble
CNN	78.74	81.72
RNN-LSTM	81.53	83.76
CNN RNN-LSTM	81.14	84.27

Table 3: Adversarial accuracies on the validation set under the word-level black-box attack based on manual lexical substitutions in questions.

this information and compute the ranks of these relevant plot sentences according to the models' sentence attention distribution. We extract the plot sentence relevance scores after the sentence-level comparison operation as average of T_q^s and $T_{a_j}^s$, where a_j corresponds to the selected answer of the model. As Table 2 reveals, both model variants pay most attention to the relevant plot sentences for 70% of the cases. Identifying the relevant sentences is an important success factor: Relevant sentences are ranked highest only in 35% of the incorrectly solved questions.

5 Limitations

To help us identifying the models' weaknesses, we design a series of systematic adversarial attacks. These attacks are defined in different categories depending on the linguistic level (word vs. sentence level) and the knowledge of the adversaries (black-box vs. white-box). According to the taxonomy proposed by Yuan et al. (2017), black-box and white-box attacks differ in the access of the adversary to the trained neural network model. In black-box settings, the adversary acts as a standard user that has only access to the output of the model in form of labels or confidence scores. In contrast, the adversary in white-box settings has access to all the details of the models such as training data, network architectures and hyperparameters. In this work, the white-box adversary has access to the attention weights of the model at the word and sentence level. We apply all our attacks to the nine selected models (see §4) for each aggregation type.

5.1 Word-level Black-box Attack

Adversarial examples for image recognition are typically created by adding some imperceptible noise (Szegedy et al., 2014; Goodfellow et al., 2015), yet this is difficult to do for natural lan-

guage because of its discrete nature. The closest analogue would be paraphrasing but high-quality paraphrases are difficult to obtain automatically: Recent attempts with a sophisticated paraphrase-generation system based on a large paraphrase database yielded about 20% contradicting adversarial examples (Iyyer et al., 2018).

Thus, we designed an adversarial black-box attack on the questions based on manual lexical substitution. We inspected the 106 most frequent words of the validation set questions and manually defined lexical substitutions of single words and multiword expressions of up to two tokens wherever applicable. We made sure that the lexical substitutions were meaning preserving and resulted in grammatical sentences in all contexts.[7] Our final set of 51 substitution rules resulted in a modification of 25% of the validation set questions.

As can be seen from Table 3, the models are quite robust against meaning-preserving lexical substitutions: The accuracy drops by less than one percentage point for all ensembles. Although the differences are small, the RNN-LSTM and CNN RNN-LSTM ensembles are even less affected by lexical substitutions than the CNN ensemble. By only modifying the questions, we have likely reduced their lexical overlap with the answer candidates and the plots. The robustness of the models against this attack can probably be attributed to the pretrained GloVe embeddings, which allow it to generalize for semantically equivalent lexical choices. Stronger attacks involving substitutions with more infrequent words that do not appear in the pretrained embeddings could show the limitation of the models in this respect. We leave the automatic generation of further-reaching adversarial examples based on paraphrases to future work.

---

[7]We only substituted with words contained in the pretrained GloVe embeddings used by the models to avoid introducing unknown words. Even though we did not restrict the substitutes to words from the training set vocabulary, it turned out that all selected words and multiword expressions were indeed contained in the training set vocabulary already, except for the synonym *buddy* for *friend*.
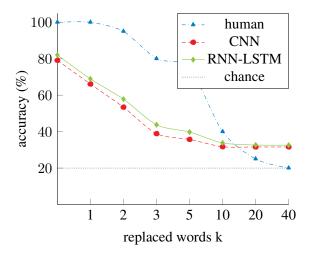
Figure 3: Adversarial accuracies on the validation set under the word-level white-box attack based on word exchange. k is the number of words that are modified in the plot sentence with most attention (average accuracies over nine models). Human evaluation is based on 20 randomly sampled questions with plots attacked for a single CNN model (single annotator, one of the authors of this paper).

5.2 Word-level White-box Attack

We performed a word-level white-box adversarial attack in which we used the models' internal attention distributions to explicitly target the plot words they base their decision on. More precisely, in this experiment we leveraged the models' sentence-level attention distribution to find the plot sentence it gave most weight to conditioned on the correct answer. In this sentence, the k words that received most attention were then exchanged by randomly chosen words from the MovieQA vocabulary.

As Figure 3 reveals, already modifying the single most important word in the most important sentence has a large effect on the average performance of both the CNN and RNN-LSTM models. For increasing k, the RNN-LSTM versions appeared to be a bit more robust against the attack, but for $k \geq 10$ the difference shrinks and the accuracy of both models drops to only about 30%. This experiment shows that manipulating the most relevant plot information by removing important words makes the model fail quickly, since it is no longer able to draw correct conclusions for the questions without the necessary plot context. Although the human annotator proved more robust against this attack for a small number of replaced words, increasing k beyond five showed the same drastic decline in performance.

Systems	Orig.	*AddC*	*AddQ*	*AddQA*
Without optimization				
CNN	76.87	76.67	76.66	76.33
RNN-LSTM	81.11	81.11	81.05	81.05
After two optimization epochs				
CNN	N/A	73.38	57.39	13.61
RNN-LSTM	N/A	79.94	68.05	23.22

Table 4: Adversarial accuracies on 200 random validation questions under the sentence-level black-box attacks (averaged results of nine models).

5.3 Sentence-level Black-box Attacks

In order to find out to which extent our models are susceptible to distracting information added to the plot, we adapt the *AddAny* attack by Jia and Liang (2017) originally designed for the SQuAD reading comprehension dataset. This adversarial attack consists of adding a distractor sentence s at the end of the plot, regardless of grammaticality. The word sequence $s = w_1 w_2 \ldots w_{10}$ is initialized by ten common English words. Then each word is greedily changed from a pool of 20 random common words (*AddC*) to minimize the model's confidence score for the correct answer. We refer the reader to Jia and Liang (2017) for the full details of this attack. Likewise we generate adversarial sentences using a pool of ten random common words for each w_i in conjunction with all question words (*AddQ*) or additionally the words from all incorrect answer candidates (*AddQA*). While these attacks do not take any particular measures to prevent the added sentence from contradicting the correct answer, this is very unlikely given the ungrammatical nature of the generated word sequences.

The first two rows in Table 4 show the effect of appending a random sentence to the plot.[8] The impact on performance is fairly small indicating the robustness of both models. However, after only two epochs of optimizing the selected words in the added sentence, the performance drops markedly under all variants of the sentence-level black-box attacks as displayed in the two bottom rows of Table 4. While composing the sentence of just common English words (*AddC*) does not affect the models too much, adding words from the question

---

[8]As this attack is computationally very expensive we only ran it on a random subset of 200 validation questions for two optimization epochs of the distractor sentence.

| | Attack optimized for | |
Evaluated systems	CNN	RNN-LSTM
CNN	13.61	21.50
RNN-LSTM	22.06	23.22

Table 5: *AddQA* attack results when testing models on adversarial examples optimized to fool another model (averaged results of nine models).

Systems	Average	Ensemble
CNN	31.59	32.07
RNN-LSTM	32.61	32.17

Table 6: Adversarial accuracies on the validation set under the sentence-level white-box attack based on removal of the plot sentence with highest attention (averaged results of nine models).

and incorrect answers (*AddQA*) is most detrimental and causes both models to perform at or even below chance level. The models' performance under *AddQ*, where the distractor sentence does not contain answer candidate words, is much higher than under *AddQA*. We observe that the models can be easily distracted by adding a single sequence of significant words, even though it bears no semantic relation to the rest of the plot. This suggests that both models heavily rely on the content of the provided answer candidates and might just perform matching of learned patterns to select the right answer.

Another observation is that the RNN-LSTM models outperform the CNN models by a large margin under all attacks. The stronger the attack, the larger is the performance gap, indicating that RNNs depend less on pattern matching and are less prone to this kind of attack. Figure 5 and 6 in the appendix provide an example of the sentence and word attention distributions of a CNN model before and after the *AddQA* attack.

To test the transferability of the adversarial examples across models, we test the CNN models on the adversarial examples optimized to fool the RNN models and vice versa. As Table 5 shows, the performance of both models is degraded to the same level independent of the model the attack was optimized for. This suggests that both models suffer from similar weaknesses.

A straightforward way to try to improve the models' robustness against adversarial attacks is to mix some adversarial examples into the training data. Jia and Liang (2017) evaluated this for the *AddAny* attack on SQuAD and found that training on a mix of adversarial and original samples indeed improves the performance with respect to this specific adversarial attack. Yet a slight change of the attack, e.g. adding the distracting sentence as first instead of last sentence, made the adversarially-trained models to fail almost as badly as without adversarial training. Therefore,

we argue that it is more promising to look for general improvements of the model than training on adversarial examples generated by a specific attack.

5.4 Sentence-level White-box Attack

Instead of modifying the words in the sentence we also attempted to attack the model by removing the whole plot sentence with the highest attention. In this experiment, we wanted to test (1) if the model really focuses on the most important sentence, so it would become more difficult to answer the question, and (2) if the model is able to pick up more subtle cues or perform answer elimination to still be able to infer the correct answer with some confidence. As can be seen from Table 6, the accuracy decreases dramatically for both models by removing only one plot sentence. This proves that the model indeed focuses on the correct sentence where the hint to answer a question is given. These results correspond to those of the white-box attack at word level with a large number k of modified words. For the remaining 30% of correctly answered questions we observed that sometimes the models still were able to answer correctly because of the context information provided in other plot sentences.

We also measured human performance under this attack on 20 randomly sampled questions on distinct plots, where the sentence containing the answer information was removed. A single annotator (one of the authors of this paper) achieved 55% accuracy on this task, which is way above chance level and the models' performance. The human reported to be able to answer nine questions with reasonable confidence by deducing from other information distributed across the plots; two answers were correct by guessing. Answering the questions under this attack took a lot of time and effort. This highlights the weakness of the model to give answers in more complex scenarios where the answer is less obvious.

6 Human vs. Machine Processing

In order to gain insights how to further improve machine reading comprehension, we performed a case study in which a human was asked to answer difficult questions that none of 11 CNN or RNN-LSTM models solved correctly. The human evaluator obtained the plots and the questions with the corresponding five answer candidates; having access to the information in the same manner as the models. There are clear motifs in the type of reasoning and logic required, inherent to human cognition. In this light, we aim at inferring the gap between the model's and human cognitive information processing to identify problems followed by potential solutions.

Since we were especially interested in getting insights on human strategies for the cases where our models failed, 50 difficult questions of the CNN models in the validation set were analyzed by a human evaluator. All of the questions were correctly answered by the human evaluator noticing several key postulates: textual entailment, choice by elimination, referential knowledge and their combination (Hummel and Holyoak, 2005).

Textual entailment is required to solve 60% of the questions, such as the question "*What do Matt, Steve, and Andrew record themselves doing weeks after their experience in the woods?*" with the relevant sentence "*Weeks later, Andrew, Matt, and Steve record themselves as they display telekinetic abilities, but begin bleeding from their noses when they overexert themselves*". The human predicts the correct answer, "*Moving objects with their mind*", based on world knowledge of the word *telekinetic*. A further example in this regard is the question "*Do the robbers take people in the bank as hostage?*" with the relevant sentence "*They seize control of a Manhattan bank and take the employees and patrons hostage.*" The human picks the correct answer "*Yes, they do*", as *people in the bank* is a hypernym of *employees* and *patrons* in this context. Lacking notion of these semantic relations, the model answers incorrectly.

The process of elimination and heuristics proved essential to solve 44% of the questions. One example is "*Where is New Penzance located?*" with the relevant sentence "*In September 1965, on a New England island called New Penzance, 12-year-old orphan Sam Shakusky is attending Camp Ivanhoe [. . .]*". The human could not infer the answer "*Off the coast of North Car-*

olina" from reading the plot alone, as this region is not inherently known to be associated with New England, the location mentioned in the plot. However, by using the process of elimination and heuristics, the annotator was able to deduce the likely answer with the certainty that the other candidates are less likely correct. Additionally, with the ranking of keywords, humans can infer the correct answer in examples such as the question "*What kind of classes does Toula take up?*", with the relevant sentence "*After some persuasion by his wife, Maria [. . .], Gus reluctantly permits Toula to begin taking computer classes at a local community college [. . .]*". In this case, the human identified the keywords *classes* and *Toula*. The word *classes* obtains a higher ranking as it appears in three of the five possible answers. Ultimately, the correct prediction was made using ranking and the main keyword to find the correct answer, "*Computer classes*".

Referential knowledge is presumed in 36% of the questions, e.g. in the question "*What does Stigman do with the money?*" with the relevant sentence "*After the heist, Stigman follows orders to betray Trench and escape with the money, managing to pull his gun right as Trench is about to pull his own*". The human chooses the correct answer "*He takes it*", however the models select either "*He splits it with Trench*" or "*He leaves it in the vault*". When analyzing the plot, we can see that the two pronouns, *He* and *it*, are ambiguous to the models but clear to the human, leading to incorrect model predictions. The variance is due to the notion that humans have the ability to understand the referents from the plot. Another example where lack of referential knowledge effects the models' performances, but not the human, can be observed with the question "*What happens to any human who is encountered in Narnia?*" with the relevant sentence "*If a human is encountered they are to be brought to her*". The human is able to select the correct answer, "*They are to be brought to the White Witch*", even though the plot refers to the character by the pronoun *her*.

Furthermore, it is apparent that many questions expect a combination of various reasoning skills. The question "*What is Xavier's mutant ability?*" with the relevant sentence "*Present are Lehnsherr, now known as Magneto, and the telepathic Professor Charles Xavier, who privately discuss their differing views on the relationship between humans*

and mutants", depicts this phenomena. The human reports that she utilized the keywords *Xavier*, *mutant* and *ability*, raking *Xavier* more predominantly. By identifying *Professor Charles Xavier* in the plot as referent of the most important keyword, she could eliminate the incorrect answers.

The human evaluator also conducted an extensive comparison of the baseline word-level models with the hierarchical CNN models. In particular, she looked at those questions where the performance of both model types differed most (in terms of the number of models out of 11 that solved the question correctly). There are 101 validation questions which the majority of hierarchical CNN models solved correctly but only a minority (at least six less) word-level models did so. These were compared to the 28 validation questions on which the word-level models outperformed the hierarchical ones.

No prevailing pattern could be identified for the few instances where the word-level models did better than the hierarchical ones. Yet, we found some evidence that the hierarchical models seem to do better for questions requiring matching longer answer candidates and handling lexical variation. An example for such a more complex question is "*What happens to Deon in the end?*". The relevant plot sentence is "*He then transfers the dying Deon's consciousness into a spare robot through the modified MOOSE helmet*", and the correct answer "*His consciousness is transferred into a robot*". All answer candidates consist of at least five words; the lexical overlap between the question and correct answer with the plot sentence is just {*into, a, robot*}. While only two baseline models identify the correct answer, all but one of the hierarchical models do so.

Additionally, among the 101 questions where the hierarchical models do far better than the word-level models there are only very few (18) questions where none of the word-level models predicted the correct answer. It seems to be the case that the hierarchical structure helps the model to gain confidence, causing more models to make the correct prediction. An example for this is the question "*What does Lucius tell Harry?*", where the relevant sentence is "*Lucius reveals that Harry only saw a dream of Sirius being tortured; it was a method to lure Harry into the Death Eaters' grasp, not an actual situation.*", and the correct answer is "*His vision of Sirius being tortured was a dream used to lure Harry to the Death*". The majority of the word-level models predicted an incorrect answer "*His vision of Sirius being tortured was true*", and only five of them selected the correct answer. In contrast, all hierarchical models solved this question correctly.

The same comparison was conducted between the hierarchical CNN and RNN-LSTM models. Although there are improvements, which indicate that sequential processing is better suited for QA tasks, the RNN-LSTM models exhibit the same fundamental drawbacks. They suffer from coreference errors, lack the entailment ability, and are inefficient at keyword elimination. This observation reveals the fundamental weaknesses of our proposed network architecture and indicates directions for future improvements.

7   Conclusion

We proposed a machine reading comprehension model based on the compare-aggregate framework with a hierarchical attention structure that achieves state-of-the-art results on the MovieQA question answering dataset, greatly outperforming previous models. Then, we explored the limitations of our models and the behavioral difference between CNNs and RNN-LSTMs with adversarial examples generated at different linguistic levels (word vs. sentence level) and from different adversary's knowledge (black-box vs. white-box). In general, RNN-LSTM models outperformed CNN models, but our results for sentence-level black-box attacks indicate they might share the same weaknesses.

Finally, our intensive analysis on the differences between the model and human inference suggest that both models seem to learn matching patterns to select the right answer rather than performing plausible inferences as humans do. The results of these studies also imply that other human like processing mechanism such as referential relations, implicit real world knowledge, i.e., entailment, and answer by elimination via ranking plausibility (Hummel and Holyoak, 2005) should be integrated in the system to further advance machine reading comprehension.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. *arXiv preprint arXiv:1606.02858*.

Ronan Collobert and Jason Weston. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, New York, New York, USA. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Li Deng and John C Platt. 2014. Ensemble deep learning for speech recognition. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Daria Dzendzik, Carl Vogel, and Qun Liu. 2017. Who framed roger rabbit? multiple choice questions answering about movie plot.

Jeffrey L Elman. 1990. Finding Structure in Time. *Cognitive science*, 14(2).

Gerd Gigerenzer and Wolfgang Gaissmaier. 2011. Heuristic decision making. *Annual review of psychology*, 62:451–482.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8).

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

John E Hummel and Keith J Holyoak. 2005. Relational reasoning in a neurally plausible cognitive architecture: An overview of the lisa project. *Current Directions in Psychological Science*, 14(3):153–157.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404.

Tzu-Chien Liu, Yu-Hsueh Wu, and Hung-yi Lee. 2017. Attention-based CNN matching net. *CoRR*, abs/1709.05036.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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. Association for Computational Linguistics.

Ivan Sanchez, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of nli models: Uncovering the influence of three factors on robustness. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1975–1985.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640.

Amos Tversky. 1972. Elimination by aspects: A theory of choice. *Psychological review*, 79(4):281.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of NAACL HLT*.

Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. 1990. Phoneme recognition using time-delay neural networks. In *Readings in speech recognition*, pages 393–404. Elsevier.

Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *CoRR*, abs/1611.01747.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *HLT-NAACL*.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Xiaoyong Yuan, Pan He, Qile Zhu, Rajendra Rana Bhat, and Xiaolin Li. 2017. Adversarial examples: Attacks and defenses for deep learning. *arXiv preprint arXiv:1712.07107*.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.

A Supplemental Material

A.1 Model Details

The word embeddings are initialized from 300-dimensional pretrained GloVe vectors (Pennington et al., 2014)[9] that are kept fixed during training. Words not contained in these embeddings are initialized randomly, while all updatable weights are initialized according to the Xavier initialization (Glorot and Bengio, 2010).

We train all models for five epochs, evaluating their performance on the validation set after each epoch. We keep the model with the best validation accuracy, which is usually achieved after one, or sometimes two or three training epochs.

For the CNN models, we started with the hyperparameters of Wang and Jiang (2016) and tuned the dropout rate of the pretrained embeddings, learning rate, and L2 regularization weight on the validation set. For the RNN-LSTM model we started from the hierarchical CNN model's hyperparameters and tuned the dropout and learning rate again. Table 7 lists the hyperparameters for our models.

	CNN	RNN-LSTM
embedding size	300	300
weight initialization	xavier	xavier
batch size	30	30
optimizer	Adam	Adam
dropout rate	0.0	0.0
learning rate	0.001	0.0025
regularization	L2	L2
reg. weights	0.0001	0.0001
size of dense layers	150	150
CNN kernel heights	[1,3,5]	N/A
LSTM units	N/A	150

Table 7: Hyperparameters for the CNN and RNN-LSTM hierarchical attention-based compare-aggregate models.

A.2 Adversarial Examples

Figure 4 shows a MovieQA example question and an adversarial sentence generated by the *AddQA* approach. The adversarial sentence that is appended to the plot has a very high overlap with the question and one of the wrong answers. Although the sentence is grammatically wrong and meaningless, our model picked the wrong answer 4) in-

stead of the correct 1). Figure 5 and 6 illustrate the sentence and the word attention weights of a CNN model before and after the adversarial attack.

Adversarial sentence: what aziz what do do what clothing opens do do

Question: What does Aziz do after he moves to Kashmir?

Candidate Answers:

0) He opens a mosque
1) **He opens a clinic**
2) He opens a school
3) He becomes a monk
4) He opens a clothing store

Figure 4: *MovieQA* example question and the generated adversarial sentence using *AddQA*.
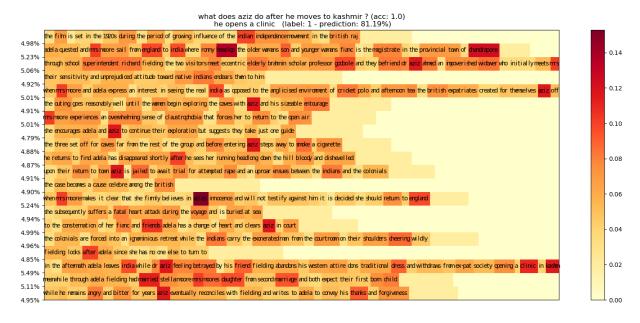

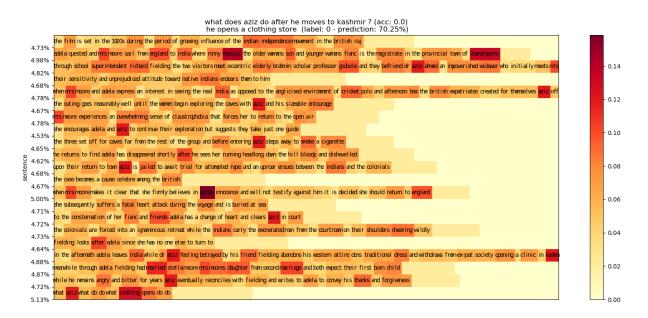Figure 5: Attention weights visualization of the CNN model.

Figure 6: Attention weights visualization of the CNN model after attacking it with *AddQA*, which added the final sentence to the plot.