DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension

Amrita Saha IBM Research
Rahul Aralikatte IBM Research
Mitesh M. Khapra IIT Madras
Karthik Sankaranarayanan IBM Research
{amrsaha4,rahul.a.r,kartsank}@in.ibm.com
miteshk@cse.iitm.ac.in

Abstract

We propose DuoRC, a novel dataset for Reading Comprehension (RC) that motivates several new challenges for neural approaches in language understanding beyond those offered by existing RC datasets. DuoRC contains 186,089 unique question-answer pairs created from a collection of 7680 pairs of movie plots where each pair in the collection reflects two versions of the same movie - one from Wikipedia and the other from IMDb - written by two different authors. We asked crowdsourced workers to create questions from one version of the plot and a different set of workers to extract or synthesize answers from the other version. This unique characteristic of DuoRC where questions and answers are created from different versions of a document narrating the same underlying story, ensures by design, that there is very little lexical overlap between the questions created from one version and the segments containing the answer in the other version. Further, since the two versions have different levels of plot detail, narration style, vocabulary, *etc*., answering questions from the second version requires deeper language understanding and incorporating external background knowledge. Additionally, the narrative style of passages arising from movie plots (as opposed to typical descriptive passages in existing datasets) exhibits the need to perform complex reasoning over events across multiple sentences. Indeed, we observe that state-of-the-art neural RC models which have achieved near human performance on the SQuAD dataset (Rajpurkar et al., 2016b), even when coupled with traditional NLP techniques to address the challenges presented in DuoRC exhibit very poor performance (F1 score of 37.42% on DuoRC v/s 86% on SQuAD dataset). This opens up several interesting research avenues wherein DuoRC could complement other RC datasets to explore novel neural approaches for studying language understanding.

1 Introduction

Natural Language Understanding is widely accepted to be one of the key capabilities required for AI systems. Scientific progress on this endeavor is measured through multiple tasks such as machine translation, reading comprehension, question-answering, and others, each of which requires the machine to demonstrate the ability to "comprehend" the given textual input (apart from other aspects) and achieve their task-specific goals. In particular, Reading Comprehension (RC) systems are required to "understand" a given text passage as input and then answer questions based on it. *It is therefore critical, that the dataset benchmarks established for the RC task keep progressing in complexity to reflect the challenges that arise in true language understanding, thereby enabling the development of models and techniques to solve these challenges.*

For RC in particular, there has been significant progress over the recent years with several benchmark datasets, the most popular of which are the SQuAD dataset (Rajpurkar et al., 2016a), TriviaQA (Joshi et al., 2017), MS MARCO (Nguyen et al., 2016), MovieQA (Tapaswi et al., 2016) and cloze-style datasets (Mostafazadeh et al., 2016; Onishi et al., 2016; Hermann et al., 2015). However, these benchmarks, owing to both the nature of the passages and the QA pairs to evaluate the RC task, have 2 primary limitations in studying language understanding: (i) Other than MovieQA, which is

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1683–1693
Melbourne, Australia, July 15 - 20, 2018.

a small dataset of 15K QA pairs, all other large-scale RC datasets deal only with factual descriptive passages and not narratives (involving events with causality linkages that require reasoning and background knowledge) which is the case with a lot of real-world content such as story books, movies, news reports, etc. (ii) their questions possess a large lexical overlap with segments of the passage, or have a high noise level in QA pairs themselves. As demonstrated by recent work, this makes it easy for even simple keyword matching algorithms to achieve high accuracy (Weissenborn et al., 2017). In fact, these models have been shown to perform poorly in the presence of adversarially inserted sentences which have a high word overlap with the question but do not contain the answer (Jia and Liang, 2017). While this problem does not exist in TriviaQA it is admittedly noisy because of the use of distant supervision. Similarly, for cloze-style datasets, due to the automatic question generation process, it is very easy for current models to reach near human performance (Cui, 2017). This therefore limits the complexity in language understanding that a machine is required to demonstrate to do well on the RC task.

Motivated by these shortcomings and to push the state-of-the-art in language understanding in RC, in this paper we propose DuoRC, which specifically presents the following challenges beyond the existing datasets:

1. DuoRC is especially designed to contain a large number of questions with low lexical overlap between questions and their corresponding passages.
2. It requires the use of background and commonsense knowledge to arrive at the answer and go beyond the content of the passage itself.
3. It contains narrative passages from movie plots that require complex reasoning across multiple sentences to infer the answer.
4. Several of the questions in DuoRC, while seeming relevant, cannot actually be answered from the given passage, thereby requiring the machine to detect the *unanswerability* of questions.

In order to capture these four challenges, DuoRC contains QA pairs created from pairs of documents describing movie plots which were gathered as follows. Each document in a pair is a different version of the same movie plot written by different authors; one version of the plot is taken from the Wikipedia page of the movie whereas the other from its IMDb page (see Fig. 1 for portions of an example pair of plots from the movie "Twelve Monkeys"). We first showed crowd workers on Amazon Mechanical Turk (AMT) the *first* version of the plot and asked them to create QA pairs from it. We then showed the *second* version of the plot along with the questions created from the *first* version to a different set of workers on AMT and asked them to provide answers by reading the second version only. Since the two versions contain different levels of plot detail, narration style, vocabulary, etc., answering questions from the second version exhibits all of the four challenges mentioned above.

We now make several interesting observations from the example in Fig. 1. For 4 out of the 8 questions (Q1, Q2, Q4, and Q7), though the answers extracted from the two plots are exactly the same, the analysis required to arrive at this answer is very different in the two cases. In particular, for Q1 even though there is no explicit mention of *the prisoner living in a subterranean shelter* and hence no lexical overlap with the question, the workers were still able to infer that the answer is *Philadelphia* because that is the city to which James Cole travels to for his mission. Another interesting characteristic of this dataset is that for a few questions (Q6, Q8) alternative but valid answers are obtained from the second plot. Further, note the kind of complex reasoning required for answering Q8 where the machine needs to resolve coreferences over multiple sentences (*that man* refers to *Dr. Peters*) and use common sense knowledge that if an item clears an airport screening, then a person can likely board the plane with it. To re-emphasize, these examples exhibit the need for machines to demonstrate new capabilities in RC such as: (i) employing a knowledge graph (e.g. to know that Philadelphia is a city in Q1), (ii) common-sense knowledge (e.g., *clearing airport security* implies *boarding*) (iii) paraphrase/semantic understanding (e.g. revolver is a type of handgun in Q7) (iv) multiple-sentence inferencing across events in the passage including coreference resolution of named entities and nouns, and (v) educated guesswork when the question is not directly answerable but there are subtle hints in the passage (as in Q1). Finally, for quite a few questions, there wasn't sufficient information in the second plot to obtain their answers. In such cases, the workers marked the question as "unanswerable". This brings out a very important challenge for machines (detect *unanswerability* of questions)

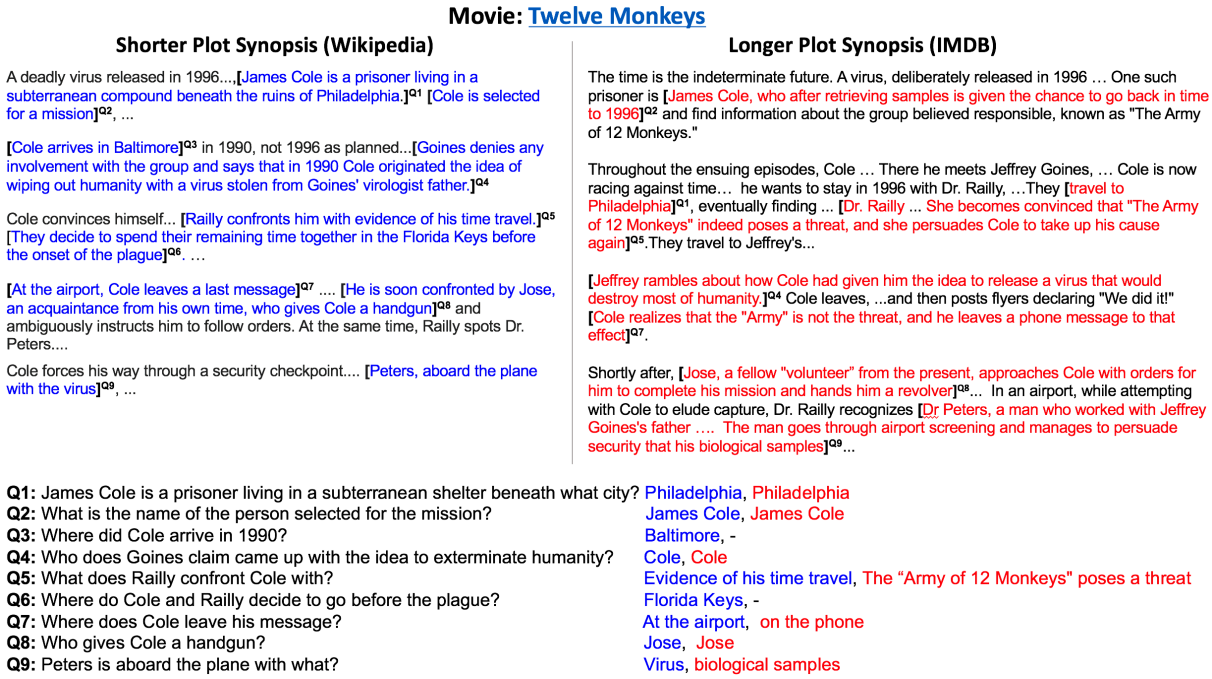

Figure 1: Example QA pairs obtained from the original movie plot and the paraphrased plot. The relevant spans needed for answering the corresponding question are highlighted in blue and red with the respective question numbers. Note that the span highlighting shown here is for illustrative purposes only and is not available in the dataset.

because a practical system should be able to know when it is not possible for it to answer a question given the data available to it, and in such cases, possibly delegate the task to a human instead.

Current RC systems built using existing datasets are far from possessing these capabilities to solve the above challenges. In Section 4, we seek to establish solid baselines for DuoRC employing state-of-the-art RC models coupled with a collection of standard NLP techniques to address few of the above challenges. Proposing novel neural models that solve all of the challenges in DuoRC is out of the scope of this paper. Our experiments demonstrate that when the existing state-of-the-art RC systems are trained and evaluated on DuoRC they perform poorly leaving a lot of scope for improvement and open new avenues for research in RC. Do note that this dataset is not a substitute for existing RC datasets but can be coupled with them to collectively address a large set of challenges in language understanding with RC (*the more the merrier*).

2 Related Work

Over the past few years, there has been a surge in datasets for Reading Comprehension. Most of these datasets differ in the manner in which questions and answers are created. For example, in SQuAD (Rajpurkar et al., 2016a), NewsQA (Trischler et al., 2016), TriviaQA (Joshi et al., 2017) and MovieQA (Tapaswi et al., 2016) the answers correspond to a span in the document. MS-MARCO uses web queries as questions and the answers are synthesized by workers from documents relevant to the query. On the other hand, in most cloze-style datasets (Mostafazadeh et al., 2016; Onishi et al., 2016) the questions are created automatically by deleting a word/entity from a sentence. There are also some datasets for RC with multiple choice questions (Richardson et al., 2013; Berant et al., 2014; Lai et al., 2017) where the task is to select one among k given candidate answers.

Another notable RC Dataset is NarrativeQA(s Koˇ ciský et al., 2018) which contains 40K QA pairs created from plot summaries of movies. It poses two tasks, where the first task involves reading the plot summaries from which the QA pairs were annotated and the second task is read the entire book or movie script (which is usually 60K words long) instead of the summary to answer the question. As acknowledged by the authors, while the first task is similar in scope to the previous datasets, the second task is at present, intractable for existing neural models, owing to the length of the passage. Due to the kind of the challenges presented by their second task, it is not comparable to our dataset and is much more futuristic in nature.

Given that there are already a few datasets for RC, a natural question to ask is "*Do we really need any more datasets?*". We believe that the answer to this question is *yes*. Each new dataset brings in new challenges and contributes towards building better QA systems. It keeps researchers on their toes and prevents research from stagnating once state-of-the-art results are achieved on one dataset. A classic example of this is the CoNLL NER dataset (Tjong Kim Sang and De Meulder, 2003). While several NER systems (Passos et al., 2014) gave close to human performance on this dataset, NER on general web text, domain specific text, noisy social media text is still an unsolved problem (mainly due to the lack of representative datasets which cover the real-world challenges of NER). In this context, DuoRC presents 4 new challenges mentioned earlier which are not exhibited in existing RC datasets and would thus enable exploring novel neural approaches in complex language understanding. The hope is that all these datasets (including ours) will collectively help in addressing a wide range of challenges in QA and prevent stagnation via overfitting on a single dataset.

3 Dataset

In this section, we elaborate on the three phases of our dataset collection process.

Extracting parallel movie plots: We first collected top 40K movies from IMDb across different genres (crime, drama, comedy, etc.) whose plot synopsis were crawled from Wikipedia as well as IMDb. We retained only 7680 movies for which both the plots were available and longer than 100 words. In general, we found that the IMDb plots were usually longer (avg. length 926 words) and more descriptive than the Wikipedia plots (avg. length 580 words). To make sure that the content between the two plots are indeed different and one is not just a subset of another, we calculated word-level jaccard distance between them i.e. the ratio of intersection to union of the bag-of-words in the two plots and found it to be 26%. This indicates that one of the plots is usually longer and descriptive, and, the two plots are infact quite different, even though the information content is very similar.

Collecting QA pairs from shorter version of the plot (*SelfRC*): As mentioned earlier, on average the longer version of the plot is almost double the size of the shorter version which is itself usually 500 words long. Intuitively, the longer version should have more details and the questions asked from the shorter version should be answerable from the longer one. Hence, we first showed the shorter version of the plot to workers on AMT and asked them to create QA pairs from it. The instructions given to the workers for this phase are as follows: (i) the answer must preferably be a single word or a short phrase, (ii) subjective questions (like asking for opinion) are not allowed, (iii) questions should be answerable only from the passage and not require any external knowledge, and (iv) questions and answers should be well formed and grammatically correct. The workers were also given freedom to either pick an answer which directly matches a span in the document or synthesize the answer from scratch. This option allowed them to be creative and ask hard questions where possible. We found that in 70% of the cases the workers picked an answer directly from the document and in 30% of the cases they synthesized the answer. We thus collected 85,773 such QA pairs along with their corresponding documents. We refer to this as the *SelfRC* dataset because the answers were derived from the same document from which the questions were asked.

Collecting answers from longer version of the plot (*ParaphraseRC*): We then paired the questions from the *SelfRC* dataset with the corresponding longer version of the plot and showed it to a different set of AMT workers asking them to answer these questions from the longer version of the plot. They now have the option to either (i) select an answer which matches a span in the longer version, (ii) synthesize the answer from scratch, or (iii) mark the question not-answerable because of lack of information in the given passage. One trick we used to reduce the fatigue of workers (caused by reading long pieces of text), and thus maintain the answer quality is to split the long plots into multiple segments. Every question obtained from the first phase of annotation is paired separately with each of these segments and each (question, segment) pair is posted as a different job. With this approach, we essentially get multiple answers to the same question, if it is answerable from more than one segment. However, on an average we get approximately one unique answer for each question. We found that in 50% of the cases the workers selected an answer which matched a span in the document, whereas in 37% cases they synthesized the answer and in 13% cases they said that question was not answerable. The workers were strictly instructed to

keep the answers short, derive the answer from the plot and use *general* knowledge or logic to answer the questions. They were not allowed to rely on *personal* knowledge about the movie (in any case given the large number of movies in our dataset the chance of a worker remembering all the plot details for a given movie is very less). For quality assessment purposes, various levels of manual and semi-automated inspections were done, especially in the second phase of annotation, such as:(i) weeding out annotators who mark majority of answers as non-answerable, by taking into account their response time, and (ii) annotators for whom a high percentage of answers have no entity (or noun phrase) overlap with the entire passage were subjected to strict manual inspection and blacklisted if necessary. Further, a wait period of 2-3 weeks was deliberately introduced between the two phases of data collection to ensure the availability of a fresh pool of workers as well as to reduce information bias among workers common to both the tasks. Overall 2559 workers took part in the first phase of the annotation, and 8021 workers in the second phase. Only 703 workers were common between the phases.

We refer to this dataset, where the questions are taken from one version of the document and the answers are obtained from a different version, as *ParaphraseRC* which contains 100,316 such {*question, answer, document*} triplets. Overall, 62% of the questions in *SelfRC* and *ParaphraseRC* have partial overlap in their answers, which is indicative of the fact that quality is reasonable. The remaining 38% where there is no overlap can be attributed to non-answerablity of the question from the bigger plot, information gap, or paraphrasing of information between the two plots.

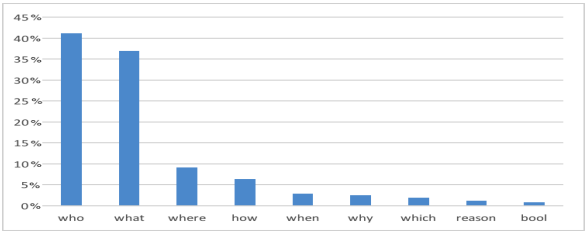

Figure 2: Analysis of the Question Types

Note that the number of unique questions in the *ParaphraseRC* dataset is the same as that in *SelfRC* because we do not create any new questions from the longer version of the plot. We end up with a greater number of {*question, answer, document*} triplets in *ParaphraseRC* as compared to *SelfRC* (100,316 v/s 85,773) since movies that are remakes of a previous movie had very little difference in their Wikipedia plots. Therefore, we did not separately collect questions from the Wikipedia plot of the remake. However, the IMDb plots of the two movies are very different and so we have two different longer versions of the movie (one for the original and one for the remake). We can thus pair the questions created from the Wikipedia plot with both the IMDb versions of the plot thus augmenting the {*question, answer, document*} triplets.

Another notable observation is that in many cases the answers to the same question are different in the two versions. Specifically, only 40.7% of the questions have the same answer in the two documents. For around 37.8% of the questions there is no overlap between the words in the two answers. For the remaining 21% of the questions there is a partial overlap between the two answers. For e.g., the answer derived from the shorter version could be "using his wife's gun" and from the longer version could be "with Dana's handgun" where Dana is the name of the wife. In Appendix **A**, we provide a few randomly picked examples from our dataset which should convince the reader of the difficulty of *ParaphraseRC* and its differences with *SelfRC*. We refer to this combined dataset containing a total of 186,089 instances as *DuoRC*[1]. Fig. 2 shows the distribution of different Wh-type questions in our dataset. Some interesting comparative analysis are presented in Table 1 and also in Appendix **B**. In Table 1, we compare various RC datasets with two embodiments of our dataset i.e. the *SelfRC* and *ParaphraseRC*. We use NER and noun phrase/verb phrase extraction over the entire dataset to iden-

Metrics for Comparative Analysis	**Movie QA**	**NarrativeQA over plot-summaries**	**Self-RC**	**Paraph-raseRC**
Avg. word distance	20.67	24.94	13.4	45.3
Avg. sentence distance	1.67	1.95	1.34	2.7
Number of sentences for inferencing	2.3	1.95	1.51	2.47
% of instances where both Query & Answer entities were found in passage	67.96	59.4	58.79	12.25
% of instances where Only Query entities were found in passage	59.61	61.77	63.39	47.05
% Length of the Longest Common sequence of non-stop words in Query (w.r.t Query Length) and Plot	25	26.26	38	21

Table 1: Comparison between various RC datasets

[1]The dataset is available at https://duorc.github.io

tify key entities in the question, plot and answer which is in turn used to compute the metrics mentioned in the table. The metrics "Avg word distance" and "Avg sentence distance" indicate the average distance (in terms of words/sentences) between the occurrence of the question entities and closest occurrence of the answer entities in the passage. "Number of sentences for inferencing" is indicative of the minimum number of sentences required to cover all the question and answer entities. It is evident that tackling *ParaphraseRC* is much harder than the others on account of (i) larger distance between the query and answer, (ii) low word-overlap between query & passage, and (iii) higher number of sentences required to infer an answer.

4 Models

In this section, we describe in detail the various state-of-the-art RC and language generation models along with a collection of traditional NLP techniques employed together that will serve to establish baseline performance on the DuoRC dataset.

Most of the current state-of-the-art models for RC assume that the answer corresponds to a span in the document and the task of the model is to predict this span. This is indeed true for the SQuAD, TriviaQA and NewsQA datasets. However, in our dataset, in many cases the answers do not correspond to an exact span in the document but are synthesized by humans. Specifically, for the *SelfRC* version of the dataset around 30% of the answers are synthesized and do not match a span in the document whereas for the *ParaphraseRC* task this number is 50%. Nevertheless, we could still leverage the advances made on the SQuAD dataset and adapt these span prediction models for our task. To do so, we propose to use two models. The first model is a basic span prediction model which we train and evaluate using only those instances in our dataset where the answer matches a span in the document. The purpose of this model is to establish whether even for instances where the answer matches a span in the document, our dataset is harder than the SQuAD dataset or not. Specifically, we want to explore the performance of state-of-the-art models (such as DCN (Xiong et al., 2016)), which exhibit near human results on the SQuAD dataset, on DuoRC (especially, in the *ParaphraseRC* setup). To do so, we seek to employ a good span prediction model for which (i) the performance is within 3-5% of the top performing model on the SQuAD leaderboard (Rajpurkar et al., 2016b) and (ii) the results are reproducible based on the code released by the authors of the paper. Note that the second criteria is important to ensure that the poor performance of the model is not due to incorrect implementation. The Bidirectional Attention Flow (BiDAF) model (Seo et al., 2016) satisfies these criteria and hence we employ this model. Due to space constraints, we do not provide details of the BiDAF model here and simply refer the reader to the original paper. In the remainder of this paper we will refer to this model as the *SpanModel*.

The second model that we employ is a two stage process which first predicts the span and then synthesizes the answers from the span. Here again, for the first step (*i.e.*, span prediction) we use the BiDAF model (Seo et al., 2016). The job of the second model is to then take the span (mini-document) and question (query) as input and generate the answer. For this, we employ a state-of-the-art query based abstractive summarization model (Nema et al., 2017) as this task is very similar to our task. Specifically, in query based abstractive summarization the training data is of the form {query, document, generated_summary} and in our case the training data is of the form {query, mini-document, generated_answer}. Once again we refer the reader to the original paper (Nema et al., 2017) for details of the model. We refer to this two stage model as the *GenModel*.

Note that (Tan et al., 2017) recently proposed an answer generation model for the MS MARCO dataset. However, the authors have not released their code and therefore, in the interest of reproducibility of our work, we omit incorporating this model in this paper.

Additional NLP pre-processing: Referring back to the example cited in Fig. 1, we reiterate that ideally a good model for *ParaphraseRC* would require: (i) employing a knowledge graph, (ii) common-sense knowledge (iii) paraphrase/semantic understanding (iv) multiple-sentence inferencing across events in the passage including coreference resolution of named entities and nouns, and (v) educated guesswork when the question is not directly answerable but there are subtle hints in the passage. While addressing all of these challenges in their entirety is beyond the scope of a single paper, in the interest of establishing a good baseline for DuoRC, we additionally

seek to address some of these challenges to a certain extent by using standard NLP techniques. Specifically, we look at the problems of paraphrase understanding, coreference resolution and handling long passages.

To do so, we prune the document and extract only those sentences which are most relevant to the question, so that the span detector does not need to look at the entire 900-word long *ParaphraseRC* plot. Now, since these relevant sentences are obtained not from the original but the paraphrased version of the document, they may have a very small word overlap with the question. For example, the question might contain the word "hand gun" and the relevant sentence in the document may contain the word "revolver". Further some of the named entities in the question may not be exactly present in the relevant sentence but may simply be co-referenced. To resolve these coreferences, we first employ the Stanford coreference resolution on the entire document. We then compute the fraction of words in a sentence which match a query word (ignoring stop words). Two words are considered to match if (a) they have the same surface form, or (b) one words is an inflected form of the word (e.g., river and rivers), or (c) the Glove (Pennington et al., 2014) and Skip-thought (Kiros et al., 2015) embeddings of the two words are very close to each other (two word vectors are considered to be close if one appears within the top 50 neighbors of the other), or (d) the two words appear in the same synset in Wordnet. We consider a sentence to be relevant for the question if at least 50% of the query words (ignoring stop words) match the words in the sentence. If none of the sentences in the document have atleast 50% overlap with the question, then we pick sentences having atleast a 30% overlap with the question. The selection of this threshold was based on manual observation of a small sample set. This observation gave us an idea of what a decent threshold value should be, that can have a reasonable precision and recall on the relevant snippet extraction step. Since this step was rule-based we could only employ such qualitative inspections to set this parameter. Also, since this step was targeted to have high recall, we relaxed the threshold to 30% if no match was found.

5 Experimental Setup

In the following sub-sections we describe (i) the evaluation metrics, and (ii) the choices considered for augmenting the training data for the answer generation model. Note that when creating the train, validation and test set, we ensure that the test set does not contain QA pairs for any movie that was seen during training. We split the movies in such a way that the resulting train, valid, test sets respectively contain 70%, 15% and 15% of the total number of QA pairs.

Span-Based Test Set and Full Test Set As mentioned earlier, the *SpanModel* only predicts the span in the document whereas the *GenModel* generates the answer after predicting the span. Ideally, the *SpanModel* should only be evaluated on those instances in the test set where the answer matches a span in the document. We refer to this subset of the test set as the *Span-based Test Set*. Though not ideal, we also evaluate the *SpanModel* model on the entire test set. This is not ideal because there are many answers in the test set which do not correspond to a span in the document whereas the model was only trained to predict spans. We refer to this as the *Full Test Set*. We also evaluate the *GenModel* on both the test sets.

Training Data for the GenModel As mentioned earlier, the *GenModel* contains two stages; the first stage predicts the span and the second stage then generates an answer from the predicted span. For the first step we plug-in the best performing *SpanModel* from our earlier exploration. To train the second stage we need training data of the form $\{x = span,\ y = answer\}$ which comes from two types of instances: one where the answer matches a span and the other where the answer is synthesized and the span corresponding to it is not known. In the first case x=y and there is nothing interesting for the model to learn (except for copying the input to the output). In the second case x is not known. To overcome this problem, for the second type of instances, we consider various approaches for finding the approximate span from which the answer could have been generated, and augment the training data with $\{x = approx\_span,\ y = answer\}$.
The easiest method was to simply treat the entire document as the true span from which the answer was generated ($x = document,\ y = answer$). The second alternative that we tried was to first extract the named entities, noun phrases and verb phrases from the question and create a lucene query from these components. We then used the lucene search engine to extract the most relevant portions of the document given this query. We then considered this portion of the document as the true span (as

opposed to treating the entire document as the true span). Note that lucene could return multiple relevant spans in which case we treat all these $\{x$ = *approx_span, y= answer*$\}$ as training instances. Another alternative was to find the longest common subsequence (LCS) between the document and the question and treat this subsequence as the span from which the answer was generated. Of these, we found that the model trained using $\{x$ = *approx_span, y= answer*$\}$ pairs created using the LCS based method gave the best results. We report numbers only for this model.

Evaluation Metrics Similar to (Rajpurkar et al., 2016a) we use Accuracy and F-score as the evaluation metrics. We also report the BLEU scores for each task. While accuracy, being a stricter metric, considers a predicted answer to be correct only if it exactly matches the true answer, F-score and BLEU also give credit to predictions partially overlapping with the true answer.

6 Results and Discussions

The results of our experiments are summarized in Tables 2 to 4 which we discuss in the following sub-sections.

Preprocessing step of Relevant Subplot Extraction	Plot Compression	Answer Recall
WordNet synonym + Glove based paraphrase	30%	66.51%
WordNet synonym + Glove based paraphrase on Coref resolved plots	50%	84.10%
WordNet synonym + Glove + Skip-thought based paraphrase on Coref resolved plots	48%	85%

Table 2: Performance of the preprocessing. Plot compression is the % size of the extracted plot w.r.t the original plot size

SelfRC	**Span Test**			**Full Test**		
	Acc.	**F1**	**BLEU**	**Acc.**	**F1**	**BLEU**
SpanModel	46.14	57.49	22.98	37.53	50.56	7.47
GenModel (with augmented training data)	16.45	26.97	7.61	15.31	24.05	5.50
ParaphraseRC	**Span Test**			**Full Test**		
	Acc.	**F1**	**BLEU**	**Acc.**	**F1**	**BLEU**
SpanModel	17.93	26.27	9.39	9.78	16.33	2.60
SpanModel with Preprocessed Data	27.49	35.10	12.78	14.92	21.53	2.75
GenModel (with augmented training data)	12.66	19.48	4.41	5.42	9.64	1.75

Table 3: Performance of the *SpanModel* and *GenModel* on the Span Test subset and the Full Test Set of the *Self* and *ParaphraseRC*.

***SpanModel* v/s *GenModel*:** Comparing the first two rows (*SelfRC*) and the last two rows (*ParaphraseRC*) of Table 3 we see that the *SpanModel* clearly outperforms the *GenModel*. This is not very surprising for two reasons. First, around 70% (and 50%) of the answers in *SelfRC* (and *ParaphraseRC*) respectively, match an exact span in the document so the *SpanModel* still has scope to do well on these answers. On the other hand, even if the first stage of the *GenModel* predicts the span correctly, the second stage could make an error in generating the correct answer from it because generation is a harder problem. For the second stage, it is expected that the *GenModel* should learn to copy the predicted span to produce the answer output (as is required in most cases) and only occasionally where necessary, generate an answer. However, surprisingly the *GenModel* fails to even do this. Manual inspection of the generated answers shows that in many cases the generator ends up generating either more or fewer words compared the true answer. This demonstrates the clear scope for the *GenModel* to perform better.

Train	**Test**	**Span Test**			**Full Test**		
		Acc.	**F1**	**BLEU**	**Acc.**	**F1**	**BLEU**
SelfRC	*SelfRC*	46.14	57.49	22.98	37.53	50.56	7.47
	ParaRC	27.85	36.82	14.48	15.16	22.70	3.90
	SelfRC+ ParaRC	37.79	48.05	18.72	25.05	35.01	5.34
ParaRC	*SelfRC*	34.85	45.71	16.01	28.25	40.16	5.15
	Para RC	19.74	27.57	9.84	10.78	17.13	2.75
	SelfRC+ ParaRC	27.94	37.42	13.00	18.50	27.31	3.75
SelfRC + ParaRC	*SelfRC*	49.66	61.45	25.87	40.24	54.04	8.42
	ParaRC	29.88	39.34	16.22	16.33	24.25	4.21
	SelfRC+ ParaRC	40.62	51.35	21.18	26.90	37.42	5.94

Table 4: Combined and Cross-Testing between *Self* and *ParaphraseRC* Dataset, by taking the best performing *SpanModel* from Table 3.*ParaRC* is an abbreviation of *ParaphraseRC*

SelfRC v/s ParaphraseRC: Comparing the *SelfRC* and *ParaphraseRC* numbers in Table 3, we observe that the performance of the models clearly drops for the latter task, thus validating our hypothesis that *ParaphraseRC* is a indeed a much harder task.

Effect of NLP pre-processing: As mentioned in Section 4, for *ParaphraseRC*, we first perform a few pre-processing steps to identify relevant sentences in the longer document. In order to evaluate whether the pre-processing method is effective, we compute: (i) the percentage of the document that gets pruned, and (ii) whether the true answer is present in the pruned document (i.e., average recall of the answer). We can compute the recall only for the span-based subset of the data since for the remaining data we do not know the true span. In Table 2, we report these two quantities for the span-based subset using different pruning strategies. Finally, comparing the *SpanModel* with and without

Paraphrasing in Table 3 for *ParaphraseRC*, we observe that the pre-processing step indeed improves the performance of the Span Detection Model.

Effect of oracle pre-processing: As noted in Section 3, the *ParaphraseRC* plot is almost double in length in comparison to the *SelfRC* plot, which while adding to the complexities of the former task, is clearly not the primary reason of the model's poor performance on that. To empirically validate this, we perform an Oracle pre-processing step, where, starting with the knowledge of the span containing the true answer, we extract a subplot around it such that the span is randomly located within that subplot and the average length of the subplot is similar to the *SelfRC* plots. The *SpanModel* with this Oracle preprocessed data exhibits a minor improvement in performance over that with rule-based preprocessing (1.6% in Accuracy and 4.3% in F1 over the Span Test), still failing to bridge the wide performance gap between the *SelfRC* and *ParaphraseRC* task.

Cross Testing We wanted to examine whether a model trained on *SelfRC* performs well on *ParaphraseRC* and vice-versa. We also wanted to evaluate if merging the two datasets improves the performance of the model. For this we experimented with various combinations of train and test data. The results of these experiments for the *SpanModel* are summarized in Table 4. The best performance is obtained when the model is trained on both (*SelfRC*) and *ParaphraseRC* and tested on *SelfRC* and the performance is poorest when *ParaphraseRC* is used for both. We believe this is because learning with the *ParaphraseRC* is more difficult given the wide range of challenges in this dataset.

Based on our experiments and empirical observations we believe that the *DuoRC* dataset indeed holds a lot of potential for advancing the horizon of complex language understanding by exposing newer challenges in this area.

7 Conclusion

In this paper we introduced DuoRC, a large scale RC dataset of 186K human-generated QA pairs created from 7680 pairs of parallel movie-plots, each pair taken from Wikipedia and IMDb. We then showed that this dataset, by design, ensures very little or no lexical overlap between the questions created from one version and segments containing answers in the other version. With this, we hope to introduce the RC community to new research challenges on QA requiring external knowledge and common-sense driven reasoning, deeper language understanding and multiple-sentence inferencing. Through our experiments, we show how the state-of-the-art RC models, which have achieved near human performance on the SQuAD dataset, perform poorly on our dataset, thus emphasizing the need to explore further avenues for research.

References

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. http://aclweb.org/anthology/D/D14/D14-1159.pdf.

Yiming Cui. 2017. Cloze explorer. https://github.com/ymcui/Eval-on-NN-of-RC/.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. pages 1693–1701. http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.

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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. pages 2011–2021. http://aclanthology.info/papers/D17-1214/d17-1214.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 1601–1611. https://doi.org/10.18653/v1/P17-1147.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. *CoRR* abs/1506.06726. http://arxiv.org/abs/1506.06726.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. *CoRR* abs/1704.04683. http://arxiv.org/abs/1704.04683.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *CoRR* abs/1604.01696. http://arxiv.org/abs/1604.01696.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. *CoRR* abs/1704.08300. http://arxiv.org/abs/1704.08300.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR* abs/1611.09268. http://arxiv.org/abs/1611.09268.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. *arXiv preprint arXiv:1608.05457* .

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014*. pages 78–86. http://aclweb.org/anthology/W/W14/W14-1609.pdf.

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. http://www.aclweb.org/anthology/D14-1162.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* .

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. Squad explorer. https://rajpurkar.github.io/SQuAD-explorer/.

Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. 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, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 193–203. http://aclweb.org/anthology/D/D13/D13-1020.pdf.

Tomáš Koˇ ciský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics* TBD:TBD. https://TBD.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR* abs/1611.01603. http://arxiv.org/abs/1611.01603.

Chuanqi Tan, Furu Wei, Nan Yang, Weifeng Lv, and Ming Zhou. 2017. S-net: From answer extraction to answer generation for machine reading comprehension. *CoRR* abs/1706.04815. http://arxiv.org/abs/1706.04815.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*. Edmonton, Canada, pages 142–147.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *CoRR* abs/1611.09830. http://arxiv.org/abs/1611.09830.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*. pages 271–280. https://doi.org/10.18653/v1/K17-1028.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *CoRR* abs/1611.01604. http://arxiv.org/abs/1611.01604.