Bend but Don't Break? Multi-Challenge Stress Test for QA Models

Hemant Pugaliya* **James Route*** **Kaixin Ma*** **Yixuan Geng** **Eric Nyberg**
Language Technologies Institute
Carnigie Mellon University
{hpugaliy, jroute, kaixinm, yixuang, ehn}@cs.cmu.edu

Abstract

The field of question answering (QA) has seen rapid growth in new tasks and modeling approaches in recent years. Large scale datasets and focus on challenging linguistic phenomena have driven development in neural models, some of which have achieved parity with human performance in limited cases. However, an examination of state-of-the-art model output reveals that a gap remains in reasoning ability compared to a human, and performance tends to degrade when models are exposed to less-constrained tasks. We are interested in more clearly defining the strengths and limitations of leading models across diverse QA challenges, intending to help future researchers with identifying pathways to generalizable performance. We conduct extensive qualitative and quantitative analyses on the results of four models across four datasets and relate common errors to model capabilities. We also illustrate limitations in the datasets we examine and discuss a way forward for achieving generalizable models and datasets that broadly test QA capabilities.

1 Introduction

Advancements in question answering, where a system generates a response to a natural language query, have led AI agents to demonstrate competency at increasingly sophisticated linguistic patterns and concepts. Neural models have achieved particularly strong results in machine reading and comprehension (MRC), a related task where a model answers questions from a given text passage. High scores on some MRC datasets, some of which even exceed human performance, seemingly imply that models are attaining a level of linguistic reasoning that approaches a human's. However, we suspect that the raw scores on MRC datasets do not fully convey the strengths and weaknesses of models, and we propose a more in-depth exploration of model results.

We investigate four publicly-available models, each of which take a different approach to QA and have attained high scores on at least one MRC dataset. We also select four MRC datasets that present a different set of challenges for the models. We aim to characterize how models perform on each challenge, going beyond reporting of standard scores like F1 or BLEU. Our goal is to understand how different models generalize to a wider range of challenges than a single dataset can provide, and determine if aspects of model design adapt well to certain conditions. We manually examine error cases and random samples of results from each model-dataset pair and employ a regression framework to model evaluation scores on various dataset characteristics.[1] Our analysis has revealed some key findings, as follows:

- Scores of high performing models are often underestimated because of noise or errors in the dataset (e.g., over 10% of a model's errors are factually correct answers scored as incorrect, as indicated in Sections 6.2-6.4).
- Manual error analysis, often overlooked when reporting new approaches, reveals useful model strengths. One example is the QANet model's apparent strong performance on multi-hop inference questions.
- Regression analysis can pinpoint dataset features that challenge models; for example, indicating that HotpotQA's difficulty stems from distractor sentences and at least partially from multihop inference, rather than simply resulting from long context lengths.

* Equal contribution

[1] Annotations are available at https://github.com/jamesrt95/Neural-QA-Eval

Based on our findings, we conclude with some guidelines which future researchers can benefit from while building new models and datasets.

2 Related Work

Wadhwa et al. (2018) explored the performance of several MRC models on SQuAD and inferred common areas of difficulty. Kaushik and Lipton (2018) examined model performance across several MRC datasets, including SQuAD. This study questioned the effective difficulty of MRC tasks by varying the amount of input data available to the models. Rondeau and Hazen (2018) presented a systematic approach for identifying the most salient features for a question's difficulty on SQuAD. They define question categories based on the number of models that could get the correct output on the question. Sugawara et al. (2017) analyzed 6 MRC datasets on the metrics of prerequisite skills and readability, which are defined from a human's perspective. Feng et al. (2018) explored model explainability on MRC and other tasks by reducing input spans until a given model failed to generate a correct prediction. Talmor and Berant (2019) investigated generalization and transferability of 10 MRC datasets and analyzed factors that contribute to these characteristics.

Our study casts a broader net by testing four MRC datasets against four models. The study tests a greater range of linguistic phenomena and examines a larger proportion of question-answer pairs. In addition, our quantitative analysis scales to larger data sizes. We focus on characterizing model outputs and errors, and in the process, make inferences about the MRC challenges. We adopt both automatic and manual analysis of QA pairs across all dataset-model pairs. We do not focus on explainability in this study, although we aim to conclude why a model performs in a certain way throughout our analysis.

3 Datasets

We selected four datasets for evaluating model performance, each of which we describe briefly. We chose datasets that are relatively well-known and test a variety of non-overlapping capabilities. Table 1 summarizes key characteristics for the datasets.

SQuAD (Rajpurkar et al., 2016) is one of the first large-scale extractive question answering datasets. We include SQuAD in this study because it is well-understood, and it tests a model's tolerance for paraphrasing and coreferences between the question and context. Although SQuAD 2.0 (Rajpurkar et al., 2018) is the most recent version of this dataset, we focus on SQuAD 1.1 because our selected models are not designed to handle the unanswerable questions in SQuAD 2.0.

Dataset	Data	Source	Answer	Size
SQuAD	Wikipedia	Crowd	Span	100K
HotpotQA	Wikipedia	Crowd	Span	113K
SearchQA	Web	Jeopardy	Span	140K
MSMARCO	Web	Bing	Free-form	1.01M

Table 1: Dataset Summary

HotpotQA (Yang et al., 2018) is similar to SQuAD but includes additional linguistic phenomena. HotpotQA stresses multihop reasoning, which requires a model to aggregate information from multiple relevant passages to locate the answers. It also contains questions that require a model to compare two entities and select the correct one. We use the distractor version of HotpotQA, where 10 passages are provided per question; two of the passages are relevant and the remaining eight contain keywords that appear in the question. We selected HotpotQA to test how well models handle consistently challenging multihop and comparison questions.

SearchQA (Dunn et al., 2017) is built using a different approach than SQuAD or HotpotQA. All question-answer pairs from the Jeopardy Challenge are collected and then augmented with text snippets from web pages retrieved by a search engine. Each question includes up to 51 snippets, and questions and snippets are cleaned to remove tokens such as stopwords. We selected SearchQA because it requires models to locate an answer within a uniquely large and noisy context, and the cleaning process creates a much more terse and uninterpretable text compared to the other datasets.

MSMARCO (Nguyen et al., 2016) is also a search-based dataset and was created using Bing queries from real users as questions and corresponding documents returned by the search engine as contexts. We include MSMARCO as the only dataset that requires models to freely generate answer sequences instead of selecting a span. Although most of the models we test are span-based, we aim to evaluate how well the models adapt to a different answer type.

4 Models

We also focus on diversity when selecting models. Each of the models described in this section is developed for a different task and they have relatively heterogeneous architecture. We specifically chose models that had strong performance on at least one popular QA dataset, particularly the ones used in this study. Some of the models were not designed to handle the challenges presented by one or more of the datasets; this is an intentional choice to measure how well a model generalizes to an out of domain task. We reduce the size of some models so that all training can be accomplished using equal hardware resources (single GPU)[2]. All changes are described in Section 5.

QANet (Yu et al., 2018) was originally developed for the SQuAD dataset and was state-of-the-art on the leaderboard in earlier 2018. The model consists of several convolutional encoding blocks, self-attention layers (Vaswani et al., 2017) and feed-forward layers. Finally, answer pointer layers (Seo et al., 2016) are used to predict start and end indices of the answer span. We used Google's implementation for our experiments[3]. To train QANet on single GPU, we reduce the number of encoder layers from 7 to 1.

BERT (Devlin et al., 2018) consists of stacked bidirectional transformer encoders and is pre-trained on large corpora for masked language modeling task and next sentence prediction. BERT has achieved state-of-the-art performance on several NLP tasks after fine-tuning, and a BERT ensemble occupied the top position on the SQuAD leaderboard. A final layer is added to BERT that predicts the start and end indices of the answer span. We select BERT for this study because we hypothesize that its strong performance across NLP tasks is indicative of generalizability on multiple QA datasets. We use the Pytorch implementation of BERT[4] and use the smaller BERT-base model. BERT-base SQuAD results are consistent with the Pytorch implementation but lower than the official SQuAD leaderboard which uses BERT-large.

Denoising Distantly Supervised(DS)-QA (Lin et al., 2018) is mainly aimed at improving Open-Domain Question Answering. The model employs a paragraph selector to filter out noisy paragraphs and a paragraph reader to extract the correct answer from those denoised paragraphs. The paragraph selector encodes all paragraphs and the question using LSTM layers and self-attention. A paragraph reader then estimates a probability distribution over all possible spans. This architecture is shown to be effective on many open-domain datasets like QuasarT(Dhingra et al., 2017), SearchQA(Dunn et al., 2017), TriviaQA(Joshi et al., 2017) and CuratedTREC(Baudis and Sedivý, 2015). We use the official implementation of this model.[5]

CommonSenseMultihop(CSM) (Bauer et al., 2018) generates an answer sequence rather than selecting a span. It uses an attention mechanism to reason over context and a pointer-generator decoder (See et al., 2017) to synthesize the answer. The model also applies common sense knowledge from an external knowledge base ConceptNet (Speer et al., 2016). The model encodes the context and question using Bi-LSTM layers, and BiDAF attention (Seo et al., 2016), then applies self-attention (Cheng et al., 2016) to perform multihop reasoning. The context is also attended by an encoded commonsense representation. Finally, the decoder generates the answer sequence and copies key spans from the context. This model has achieved promising performance on the NarrativeQA (Kocisky et al., 2018) and WikiHop (Welbl et al., 2018) datasets. We choose this model to test how it generalizes to extractive datasets and whether common sense knowledge is helpful for other QA tasks. We use the official implementation of this model. [6]

5 Experiments

We train the four selected models on each dataset as outlined in Sections 3 and 4, and where possible replicate the same training procedures used for the original models. Many of the datasets have features that the models were not designed to handle, in these cases, we perform preprocessing to adapt the dataset to the model without conferring an unfair advantage. We use official evaluation scripts to compute scores for the models.[7]

[2]Some models accept the number of layers/encoders as hyperparameters

[3]https://github.com/tensorflow/tpu/tree/master/models/experimental/QANet

[4]https://github.com/huggingface/pytorch-pretrained-BERT

[5]https://github.com/thunlp/OpenQA

[6]https://github.com/yicheng-w/CommonSenseMultiHopQA

[7]We used SQuAD's scripts for HotpotQA and SearchQA

Models	SQuAD		HotpotQA		SearchQA		MSMARCO	
	EM	F1	EM	F1	EM	F1	Rouge-L	Bleu1
QANet	71.08	80.33	**49.78**	**63.73**	57.33	64.06	33.23	27.90
BERT	**81.25**	**88.45**	44.22	56.84	**62.36**	**68.18**	**42.99**	33.00
CSM	57.90	69.49	48.09	50.60	54.03	60.03	39.25	**38.57**
DS-QA	60.24	70.95	35.83	45.99	60.31	65.89	23.42	9.00

Table 2: Results from all experiments

We evaluate models on every dataset's dev set, sample 100 question-answer pairs to characterize the linguistic phenomena and inference type needed to answer correctly, and then inspect performance on the sampled pairs. Definitions and examples of each inference type can be found in supplementary. We perform a further manual evaluation on 100 sampled cases where the prediction is completely incorrect for each dataset-model pair. A single annotator evaluated the samples for each dataset, although we performed limited cross-validation to promote consistency. We characterize errors and relative strengths and weaknesses for models in Section 6.

In addition to manual error analysis, we perform regressions to evaluate model performance on an entire dev set. This enables us to evaluate many course-grained hypotheses, such as the assertion that models perform worse on longer contexts. We performed logistic regression for dataset and model pairs on the EM metric (feature templates and regression tables are provided in supplementary). Although OLS regression on a continuous variable may seem like a more intuitive choice, the F1 score distributions are bimodal and heteroskedastic, which violate key OLS assumptions. We perform stepwise regression using AIC to select features and apply a Bonferroni correction to p-values based on the number of features we originally collected. We do not report regression results for MSMARCO because complete separation occurs for two features (discussed further in Section 6).

5.1 Data Preprocessing

Here we describe preprocessing decisions and experimental adaptations for the datasets.

SQuAD's contexts are relatively small, so no substantial preprocessing was done. We disabled the paragraph selector in DS-QA since each context is a single paragraph.

HotpotQA contains questions with *yes* / *no* answers, and we prepend these tokens to the context spans so extractive models can select them. We also exclude supporting evidence annotations because the models do not support these outputs. For QANet and CSM, we concatenate all paragraphs as context. For BERT, we follow Nogueira et al. (2018) and Buck et al. (2017) by concatenating contexts and using a sliding window approach, because of the models' limits on input length. During training we reduce context size to 5 paragraphs by randomly discarding non-relevant segments, so BERT is more likely to see relevant spans in one window.

SearchQA We concatenate the first 10 passages and discard the remainder for the training and dev sets for all models except DS-QA. For BERT, we follow the same sliding window approach as HotpotQA.

MSMARCO is an order of magnitude larger than the other datasets and since our primary interest is in exploring model performance, we randomly sample 20% of the training and dev QA pairs. We also remove all unanswerable questions, resulting in 101K training samples and 11K for dev. The QANet, BERT, and DS-QA model require answers to be extracted spans for training, so for each QA pair, we locate the span in the answer-bearing document with the highest Rouge score compared to the true answer and use the corresponding start and end indices for training. We also append *yes* and *no* tokens to the context so these answers are available to the extractive models. For QANet, BERT and CSM, we concatenate all snippets as context.

6 Results and Error Analysis

The evaluation scores across all models and datasets are shown in Table 2. In the remainder of this section, we examine model performance on a per-dataset basis and explore possible reasons that explain the results. For each dataset, we break down performance by the types of inference required to answer the question. We also introduce categories for common errors observed across all datasets below; Table 3 shows examples for every

Error Type	Question	Answer	Prediction
Random Guess	How high do plague fevers run?	38-41C	near 100%
Same Entity Type	What team lost Super Bowl XXXIII?	Atlanta Falcons	Denver
Sentence Selection	What did Marlee Matlin translate?	the national anthem	American Sign Language
Copying From Question	What was Apple Talk?	proprietary suite of networking protocols	AppleTalk
Factually Correct	How long are car loans typically?	60-month	5 years
Reasonable Answer	What did Edison offer Tesla ...	$10 a week raise	payment
Multihop Inference	How long is the river for which Frenchmans Creek is a Tributary?	2844 km	729 km
Span Selection	Which "Roseanne" star is in Scream 2?	Laurie Metcalf	Rebecca Gayheart
Confused By Question	What type of word play does "What Are Little Girls Made Of?" and "What Are Little Boys Made "Of" have in common?	ryhme	rock
Entity Choice	Which band has released more albums with their original members, Sick Puppies or Third Eye Blind?	Sick Puppies	Third Eye Blind
Yes/No Choice	Are Uber Goober and American Jobs both documentaries about gaming?	No	Yes
Numeric Inference	Which genus is native to more continents, Nothoscordum or Callirhoe ?	Nothoscordum	Callirhoe
Answer Missing	jan 20 , 2009 man lose 400,000 year plus 50 grand expenses federal ...	george w bush	willie pearl russell

Table 3: Examples of frequent error types from all 4 datasets

error type. We refer readers to supplementary for dataset specific examples of these error categories.

Random Guess: The answer appeared randomly selected, with no clear logic behind the choice.

Same Entity Confusion: The model selected the right type of entity (e.g., a person) but chose the wrong span.

Sentence Selection: The model predicted a span from an irrelevant sentence that shared one or more words with the question.

Copying From Question: The model picked a span that appeared in the question.

Factually Correct: The model's answer is correct but does not match a reference answer.

Reasonable Answer: The prediction makes sense semantically to the question but is not exactly correct.

Multihop Inference: In a "bridge" type question, the model's answer was only informed by one of the supporting facts. Typically the selected span answers part of the question but fails to address an additional clue or constraint.

Span Selection: The model located the answer-bearing sentence but chose the wrong span. These errors frequently happen when the correct answer is a date or number and the model chooses a nearby number instead.

Confused by Question: The question is malformed or the true answer is illogical, causing the model to choose a loosely related or random span.

Entity Choice: The question provided a choice of two entities and the model picked the wrong one.

Yes/No Choice: The question required a Yes/No response and the model picked the incorrect one.

Numeric Inference: The question required the model to choose between two numeric quantities, such as which is greater or came first. The models largely appear to guess at these questions, because none of them are designed to perform such evaluations.

Answer Missing: The answer span does not appear in the context, therefore making it impossible for the model to locate the answer.

Overall, we observe that BERT achieves the highest performance on extractive datasets with relatively straightforward questions (SQuAD and SearchQA). BERT's extensive pretraining as a language model and sentence predictor probably confers a strong advantage in these settings. QANet performs best on HotpotQA: it can process longer contexts than BERT, and our error analysis finds that QANet handles questions that require multihop inference better than the other models. BERT achieves the highest Rouge-L score on MSMARCO, but CSM has the highest Bleu1 score. This is somewhat unexpected because MSMARCO answers are often not contiguous spans, which would seem to favor CSM as the only model that generates answer sequences. We discuss these findings in more detail below.

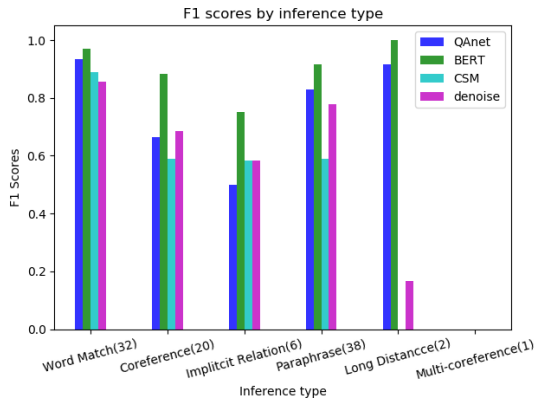

Figure 1: Comparison of Model Performance on SQuAD By Question Inference Types (numbers beside the labels indicate how many samples out of 100 fall into that category)

Error Type	**QANet**	**BERT**	**CSM**	**DS-QA**
Random Guess	28%	16%	26%	35%
Same Entity Type	30%	34%	24%	39%
Sent. Selection	20%	22%	10%	7%
Copy From Ques.	4%	0%	10%	2%
Factually Correct.	7%	11%	3%	5%
Reasonable Ans.	5%	8%	6%	3%
Regression Feature	**QANet**	**BERT**	**CSM**	**DS-QA**
Q-A Jaccard	22.3	15.0	16.6	23.0
"Who" Q	2.58	2.83	2.49	2.07
"When" Q	3.70	4.05	2.92	2.93
"How Many" Q	3.04	2.78	3.30	2.58

Table 4: Common Types of Errors on SQuAD (top) and Select SQuAD Regression Features and Odds Ratios (bottom)

6.1 SQuAD

Figure 1 compares results by inference type on SQuAD. All models did well on questions that require simple word match and BERT's advantage is less obvious. BERT is less affected by challenging inference types such as coreference and implicit relation, resulting in a large lead over other models.

Table 4 shows the error distribution for all models. The numbers in each column may not sum to 100% because multiple categories may apply to a single QA pair and we do not include error types that rarely occur.

We find that BERT is relatively precise at locating answer spans: it makes the fewest random guesses, and its most common mistake is confusing a similar entity with the answer. QANet is prone to the same error type; however, because this kind of mistake is relatively subtle, it may also be an indicator of stronger performance.

We note that 10% of the CSM model's errors are the result of selecting words that appear in the question, which is much more frequent than other models. We hypothesize that the model's copying mechanism assigns a higher probability to question keywords that appear frequently in the context, making these words more likely to appear during generation. Given that other models do not have the score aggregation step, they are less susceptible to copying words from the question.

Here we describe the features used for regression analysis and some details of how we compute them.

Lengths: The number of tokens in the question and answer respectively.

Word Match: Binary feature indicating if the sentence that has most words overlap with question contains the answer.

Question-Answer: The Jaccard similarity between the question and the answer bearing sentence. All tokens in the question and the context sentence are lemmatized using Spacy[8].

Question-Sentence: The number of overlapping words between question and answer bearing sentence.

Avg Word Match: We first segment the context into sentences and compute the average number of overlapping words between the question and sentences.

Question Types: Dummy variables signifying if a question keyword appears anywhere in the question.

Entity Counts - Question: We use Spacy to annotate entities in the question and count the number of entities.

Pronouns (Passage): We count the number of pronouns in the context from Spacy annotation.

Regression analysis shows that the Jaccard similarity between the question and answer-bearing sentence is highly predictive of EM score for all models: an increase in Jaccard similarity of 0.1 correlates with at least a 30% increase of a model answering correctly (Table 4. Questions asking *who*, *when* and *how many* are easier to answer for all models (the chances of a correct answer increase by 2-4 times). The effects are particularly strong for "when" and "how many," because the answers are numeric and distinctive from other tokens in the context. Complete regression results, including p-values, are given in supplemental (re-

[8] https://spacy.io/

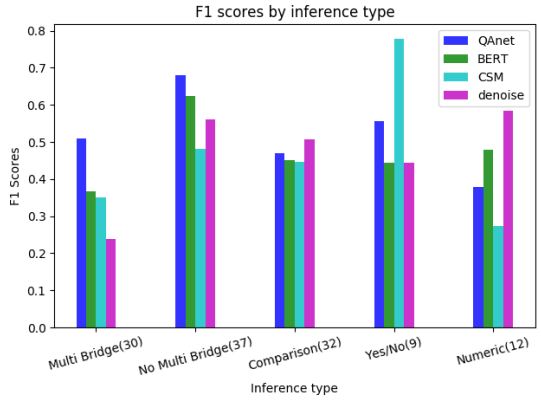

Figure 2: Comparison of Model Performance on HotpotQA By Question Inference Types

Error Type	**QANet**	**BERT**	**CSM**	**DS-QA**
Multihop Inference	13%	8%	12%	35%
Sent. Selection	12%	18%	29%	34%
Span Selection	33%	22%	19%	7%
Confused By Ques.	9%	14%	15%	7%
Factually Correct	13%	12%	7%	5%
Entity Choice	10%	16%	11%	9%
Yes/No Choice	10%	9%	5%	4%
Numeric Inference	8%	2%	8%	6%
Regression Feature	**QANet**	**BERT**	**CSM**	**DS-QA**
Ans Len	.956	.954	.962	.966
Fact Dist	.992	.992	.993	-
Context Len	-	-	-	-
Question Type	-	-	-	-

Table 5: Common Types of Errors on HotpotQA (top) and Select HotpotQA Regression Features and Odds Ratios (bottom, - denotes insignificant results)

sults in Table 4 are all significant).

6.2 HotpotQA

QANet unexpectedly recorded a higher score than BERT, a departure from the other extractive datasets. QANet is the only model with CNN layers, which may be suited to identifying related text in long contexts, necessary for multihop inference. As shown in Table 5, the most frequent errors in HotpotQA involve distractor sentences and multihop inference. QANet and BERT clearly make these errors less frequently than the other models. We attribute this to the models' more extensive attention mechanisms that better model interactions and dependencies in the context.

Nearly 25% of QANet and BERT errors are due to problems with the question or answer. This is almost certainly due to the complexity of HotpotQA questions, which increases the chances of crowdworkers erroneously formulating the question and answer. As a result, the true performance for QANet and BERT may be well over 10% higher than the actual evaluation scores; this is an issue we observe in MSMARCO as well.

Many HotpotQA questions do not require multihop inference. The question often contains a keyword or phrase that occurs only near the correct answer, or the question asks for an entity type that appears once in the context. During the manual evaluation, this was the only question type that all four models could frequently answer without error. We *only* assigned the multihop inference label to a QA pair if the correct answer could not be deduced from reading a single passage in the context. Here are some of the regression features we used besides ones that are identical to those in SQuAD.

Dist between Sup. Facts The number of tokens (in hundreds) between the starting point of each paragraph that contains a supporting fact. This is computed after concatenating the paragraphs into a single context.

Question-Answer Overlap: The number of tokens common to the question and the answer-bearing sentence.

Distractor Sentences: The number of sentences with at least the same amount of overlap as the question and answer-bearing sentence.

Yes/No: Dummy variable set to 1 if the question requires a yes or no answer.

Comparison: Dummy variable set to 1 if the answer is a selection between 2 entities.

Numeric: Dummy variable set to 1 if the answer is a number.

Regression analysis indicates that question type (e.g., "who" or "when") has insignificant predictive power, which is unusual. This is probably because knowledge of the answer's entity type does not help narrow candidate spans when questions truly require multihop inference. We also find that context length has no significant predictive power, and we even exclude it from the final regression because it worsens fit. HotpotQA is notable in that its contexts are long compared to other datasets, and this result indicates that HotpotQA's difficulty is not simply the result of long contexts. There is one case where context size matters, which is the distance in tokens between passages with supporting facts. For three of the models, an increase of 100 tokens between supporting facts correlates with approximately a halved probability of a correct answer. There is also a negative correlation for all models with answer span length. We find

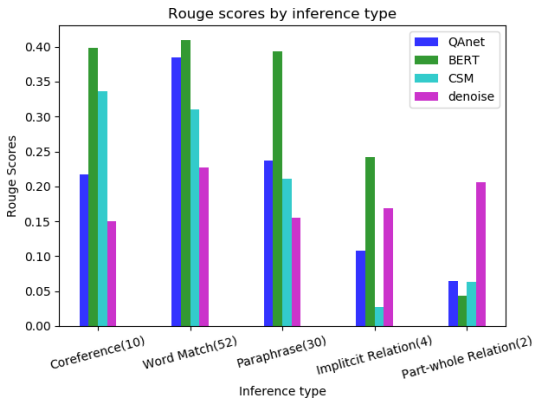

Figure 3: Comparison of Model Performance on MSMARCO By Question Inference Types

Error Type	QANet	BERT	CSM	DS-QA
Random Guess	42%	14%	26%	48%
Same Entity Type	10%	18%	23%	25%
Sent. Selection	9%	15%	16%	6%
Factually Correct	14%	40%	12%	11%
Reasonable Ans.	17%	11%	11%	4%
Yes/No Choice	8%	11%	4%	0%

Table 6: Common Types of Errors on MSMARCO

that long answers are more likely to be faulty or badly chosen. More than half of the dev set answers that contain least 10 words are improperly chosen or contain spurious information, making it very unlikely for a model to choose the exact span.

6.3 MSMARCO

Figure 3 compares Rouge scores across question inference types for MSMARCO. We primarily focus on the first three inference types since there are relatively more samples. Although QANet's performance is comparable to BERT on word match, BERT is better on questions involving coreference resolution or paraphrasing. We again attribute this to BERT's pre-training, which we suspect makes it more robust to variations in language. The error types we observed in MSMARCO are identical to those in previous sections.

Table 6 shows the distribution of common errors on MSMARCO. Similar to SQuAD, BERT is least likely to guess randomly. To our surprise, 40% of BERT's predictions that are scored as 0 are correct, and another 11% are at least reasonable. This indicates that MSMARCO's annotations are noisy and that model performance may be systematically understated. In practical terms, however, MSMARCO's questions are based on real user queries, many of which are open-ended and have too many correct answers to exhaustively list. It is worth mentioning that the reason the DS-QA model makes no yes/no choice errors is because it failed to identify the correct answer type and instead outputs random spans. Essentially, higher errors in the yes/no category at least indicate that a model can detect a yes/no question and provide an applicable answer, even if it is incorrect.

We do not report regression results for MSMARCO. The Rouge and Bleu scores are continuous but cannot be well-modeled by OLS for the same reason as F1 scores on the other datasets (see Section 5). Logistic regression is non-ideal because the scores must be coerced to either 0 or 1, and in any case, complete separation occurs because two variables trivially predict whether a question can be perfectly answered. For the CSM model, any question with an answer longer than approximately 50 words is never perfectly answered. For the remaining models, if no contiguous span from the context matches the true answer, the question is never perfectly answered.

6.4 SearchQA

As SearchQA is built by collecting documents from a search query, and aggressive preprocessing has been performed to remove common words, the inference types used for other datasets do not hold. However, each search query may have one or more clues pointing to the answer. Figure 4 shows model performance by the number of clues in a query. Model performance generally improves with more clues, and we observe that a higher number of clues correlates with more answer mentions in the provided documents.

From Table 7, we see that the Same Entity Type is the major error across all models. All the models have a similar number of Same Entity Type errors. For the Random Guess error, we see that QANet, BERT and DS-QA have similar error distributions; however, CSM has a high random error rate. This could be attributed to the decoding layer copying something useless from the context when it is unsure. Similarly, a high number of word match distractions were expected for DS-QA as its initial paragraph selector has a simple architecture and is expected to be distracted by lexical matches. Another thing to notice is that the last three error types (Factually Correct, Reasonable Answer and Answer Missing) make up between 14-24% of the errors across the models. This suggests that the

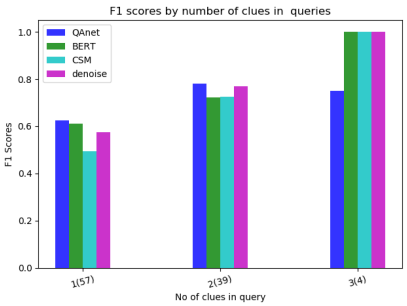

Figure 4: Comparison of Performance on SearchQA By Number of clues in Question

Error Type	QANet	BERT	CSM	DS-QA
Random Guess	19%	16%	28%	18%
Same Entity Type	30%	29%	32%	37%
Sent. Selection	20%	22%	19%	24%
Factually Correct	8%	10%	7%	6%
Reasonable Ans.	6%	7%	6%	4%
Answer Missing	5%	7%	5%	4%
Regression Feature	**QANet**	**BERT**	**CSM**	**DS-QA**
Ans Len	1.34	1.27	1.16	-
Q Len	1.03	1.03	1.02	1.03
Context Len	1.02	1.03	1.02	1.03
Any Entity Type	$\geq$1.29	$\geq$1.30	$\geq$1.45	$\geq$1.50

Table 7: Common Types of Errors on SearchQA (top) and Select SearchQA Regression Features and Odds Ratios (bottom, - denotes insignificant results)

actual model performance is better than what is portrayed in Table 2. Regarding regression analysis, we describe only features that are new for SearchQA:

Passage (Avg): Average number of tokens in all passages in the context.

Answer-Bearing Passages: The number of passages in the context that contain the correct answer.

Answer Mentions: The number of times the correct answer appears in the context.

Answer Entity Type: Dummy variable signifying the entity type of the correct answer.

Based on the regressions done on model scores (Table 7), an interesting common trend is suggested across all models. Whenever the answer is an entity[9], the odds that the models get the answer right increases significantly, frequently by a factor of 2 or 3. Although somewhat counterintuitive, the lengths of the question, answer, and context all correlate positively with the odds of selecting the right answer. We attribute this to the terse language of SearchQA, as longer questions and answers often include useful clues to narrow the list of possible answers. We further speculate that large contexts may have lengthy sections of irrelevant text that are easier to exclude during answer selection.

7 Conclusion

We conclude our discussion by presenting suggestions for good future practices when building and presenting new models and datasets. We constructively offer these points and have no intent to criticize authors whose prior work we reference.

Diverse Selection of Datasets. QA models are frequently evaluated on a single dataset, and even when multiple datasets are used, they tend to be similar. We encourage future authors to evaluate performance against a dataset with substantial differences from the one used for initial evaluation. For datasets like SQuAD, where the leaderboard is crowded with high-performing models, results on an additional challenge may provide better information on an approach's strengths and limits.

Limited Dataset Annotation. To assist in characterizing model performance, future datasets could include a small set of QA pairs that have been manually annotated with data on inference types or linguistic phenomena being tested. This information would provide a much more detailed view of model performance than a raw score, and could be incorporated into the evaluation script for an automatic presentation.

Question-Answer Quality Control. Model performance is consistently underestimated because correct answers are scored as wrong, and some questions are unanswerable because of human error. Crowdsourced datasets could include an additional task where a separate pool of workers checks QA pairs for mistakes or adds additional accepted answers to the QA pair. Standardization of answers, such as whether to include "the" before an entity, would also make scoring more precise.

[9]Detected using Google Natural language API

References

Petr Baudis and Jan Sedivý. 2015. Modeling of the question answering task in the yodaqa system. In *CLEF*.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Andrea Gesmundo, Neil Houlsby, Wojciech Gajewski, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *CoRR*, abs/1705.07830.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, Austin, Texas. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904*.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *CoRR*, abs/1704.05179.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

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*, Vancouver, Canada. Association for Computational Linguistics.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Tomas Kocisky, 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.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745, Melbourne, Australia. Association for Computational Linguistics.

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.

Rodrigo Nogueira, Jannis Bulian, and Massimiliano Ciaramita. 2018. Learning to coordinate multiple reinforcement learning agents for diverse query reformulation. *CoRR*, abs/1809.10658.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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, Austin, Texas. Association for Computational Linguistics.

Marc-Antoine Rondeau and T. J. Hazen. 2018. Systematic error analysis of the Stanford question answering dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 12–20, Melbourne, Australia. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

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

Robert Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*.

Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. 2017. Evaluation metrics for machine reading comprehension: Prerequisite skills and readability. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 806–817, Vancouver, Canada. Association for Computational Linguistics.

Alon Talmor and Jonathan Berant. 2019. MultiQA: An empirical investigation of generalization and transfer in reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4911–4921, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Soumya Wadhwa, Khyathi Chandu, and Eric Nyberg. 2018. Comparative analysis of neural QA models on SQuAD. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 89–97, Melbourne, Australia. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Thang Luong, Rui Zhao, Kai Chen, and Quoc Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension.

A Inference Types

A.1 SQuAD

Word Match: The model can simply match keys words in the question to find the answer bearing sentence and select the correct span.
Coreference: The model need to resolve a pronoun in the answer bearing sentence to find the answer.
Implicit Relation: Key entities in the context share a relationship that is not explicitly stated in the question. The model must infer the relationship to select the answer.
Paraphrase: The question paraphrases the answer bearing sentence.
Long Distance: Evidence for the answer is separated by a long sequence of irrelevant words.
Multi-coreference: The model needs to infer that one pronoun is referring to multiple entities.
Table 8 shows an example for each inference type.

A.2 HotpotQA

Multi Bridge: The model must perform multihop inference by finding and evaluating both supporting facts in the context. Each supporting fact is linked by a common "bridge" entity.
No Multi Bridge: Context clues alone can identify the answer. No multihop inference required.
Comparison: The question compares two entities, and the model must select the correct one.
Yes/No: The model must choose between a yes or no answer.
Numeric: The model must compare numeric quantities to choose the answer.

A.3 MSMARCO

There is only one new category in MSMARCO:
Part-whole Relation The model would need to infer that one entity is an example or a subset of another entity and leverage inherited properties to answer the question. An example would be:
Question: *cannot uninstall windirstat*
Gold Context: *Windows Add/ Remove Programs offers users a way to uninstall the program ... Click Start menu and run Control Panel ...*
Answer: *Click Start menu and run Control Panel...*
The model would have to understand that windirstat is a program to make correct prediction.