



# QuAC

## Question Answering in Context

### What is QuAC?

---

Question Answering in Context is a dataset for modeling, understanding, and participating in information seeking dialog. Data instances consist of an interactive dialog between two crowd workers: (1) a *student* who poses a sequence of freeform questions to learn as much as possible about a hidden Wikipedia text, and (2) a *teacher* who answers the questions by providing short excerpts (spans) from the text. QuAC introduces challenges not found in existing machine comprehension datasets: its questions are often more open-ended, unanswerable, or only meaningful within the dialog context.

[QuAC paper \(https://arxiv.org/abs/1808.07036\)](https://arxiv.org/abs/1808.07036)

[QuAC poster \(/quac\\_poster\\_pdf.pdf\)](/quac_poster_pdf.pdf)

QuAC is meant to be an academic resource and has significant limitations. Please read our detailed datasheet before considering it for any practical application.

[Datasheet \(/datasheet.pdf\)](/datasheet.pdf)

### Is QuAC exactly like SQuAD 2.0?

---

No, QuAC shares many principles with SQuAD 2.0 (<https://rajpurkar.github.io/SQuAD-explorer/>) such as span based evaluation and unanswerable questions (including website design principles! Big thanks for sharing the code!) but incorporates a new dialog component. We expect models can be easily evaluated on both resources and have tried to make our evaluation protocol as similar as possible to their own.

### Getting Started

---

Download a copy of the dataset (distributed under the CC BY-SA 4.0 (<http://creativecommons.org/licenses/by-sa/4.0/legalcode>) license):

#### Training Set

([https://s3.amazonaws.com/my89public/quac/train\\_v0.2.json](https://s3.amazonaws.com/my89public/quac/train_v0.2.json))

#### Val Set

([https://s3.amazonaws.com/my89public/quac/val\\_v0.2.json](https://s3.amazonaws.com/my89public/quac/val_v0.2.json))

To evaluate your models, we have also made available the evaluation script we will use for official evaluation, along with a sample prediction file that the script will take as input. To run the evaluation, use

```
python scorer.py --val_file <path_to_val> --model_output <path_to_predictions> --o eval.json; .
```

#### Evaluation Script

(<https://s3.amazonaws.com/my89public/quac/scorer.py>)

#### Sample Prediction File (on Dev)

(<https://s3.amazonaws.com/my89public/quac/example.json>)

Once you have built a model that works to your expectations on the dev set, you submit it to get official scores on the dev and a hidden test set. To preserve the integrity of test results, we do not release the test set to the public. Instead, we require you to submit your model so that we can run it on the test set for you. The submission process is very similar to SQuAD 2.0 (Live!):

#### Submission Tutorial

(<https://worksheets.codalab.org/worksheets/0x6c09e167a1884d359b171e13b80b95>)

## Baseline Models

---

All baseline models are available through AllenNLP. Specifically, model is here ([https://github.com/allenai/allennlp/blob/master/allennlp/models/reading\\_comprehension/dialog\\_qa.py](https://github.com/allenai/allennlp/blob/master/allennlp/models/reading_comprehension/dialog_qa.py)) and the configuration is here.

([https://github.com/allenai/allennlp/blob/master/training\\_config/dialog\\_qa.jsonnet](https://github.com/allenai/allennlp/blob/master/training_config/dialog_qa.jsonnet))

#### AllenNLP Model

(<https://github.com/allenai/allennlp>)

## How do I get the duck in my paper?

---

First, download the duck

#### The Duck

(<https://s3.amazonaws.com/my89public/quac/daffychand.pdf>)

Then, put this macro in your latex: `\newcommand{\daffychand}[0]`

`{\includegraphics [width=.04\textwidth] {path_to_daffychand/daffychand.pdf}}`

Finally, enjoy the command `\daffychand` in your paper!

## Have Questions?


---

Ask us questions at our google group ([https://groups.google.com/forum/#!forum/quac\\_ai](https://groups.google.com/forum/#!forum/quac_ai)) or at [eunsol@cs.washington.edu](mailto:eunsol@cs.washington.edu) (<mailto:eunsol@cs.washington.edu>) [hehe@stanford.edu](mailto:hehe@stanford.edu) (<mailto:hehe@stanford.edu>)

[miyyer@cs.umass.edu](mailto:miyyer@cs.umass.edu) (<mailto:mohiti@cs.umass.edu>) [marky@allenai.org](mailto:marky@allenai.org) (<mailto:marky@allenai.org>)

# Leaderboard

There can be only one duck.

Rank	Model	F1	HEQQ	HEQD
	Human Performance (Choi et al. EMNLP '18) ( <a href="https://arxiv.org/pdf/1808.07036.pdf">https://arxiv.org/pdf/1808.07036.pdf</a> )	81.1	100	100
 Mar 14, 2019	ConvBERT (single model) <i>Joint Laboratory of HIT and iFLYTEK Research</i>	<b>68.0</b>	<b>63.5</b>	<b>9.1</b>
2 Mar 14, 2019	Bert-FlowDelta (single model) <i>Anonymous</i>	66.1	61.0	7.4
3 Mar 7, 2019	BERT w/ 2-context (single model) <i>NTT Media Intelligence Labs</i>	64.9	60.2	6.1
4 Feb 21, 2019	GraphFlow (single model) <i>Anonymous</i>	64.9	60.3	5.1
5 Sep 26, 2018	FlowQA (single model) <i>Allen Institute of AI</i> <a href="https://arxiv.org/abs/1810.06683">https://arxiv.org/abs/1810.06683</a> ( <a href="https://arxiv.org/abs/1810.06683">https://arxiv.org/abs/1810.06683</a> )	64.1	59.6	5.8
6 Jan 30, 2019	BERT + History Answer Embedding (single model) <i>Anonymous</i>	62.4	57.8	5.1
7 Aug 20, 2018	BiDAF++ w/ 2-Context (single model) <i>baseline</i>	60.1	54.8	4.0
8 Aug 20, 2018	BiDAF++ (single model) <i>baseline</i>	50.2	43.3	2.2