DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs

Dheeru Dua[♣], **Yizhong Wang**[♢*], **Pradeep Dasigi**[♡], **Gabriel Stanovsky**[♡+], **Sameer Singh**[♣], and **Matt Gardner**[♠]
[♣]University of California, Irvine, USA
[♢]Peking University, Beijing, China
[♡]Allen Institute for Artificial Intelligence, Seattle, Washington, USA
[♠]Allen Institute for Artificial Intelligence, Irvine, California, USA
[+]University of Washington, Seattle, Washington, USA
ddua@uci.edu

Abstract

Reading comprehension has recently seen rapid progress, with systems matching humans on the most popular datasets for the task. However, a large body of work has highlighted the brittleness of these systems, showing that there is much work left to be done. We introduce a new English reading comprehension benchmark, DROP, which requires **D**iscrete **R**easoning **O**ver the content of **P**aragraphs. In this crowdsourced, adversarially-created, 96k-question benchmark, a system must resolve references in a question, perhaps to multiple input positions, and perform discrete operations over them (such as addition, counting, or sorting). These operations require a much more comprehensive understanding of the content of paragraphs than what was necessary for prior datasets. We apply state-of-the-art methods from both the reading comprehension and semantic parsing literatures on this dataset and show that the best systems only achieve 32.7% F_1 on our generalized accuracy metric, while expert human performance is 96.4%. We additionally present a new model that combines reading comprehension methods with simple numerical reasoning to achieve 47.0% F_1.

1 Introduction

The task of *reading comprehension*, where systems must understand a single passage of text well enough to answer arbitrary questions about it, has seen significant progress in the last few years, so much that the most popular datasets available for this task have been solved (Chen et al., 2016; Devlin et al., 2019). We introduce a substantially more challenging English reading comprehension dataset aimed at pushing the field towards more comprehensive analysis of paragraphs of text. In this new benchmark, which we call DROP, a system is given a paragraph and a question and must perform some kind of **D**iscrete **R**easoning **O**ver the text in the **P**aragraph to obtain the correct answer.

These questions that require discrete reasoning (such as addition, sorting, or counting; see Table 1) are inspired by the complex, compositional questions commonly found in the semantic parsing literature. We focus on this type of questions because they force a structured analysis of the content of the paragraph that is detailed enough to permit reasoning. Our goal is to further *paragraph understanding*; complex questions allow us to test a system's understanding of the paragraph's semantics.

DROP is also designed to further research on methods that combine distributed representations with symbolic, discrete reasoning. In order to do well on this dataset, a system must be able to find multiple occurrences of an event described in a question (presumably using some kind of soft matching), extract arguments from the events, then perform a numerical operation such as a sort, to answer a question like *"Who threw the longest touchdown pass?"*.

We constructed this dataset through crowdsourcing, first collecting passages from Wikipedia that are easy to ask hard questions about, then encouraging crowd workers to produce challenging questions. This encouragement was partially through instructions given to workers, and partially through the use of an *adversarial baseline*: we ran a baseline reading comprehension method (BiDAF) (Seo et al., 2017) in the background as crowd workers were writing questions, requiring them to give questions that the baseline system could not correctly answer. This resulted in a dataset of 96,567 questions from a variety of categories in Wikipedia, with a particular emphasis on sports game summaries and history passages. The answers to the questions are required to be spans in the passage or

[*]Work done as an intern at the Allen Institute for Artificial Intelligence in Irvine, California.

question, numbers, or dates, which allows for easy and accurate evaluation metrics.

We present an analysis of the resulting dataset to show what phenomena are present. We find that many questions combine complex question semantics with SQuAD-style argument finding; e.g., in the first question in Table 1, BiDAF correctly finds the amount the painting sold for, but does not understand the question semantics and cannot perform the numerical reasoning required to answer the question. Other questions, such as the fifth question in Table 1, require finding all events in the passage that match a description in the question, then aggregating them somehow (in this instance, by counting them and then performing an argmax). Very often entity coreference is required. Table 1 gives a number of different phenomena, with their proportions in the dataset.

We used three types of systems to judge baseline performance on DROP: (1) heuristic baselines, to check for biases in the data; (2) SQuAD-style reading comprehension methods; and (3) semantic parsers operating on a pipelined analysis of the passage. The reading comprehension methods perform the best, with our best baseline achieving 32.7% F_1 on our generalized accuracy metric, while expert human performance is 96.4%. Finally, we contribute a new model for this task that combines limited numerical reasoning with standard reading comprehension methods, allowing the model to answer questions involving counting, addition and subtraction. This model reaches 47% F_1, a 14.3% absolute increase over the best baseline system.

The dataset, code for the baseline systems, and a leaderboard with a hidden test set can be found at `https://allennlp.org/drop`.

2 Related Work

Question answering datasets With systems reaching human performance on the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016), many follow-on tasks are currently being proposed. All of these datasets throw in additional complexities to the reading comprehension challenge, around tracking conversational state (Reddy et al., 2019; Choi et al., 2018), requiring passage retrieval (Joshi et al., 2017; Yang et al., 2018; Talmor and Berant, 2018), mismatched passages and questions (Saha et al., 2018; Kociský et al., 2018; Rajpurkar et al., 2018), integrating knowledge from external sources (Mihaylov et al., 2018; Zhang et al., 2019), tracking entity state changes (Mishra et al., 2018; Ostermann et al., 2018) or a particular kind of "multi-step" reasoning over multiple documents (Welbl et al., 2018; Khashabi et al., 2018). Similar facets are explored in medical domain datasets (Pampari et al., 2018; Šuster and Daelemans, 2018) which contain automatically generated queries on medical records based on predefined templates. We applaud these efforts, which offer good avenues to study these additional phenomena. However, we are concerned with *paragraph understanding*, which on its own is far from solved, so DROP has none of these additional complexities. It consists of single passages of text paired with independent questions, with only linguistic facility required to answer the questions.[1] One could argue that we are adding numerical reasoning as an "additional complexity", and this is true; however, it is only simple reasoning that is relatively well-understood in the semantic parsing literature, and we use it as a necessary means to force more comprehensive passage understanding.

Many existing algebra word problem datasets also contain similar phenomena to what is in DROP (Koncel-Kedziorski et al., 2015; Kushman et al., 2014; Hosseini et al., 2014; Clark et al., 2016; Ling et al., 2017). Our dataset is different in that it uses much longer contexts, is more open domain, and requires deeper paragraph understanding.

Semantic parsing The semantic parsing literature has a long history of trying to understand complex, compositional question semantics in terms of some grounded knowledge base or other environment (Zelle and Mooney, 1996; Zettlemoyer and Collins, 2005; Berant et al., 2013a, *inter alia*). It is this literature that we modeled our questions on, particularly looking at the questions in the WikiTableQuestions dataset (Pasupat and Liang, 2015). If we had a structured, tabular representation of the content of our paragraphs, DROP would be largely the same as WikiTableQuestions, with similar (possibly even simpler) question semantics. Our novelty is that we are the first to combine these complex questions with paragraph understanding, with the aim of encouraging systems that can produce comprehensive structural analyses of paragraphs, either explicitly or implicitly.

Adversarial dataset construction We continue

[1] Some questions in our dataset require limited sports domain knowledge to answer; we expect that there are enough such questions that systems can reasonably learn this knowledge from the data.

Reasoning	Passage (some parts shortened)	Question	Answer	BiDAF
Subtraction (28.8%)	That year, his **Untitled (1981)**, a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was **sold by Robert Lehrman for $16.3 million, well above its $12 million high estimate**.	How many more dollars was the Untitled (1981) painting sold for than the 12 million dollar estimation?	4300000	$16.3 million
Comparison (18.2%)	In **1517, the seventeen-year-old King sailed to Castile**. There, his Flemish court .... **In May 1518, Charles traveled to Barcelona in Aragon**.	Where did Charles travel to first, Castile or Barcelona?	Castile	Aragon
Selection (19.4%)	In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, **Baker University professor and playwright Don Mueller and Phyllis E. Braun, Business Manager, produced a musical play entitled The Ballad Of Black Jack** to tell the story of the events that led up to the battle.	Who was the University professor that helped produce The Ballad Of Black Jack, Ivan Boyd or Don Mueller?	Don Mueller	Baker
Addition (11.7%)	Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSK in the village of Nos Kalik, located in a pink zone near Šibenik, and captured the village at 4:45 p.m. on **2 March 1992**. The JNA formed a battlegroup to counterattack the **next day**.	What date did the JNA form a battlegroup to counterattack after the village of Nos Kalik was captured?	3 March 1992	2 March 1992
Count (16.5%) and Sort (11.7%)	Denver would retake the lead with kicker **Matt Prater nailing a 43-yard field goal**, yet Carolina answered as kicker **John Kasay ties the game with a 39-yard field goal**. ... Carolina closed out the half with **Kasay nailing a 44-yard field goal**. ... In the fourth quarter, Carolina sealed the win with **Kasay's 42-yard field goal**.	Which kicker kicked the most field goals?	John Kasay	Matt Prater
Coreference Resolution (3.7%)	**James Douglas** was the second son of Sir George Douglas of Pittendreich, and Elizabeth Douglas, daughter David Douglas of Pittendreich. Before **1543 he married Elizabeth**, daughter of James Douglas, 3rd Earl of Morton. **In 1553 James Douglas succeeded to the title and estates of his father-in-law**.	How many years after he married Elizabeth did James Douglas succeed to the title and estates of his father-in-law?	10	1553
Other Arithmetic (3.2%)	Although the movement initially gathered some **60,000 adherents**, the subsequent establishment of the Bulgarian Exarchate **reduced their number by some 75%**.	How many adherents were left after the establishment of the Bulgarian Exarchate?	15000	60,000
Set of spans (6.0%)	According to some sources 363 civilians were killed in **Kavadarci**, 230 in **Negotino** and 40 in **Vatasha**.	What were the 3 villages that people were killed in?	Kavadarci, Negotino, Vatasha	Negotino and 40 in Vatasha
Other (6.8%)	This **Annual Financial Report** is our principal financial statement of accountability. The **AFR gives a comprehensive view** of the Department's financial activities ...	What does AFR stand for?	Annual Financial Report	one of the Big Four audit firms

Table 1: Example questions and answers from the DROP dataset, showing the relevant parts of the associated passage and the reasoning required to answer the question.

a recent trend in creating datasets with adversarial baselines in the loop (Paperno et al., 2016; Minervini and Riedel, 2018; Zellers et al., 2018; Zhang et al., 2019; Zellers et al., 2019). In our case, instead of using an adversarial baseline to filter automatically generated examples, we use it in a crowdsourcing task, to teach crowd workers to avoid easy questions, raising the difficulty level of the questions they provide.

Neural symbolic reasoning DROP is designed to encourage research on methods that combine neural methods with discrete, symbolic reasoning. We present one such model in Section 6. Other related work along these lines has been done by Reed and de Freitas (2016), Neelakantan et al. (2016), and Liang et al. (2017).

3 DROP Data Collection

In this section, we describe our annotation protocol, which consists of three phases. First, we automatically extract passages from Wikipedia which are expected to be amenable to complex questions. Second, we crowdsource question-answer pairs on these passages, eliciting questions which require

discrete reasoning. Finally, we validate the development and test portions of DROP to ensure their quality and report inter-annotator agreement.

Passage extraction We searched Wikipedia for passages that had a narrative sequence of events, particularly with a high proportion of numbers, as our initial pilots indicated that these passages were the easiest to ask complex questions about. We found that National Football League (NFL) game summaries and history articles were particularly promising, and we additionally sampled from any Wikipedia passage that contained at least twenty numbers.[2] This process yielded a collection of about 7,000 passages.

Question collection We used Amazon Mechanical Turk[3] to crowdsource the collection of question-answer pairs, where each question could be answered in the context of a single Wikipedia passage. In order to allow some flexibility during the annotation process, in each human intelligence task (HIT) workers were presented with a random sample of 5 of our Wikipedia passages, and were asked to produce a total of at least 12 question-answer pairs on any of these.

We presented workers with example questions from five main categories, inspired by questions from the semantic parsing literature (addition/subtraction, minimum/maximum, counting, selection and comparison; see examples in Table 1), to elicit questions that require complex linguistic understanding and discrete reasoning. In addition, to further increase the difficulty of the questions in DROP, we employed a novel adverserial annotation setting, where workers were only allowed to submit questions which a real-time QA model BiDAF *could not* solve.[4]

Next, each worker answered their own question with one of three answer types: spans of text from either question or passage, a date (which was common in history and open-domain text) and numbers, allowed only for questions which explicitly stated a specific unit of measurement (e.g., "How many *yards* did Brady run?"), in an attempt to simplify the evaluation process.

Initially, we opened our HITs to all United States workers and gradually reduced our worker pool to workers who understood the task and annotated it well. Each HIT paid 5 USD and could be completed within 30 minutes, compensating a trained worker with an average pay of 10 USD/ hour.

Overall, we collected a total of 96,567 question-answer pairs with a total Mechanical Turk budget of 60k USD (including validation). The dataset was randomly partitioned by passage into training (80%), development (10%) and test (10%) sets, so all questions about a particular passage belong to only one of the splits.

Statistic	Train	Dev	Test
Number of passages	5565	582	588
Avg. passage len [words]	213.45	191.62	195.12
Number of questions	77,409	9,536	9,622
Avg. question len [words]	10.79	11.17	11.23
Avg. questions / passage	13.91	16.38	16.36
Question vocabulary size	29,929	8,023	8,007

Table 2: Dataset statistics across the different splits.

Validation In order to test inter-annotator agreement and to improve the quality of evaluation against DROP, we collected at least two additional answers for each question in the development and test sets.

In a separate HIT, workers were given context passages and a previously crowdsourced question, and were asked to either answer the question or mark it as invalid (this occurred for 0.7% of the data, which we subsequently filtered out). We found that the resulting inter-annotator agreement was good and on par with other QA tasks; overall Cohen's κ was 0.74, with 0.81 for numbers, 0.62 for spans, and 0.65 for dates.

4 DROP Data Analysis

In the following, we quantitatively analyze properties of passages, questions, and answers in DROP. Different statistics of the dataset are depicted in Table 2. Notably, questions have a diverse vocabulary of around 30k different words in our training set.

Question analysis To assess the question type distribution, we sampled 350 questions from the training and development sets and manually annotated the categories of discrete operations required to answer the question. Table 1 shows the distribution of these categories in the dataset. In addition, to get a better sense of the lexical diversity of questions in the dataset, we find the most frequent

[2] We used an October 2018 Wikipedia dump, as well as scraping of online Wikipedia.

[3] www.mturk.com

[4] While BiDAF is no longer state-of-the-art, performance is reasonable and the AllenNLP implementation (Gardner et al., 2017) made it the easiest to deploy as a server.

Answer Type	Percent	Example
NUMBER	66.1	12
PERSON	12.2	Jerry Porter
OTHER	9.4	males
OTHER ENTITIES	7.3	Seahawks
VERB PHRASE	3.5	Tom arrived at Acre
DATE	1.5	3 March 1992

Table 3: Distribution of answer types in training set, according to an automatic named entity recognition.

trigram patterns in the questions per answer type. We find that the dataset offers a huge variety of linguistic constructs, with the most frequent pattern ("Which team scored") appearing in only 4% of the span type questions. For number type questions, the 5 most frequent question patterns all start with "How many", indicating the need to perform counting and other arithmetic operations. A distribution of the trigrams containing the start of the questions are shown in Figure 1.

Answer analysis To discern the level of *passage understanding* needed to answer the questions in DROP, we annotate the set of spans in the passage that are necessary for answering the 350 questions mentioned above. We find that on an average 2.18 spans need to be considered to answer a question and the average distance between these spans is 26 words, with 20% of samples needing at least 3 spans (see appendix for examples). Finally, we assess the answer distribution in Table 3, by running the part-of-speech tagger and named entity recognizer from spaCy[5] to automatically partition all the answers into various categories. We find that a majority of the answers are numerical values and proper nouns.

5 Baseline Systems

In this section we describe the initial baselines that we evaluated on the DROP dataset. We used three types of baselines: state-of-the-art semantic parsers (§5.1), state-of-the-art reading comprehension models (§5.2), and heuristics looking for annotation artifacts (§5.3). We use two evaluation metrics to compare model performance: *Exact-Match*, and a numeracy-focused (macro-averaged) F_1 score, which measures overlap between a bag-of-words representation of the gold and predicted answers. We employ the same implementation of Exact-Match accuracy as used by SQuAD, which removes articles and does other simple normalization, and our F_1 score is based on that used by SQuAD. Since DROP is numeracy-focused, we define F_1 to be 0 when there is a number mismatch between the gold and predicted answers, regardless of other word overlap. When an answer has multiple spans, we first perform a one-to-one alignment greedily based on bag-of-word overlap on the set of spans and then compute average F_1 over each span. When there are multiple annotated answers, both metrics take a max over all gold answers.

5.1 Semantic Parsing

Semantic parsing has been used to translate natural language utterances into formal executable languages (e.g., SQL) that can perform discrete operations against a structured knowledge representation, such as knowledge graphs or tabular databases (Zettlemoyer and Collins, 2005; Berant et al., 2013b; Yin and Neubig, 2017; Chen and Mooney, 2011, *inter alia*). Since many of DROP's questions require similar discrete reasoning, it is appealing to port some of the successful work in semantic parsing to the DROP dataset. Specifically, we use the grammar-constrained semantic parsing model built by Krishnamurthy et al. (2017) (KDG) for the WIKITABLEQUESTIONS tabular dataset (Pasupat and Liang, 2015).

Sentence representation schemes We experimented with three paradigms to represent paragraphs as structured contexts: (1) Stanford dependencies (de Marneffe and Manning, 2008, Syn Dep); which capture word-level syntactic relations, (2) Open Information Extraction (Banko et al., 2007, Open IE), a shallow semantic representation which directly links predicates and arguments; and (3) Semantic Role Labeling (Carreras and Màrquez, 2005, SRL), which disambiguates senses for polysemous predicates and assigns predicate-specific argument roles.[6] To adhere to KDG's structured representation format, we convert each of these representations into a table, where rows are predicate-argument structures and columns correspond to different argument roles.

Logical form language Our logical form language identifies five basic elements in the table representation: *predicate-argument structures* (i.e., table rows), *relations* (column-headers), *strings*, *num-*

[5] https://spacy.io/

[6] We used the AllenNLP implementations of state-of-the-art models for all of these representations (Gardner et al., 2017; Dozat et al., 2017; He et al., 2017; Stanovsky et al., 2018).

(a) For span type answers

(b) For number type answers

Figure 1: Distribution of the most popular question prefixes for two different subsets of the training data.

bers, and *dates*. In addition, it defines functions that operate on these elements, such as counters and filters.[7] Following Krishnamurthy et al. (2017), we use the argument and return types of these functions to automatically induce a grammar to constrain the parser. We also add context-specific rules to produce strings occurring in both question and paragraph, and those paragraph strings that are neighbors of question tokens in the GloVe embedding space (Pennington et al., 2014), up to a cosine distance of d.[8] The complete set of functions used in our language and their induced grammar can be found in the code release.

Training and inference During training, the KDG parser maximizes the marginal likelihood of a set of (possibly spurious) question logical forms that evaluate to the correct answer. We obtain this set by performing an exhaustive search over the grammar up to a preset tree depth. At test time, we use beam search to produce the most likely logical form, which is then executed to predict an answer.

5.2 SQuAD-style Reading Comprehension

We test four different SQuAD-style reading comprehension models on DROP: (1) **BiDAF** (Seo et al., 2017), which is the adversarial baseline we used in data construction (66.8% EM on SQuAD 1.1); (2) **QANet** (Yu et al., 2018), currently the best-performing published model on SQuAD 1.1 without data augmentation or pretraining (72.7% EM); (3) **QANet + ELMo**, which enhances the QANet model by concatenating pretrained ELMo representations (Peters et al., 2018) to the original embeddings (78.7% EM); (4) **BERT** (Devlin et al., 2019), which recently achieved improvements on many NLP tasks with a novel pretraining technique (84.7% EM).[9]

These models require a few minor adaptations when training on DROP. While SQuAD provides answer indices in the passage, our dataset only provides the answer strings. To address this, we use the marginal likelihood objective function proposed by Clark and Gardner (2018), which sums over the probabilities of all the matching spans.[10] We also omitted the training questions which cannot be answered by a span in the passage (45%), and therefore cannot be represented by these systems.

For the BiDAF baseline, we use the implementation in AllenNLP but change it to use the marginal objective. For the QANet model, our settings differ from the original paper only in the batch size (16 v.s. 32) and number of blocks in the modeling layer

[7] For example `filter_number_greater` takes a set of predicate-argument structures, the name of a relation, and a number, and returns all those structures where the numbers in the argument specified by the relation are greater than the given number.

[8] $d = 0.3$ was manually tuned on the development set.

[9] The first three scores are based on our own implementation, while the score for BERT is based on an open-source implementation from Hugging Face: https://github.com/huggingface/pytorch-pretrained-bert

[10] For the black-box BERT model, we convert DROP to SQuAD format by using the first match as the gold span.

(6 v.s. 7) due to the GPU memory limit. We adopt the ELMo representations trained on 5.5B corpus for the QANet+ELMo baseline and the large uncased BERT model for the BERT baseline. The hyper-parameters for our NAQANet model (§6) are the same as for the QANet baseline.

5.3 Heuristic Baselines

A recent line of work (Gururangan et al., 2018; Kaushik and Lipton, 2018) has identified that popular crowdsourced NLP datasets (such as SQuAD (Rajpurkar et al., 2016) or SNLI (Bowman et al., 2015)) are prone to have artifacts and annotation biases which can be exploited by supervised algorithms that learn to pick up these artifacts as signal instead of more meaningful semantic features. We estimate artifacts by training the QANet model described in Section 5.2 on a version of DROP where either the question or the paragraph input representation vectors are zeroed out (**question-only** and **paragraph-only**, respectively). Consequently, the resulting models can then only predict answer spans from either the question or the paragraph.

In addition, we devise a baseline that estimates the answer variance in DROP. We start by counting the unigram and bigram answer frequency for each wh question-word in the train set (as the first word in the question). The **majority baseline** then predicts an answer as the set of 3 most common answer spans for the input question word (e.g., for "when", these were "quarter", "end" and "October").

6 NAQANet

DROP is designed to encourage models that combine neural reading comprehension with symbolic reasoning. None of the baselines we described in Section 5 can do this. As a preliminary attempt toward this goal, we propose a numerically-aware QANet model, NAQANet, which allows the state-of-the-art reading comprehension system to produce three new answer types: (1) spans from the question; (2) counts; (3) addition or subtraction over numbers. To predict numbers, the model first predicts whether the answer is a count or an arithmetic expression. It then predicts the specific numbers involved in the expression. This can be viewed as the neural model producing a *partially executed* logical form, leaving the final arithmetic to a symbolic system. While this model can currently only handle a very limited set of operations, we believe this is a promising approach to combining neural methods and symbolic reasoning. The model is trained by marginalizing over all execution paths that lead to the correct answer.

6.1 Model Description

Our NAQANet model follows the typical architecture of previous reading comprehension models, which is composed of embedding, encoding, passage-question attention, and output layers. We use the original QANet architecture for everything up to the output layer. This gives us a question representation $\mathbf{Q} \in \mathbb{R}^{m \times d}$, and a projected question-aware passage representation $\bar{\mathbf{P}} \in \mathbb{R}^{n \times d}$. We have four different output layers, for the four different kinds of answers the model can produce:

Passage span As in the original QANet model, to predict an answer in the passage we apply three repetitions of the QANet encoder to the passage representation $\bar{\mathbf{P}}$ and get their outputs as $\mathbf{M}_0$, $\mathbf{M}_1$, $\mathbf{M}_2$ respectively. Then the probabilities of the starting and ending positions from the passage can be computed as:

$$\mathbf{p}^{\text{p\_start}} = \text{softmax}(\text{FFN}([\mathbf{M}_0; \mathbf{M}_1]), \quad (1)$$

$$\mathbf{p}^{\text{p\_end}} = \text{softmax}(\text{FFN}([\mathbf{M}_0; \mathbf{M}_2]) \quad (2)$$

where FFN is a two-layer feed-forward network with the RELU activation.

Question span Some questions in DROP have their answer in the *question* instead of the passage. To predict an answer from the question, the model first computes a vector $\mathbf{h}^P$ that represents the information it finds in the passage:

$$\boldsymbol{\alpha}^P = \text{softmax}(\mathbf{W}^P \bar{\mathbf{P}}), \quad (3)$$

$$\mathbf{h}^P = \boldsymbol{\alpha}^P \bar{\mathbf{P}} \quad (4)$$

Then it computes the probabilities of the starting and ending positions from the question as:

$$\mathbf{p}^{\text{q\_start}} = \text{softmax}(\text{FFN}([\mathbf{Q}; \mathbf{e}^{|Q|} \otimes \mathbf{h}^P]), \quad (5)$$

$$\mathbf{p}^{\text{q\_end}} = \text{softmax}(\text{FFN}([\mathbf{Q}; \mathbf{e}^{|Q|} \otimes \mathbf{h}^P]) \quad (6)$$

where the outer product with the identity $(\mathbf{e}^{|Q|} \otimes \cdot)$ simply repeats $\mathbf{h}^P$ for each question word.

Count We model the capability of counting as a multi-class classification problem. Specifically, we consider ten numbers (0–9) in this preliminary model and the probabilities of choosing these numbers is computed based on the passage vector $\mathbf{h}^P$:

$$\mathbf{p}^{\text{count}} = \text{softmax}(\text{FFN}(\mathbf{h}^P)) \quad (7)$$

Arithmetic expression Many questions in DROP require the model to locate multiple numbers in the passage and add or subtract them to get the final answer. To model this process, we first extract all the numbers from the passage and then learn to assign a plus, minus or zero for each number. In this way, we get an arithmetic expression composed of signed numbers, which can be evaluated to give the final answer.

To do this, we first apply another QANet encoder to $\mathbf{M}_2$ and get a new passage representation $\mathbf{M}_3$. Then we select an index over the concatenation of $\mathbf{M}_0$ and $\mathbf{M}_3$, to get a representation for each number in this passage. The i^{th} number can be represented as $\mathbf{h}_i^N$ and the probabilities of this number being assigned a plus, minus or zero are:

$$\mathbf{p}_i^{\text{sign}} = \text{softmax}(\text{FFN}(\mathbf{h}_i^N)) \quad (8)$$

Answer type prediction We use a categorical variable to decide between the above four answer types, with probabilities computed as:

$$\mathbf{p}^{\text{type}} = \text{softmax}(\text{FFN}([\mathbf{h}^P, \mathbf{h}^Q])) \quad (9)$$

where $\mathbf{h}^Q$ is computed over $\mathbf{Q}$, in a similar way as we did for $\mathbf{h}^P$. At test time, we first determine this answer type greedily and then get the best answer from the selected type.

6.2 Weakly-Supervised Training

For supervision, DROP contains only the answer string, not which of the above answer types is used to arrive at the answer. To train our model, we adopt the weakly supervised training method widely used in the semantic parsing literature (Berant et al., 2013a). We find all executions that evaluate to the correct answer, including matching passage spans and question spans, correct count numbers, as well as sign assignments for numbers. Our training objective is then to maximize the marginal likelihood of these executions.[11]

7 Results and Discussion

The performance of all tested models on the DROP dataset is presented in Table 4. Most notably, all models perform significantly worse than on other prominent reading comprehension datasets, while human performance remains at similar high levels.[12] For example, BERT, the current state-of-the-art on SQuAD, *drops* by more than 50 absolute F1 points. This is a positive indication that DROP is indeed a challenging reading comprehension dataset, which opens the door for tackling new and complex reasoning problems on a large scale.

Method	Dev		Test	
	EM	F_1	EM	F_1
Heuristic Baselines				
Majority	0.09	1.38	0.07	1.44
Q-only	4.28	8.07	4.18	8.59
P-only	0.13	2.27	0.14	2.26
Semantic Parsing				
Syn Dep	9.38	11.64	8.51	10.84
OpenIE	8.80	11.31	8.53	10.77
SRL	9.28	11.72	8.98	11.45
SQuAD-style RC				
BiDAF	26.06	28.85	24.75	27.49
QANet	27.50	30.44	25.50	28.36
QANet+ELMo	27.71	30.33	27.08	29.67
BERT	30.10	33.36	29.45	32.70
NAQANet				
+ Q Span	25.94	29.17	24.98	28.18
+ Count	30.09	33.92	30.04	32.75
+ Add/Sub	43.07	45.71	40.40	42.96
Complete Model	**46.20**	**49.24**	**44.07**	**47.01**
Human	-	-	94.09	96.42

Table 4: Performance of the different models on our development and test set, in terms of Exact Match (EM), and numerically-focused F_1 (§5). Both metrics are calculated as the maximum against a set of gold answers.

The best performance is obtained by our NAQANet model. Table 6 shows that our gains are obtained on the challenging and frequent number answer type, which requires various complex types of reasoning. Future work may also try combining our model with BERT. Furthermore, we find that all heuristic baselines do poorly on our data, hopefully attesting to relatively small biases in DROP.

Difficulties of building semantic parsers We see that all the semantic parsing baselines perform quite poorly on DROP. This is mainly because of our pipeline of extracting tabular information from paragraphs, followed by the denotation-driven logical form search, can yield logical forms only for a subset of the training data. For SRL and syntactic dependency sentence representation schemes,

[11] Due to the exponential search space and the possible noise, we only search the addition/subtraction of two numbers. Given this limited search space, the search and marginalization are exact.

[12] Human performance was estimated by the authors collectively answering 560 questions from the test set, which were then evaluated using the same metric as learned systems. This is in contrast to holding out one gold annotation and evaluating it against the other annotations, as done in prior work, which underestimates human performance relative to systems.

Phenomenon	Passage Highlights	Question	Answer	Our model
Subtraction + Coreference	... Twenty-five of his 150 men were sick, and his advance stalled ...	How many of Bartolom de Amsqueta's 150 men were not sick?	125	145
Count + Filter	... Macedonians were the largest ethnic group in Skopje, with 338,358 inhabitants ... Then came ... Serbs (14,298 inhabitants), Turks (8,595), Bosniaks (7,585) and Vlachs (2,557) ...	How many ethnicities had less than 10000 people?	3	2
Domain knowledge	... Smith was sidelined by a torn pectoral muscle suffered during practice ...	How many quarters did Smith play?	0	2
Addition	... culminating in the Battle of Vienna of 1683, which marked the start of the 15-year-long Great Turkish War ...	What year did the Great Turkish War end?	1698	1668

Table 5: Representative examples from our model's error analysis. We list the identified semantic phenomenon, the relevant passage highlights, a gold question-answer pair, and the erroneous prediction by our model.

the search was able to yield logical forms for 34% of the training data, whereas with OpenIE, it was only 25%. On closer examination of a sample of 60 questions and the information extracted by the SRL scheme (the best performing of the three), we found that only 25% of the resulting tables contained information needed to the answer the questions. These observations show that high quality information extraction is a strong prerequisite for building semantic parsers for DROP. Additionally, the fact that this is a weakly supervised semantic parsing problem also makes training hard. The biggest challenge in this setup is the spuriousness of logical forms used for training, where the logical form evaluates to the correct denotation but does not actually reflect the semantics of the question. This makes it hard for the model trained on these spurious logical forms to generalize to unseen data. From the set of logical forms for a sample of 60 questions analyzed, we found that only 8 questions (13%) contained non-spurious logical forms.

Error Analysis Finally, in order to better understand the outstanding challenges in DROP, we conducted an error analysis on a random sample of 100 erroneous NAQANet predictions. The most common errors were on questions which required complex type of reasoning, such as arithmetic operations (evident in 51% of the errors), counting (30%), domain knowledge and common sense (23%), co-reference (6%), or a combination of different types of reasoning (40%). See Table 5 for examples of some of the common phenomena.

Type	(%)	Exact Match		F1	
		QN+	BERT	QN+	BERT
Date	1.57	28.7	38.7	35.5	42.8
Numbers	61.94	44.0	14.5	44.2	14.8
Single Span	31.71	58.2	64.6	64.6	70.1
>1 Spans	4.77	0	0	17.13	25.0

Table 6: Dev set performance breakdown by different answer types; our model (NAQANet, marked as *QN+*) vs. BERT, the best-performing baseline.

8 Conclusion

We have presented DROP, a dataset of complex reading comprehension questions that require **D**iscrete **R**easoning **O**ver **P**aragraphs. This dataset is substantially more challenging than existing datasets, with the best baseline achieving only 32.7% F1, while humans achieve 96%. We hope this dataset will spur research into more comprehensive analysis of paragraphs, and into methods that combine distributed representations with symbolic reasoning. We have additionally presented initial work in this direction, with a model that augments QANet with limited numerical reasoning capability, achieving 47% F1 on DROP.

Acknowledgments

We would like to thank Noah Smith, Yoav Goldberg, and Jonathan Berant for insightful discussions that informed the direction of this work. The computations on beaker.org were supported in part by credits from Google Cloud.

References

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew G Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013a. Semantic parsing on freebase from question-answer pairs. In *EMNLP*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013b. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of CONLL*, pages 152–164.

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

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *AAAI*, volume 2, pages 1–2.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke S. Zettlemoyer. 2018. Quac: Question answering in context. In *EMNLP*.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *ACL*.

Peter Clark, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Turney, and Daniel Khashabi. 2016. Combining retrieval, statistics, and inference to answer elementary science questions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, abs/1810.04805.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the conll 2017 shared task. In *CoNLL Shared Task*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proc. of NAACL*.

Luheng He, Kenton Lee, Mike Lewis, and Luke S. Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *ACL*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533.

Mandar S. Joshi, Eunsol Choi, Daniel S. Weld, and Luke S. Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*.

Divyansh Kaushik and Zachary Chase Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *EMNLP*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL-HLT*.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *TACL*, 6:317–328.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *TACL*, 3:585–597.

Jayant Krishnamurthy, Pradeep Dasigi, and Matt Gardner. 2017. Neural semantic parsing with type constraints for semi-structured tables. In *EMNLP*.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 271–281.

Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. 2017. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *ACL*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *ACL*.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *CFCFPE@COLING*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural nli models to integrate logical background knowledge. In *CoNLL*.

Bhavana Dalvi Mishra, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: A challenge dataset and models for process paragraph comprehension.

Arvind Neelakantan, Quoc V. Le, and Ilya Sutskever. 2016. Neural programmer: Inducing latent programs with gradient descent. *ICLR*.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. Mcscript: a novel dataset for assessing machine comprehension using script knowledge. *LREC Proceedings, 2018*.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. *ACL*.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

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

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *TACL*.

Scott E. Reed and Nando de Freitas. 2016. Neural programmer-interpreters. *ICLR*.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. Duorc: Towards complex language understanding with paraphrased reading comprehension. In *ACL*.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *ICLR*.

Gabriel Stanovsky, Julian Michael, Luke S. Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *NAACL-HLT*.

Simon Šuster and Walter Daelemans. 2018. Clicr: a dataset of clinical case reports for machine reading comprehension.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL-HLT*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*, 6:287–302.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. In *ACL'17*.

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

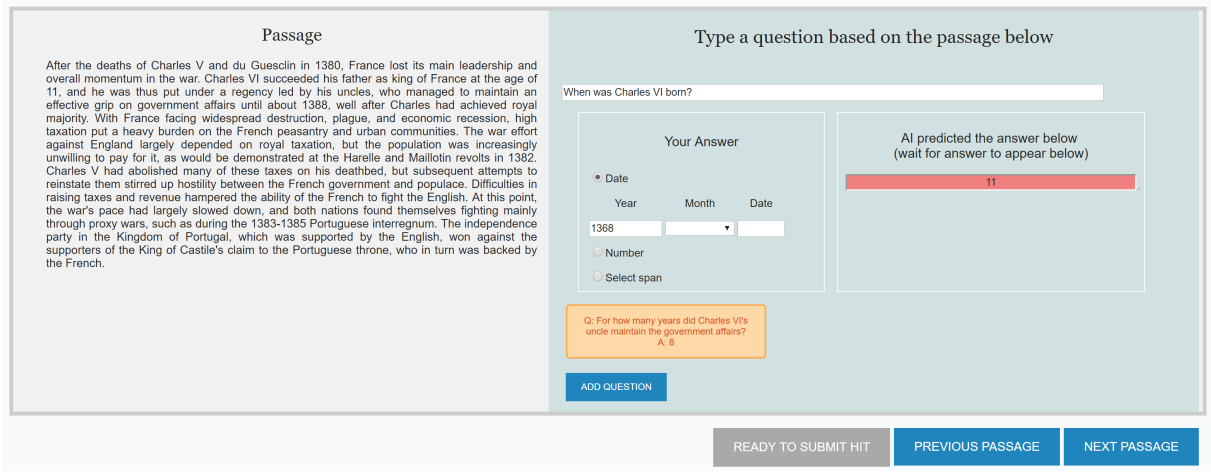
John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI, Vol. 2*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. *CVPR*, abs/1811.10830.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2019. ReCoRD: Bridging the gap between human and machine commonsense reading comprehension.



Question: Which team had the longest touchdown pass?

Still searching for their first win, the Bengals flew to Texas Stadium for a Week 5 interconference duel with the Dallas Cowboys. In the first quarter, Cincinnati trailed early as Cowboys kicker Nick Folk got a 30-yard field goal, along with RB Felix Jones getting a 33-yard TD run. In the second quarter, Dallas increased its lead as QB Tony Romo completed a 4-yard TD pass to TE Jason Witten. The Bengals would end the half with kicker Shayne Graham getting a 41-yard and a 31-yard field goal. In the third quarter, Cincinnati tried to rally as QB Carson Palmer completed an 18-yard TD pass to WR T. J. Houshmandzadeh. In the fourth quarter, the Bengals got closer as Graham got a 40-yard field goal, yet the Cowboys answered with Romo completing a 57-yard TD pass to WR Terrell Owens. Cincinnati tried to come back as Palmer completed a 10-yard TD pass to Houshmandzadeh (with a failed 2-point conversion), but Dallas pulled away with Romo completing a 15-yard TD pass to WR Patrick Crayton.

Which Bengals receiver scored two touchdowns?

Still searching for their first win, the Bengals flew to Texas Stadium for a Week 5 interconference duel with the Dallas Cowboys. In the first quarter, Cincinnati trailed early as Cowboys kicker Nick Folk got a 30-yard field goal, along with RB Felix Jones getting a 33-yard TD run. In the second quarter, Dallas increased its lead as QB Tony Romo completed a 4-yard TD pass to TE Jason Witten. The Bengals would end the half with kicker Shayne Graham getting a 41-yard and a 31-yard field goal. In the third quarter, Cincinnati tried to rally as QB Carson Palmer completed an 18-yard TD pass to WR T. J. Houshmandzadeh. In the fourth quarter, the Bengals got closer as Graham got a 40-yard field goal, yet the Cowboys answered with Romo completing a 57-yard TD pass to WR Terrell Owens. Cincinnati tried to come back as Palmer completed a 10-yard TD pass to Houshmandzadeh (with a failed 2-point conversion), but Dallas pulled away with Romo completing a 15-yard TD pass to WR Patrick Crayton.

Which alliance lost more troops to prisoner status?

About eight million men surrendered and were held in POW camps during the war. All nations pledged to follow the Hague Conventions on fair treatment of prisoners of war, and the survival rate for POWs was generally much higher than that of combatants at the front. Individual surrenders were uncommon; large units usually surrendered en masse. At the siege of Maubeuge about 40,000 French soldiers surrendered, at the battle of Galicia Russians took about 100,000 to 120,000 Austrian captives, at the Brusilov Offensive about 325,000 to 417,000 Germans and Austrians surrendered to Russians, and at the Battle of Tannenberg 92,000 Russians surrendered. When the besieged garrison of Kaunas surrendered in 1915, some 20,000 Russians became prisoners, at the battle near Przasnysz 14,000 Germans surrendered to Russians, and at the First Battle of the Marne about 12,000 Germans surrendered to the Allies. 25-31% of Russian losses were to prisoner status; for Austria-Hungary 32%, for Italy 26%, for France 12%, for Germany 9%; for Britain 7%. Prisoners from the Allied armies totalled about 1.4 million . From the Central Powers about 3.3 million men became prisoners; most of them surrendered to Russians. Germany held 2.5 million prisoners; Russia held 2.2-2.9 million; while Britain and France held about 720,000. Most were captured just before the Armistice. The United States held 48,000. The most dangerous moment was the act of surrender, when helpless soldiers were sometimes gunned down. Once prisoners reached a camp, conditions were, in general, satisfactory , thanks in part to the efforts of the International Red Cross and inspections by neutral nations. However, conditions were terrible in Russia: starvation was common for prisoners and civilians alike; about 15-20% of the prisoners in Russia died, and in Central Powers imprisonment 8% of Russians. In Germany, food was scarce, but only 5% died.

How many years after the first Kandyan War did the second Kandyan War happen?

During the Napoleonic Wars, Great Britain, fearing that French control of the Netherlands might deliver Sri Lanka to the French, occupied the coastal areas of the island with little difficulty in 1796. In 1802, the Treaty of Amiens formally ceded the Dutch part of the island to Britain and it became a crown colony. In 1803, the British invaded the Kingdom of Kandy in the first Kandyan War, but were repulsed. In 1815 Kandy was occupied in the second Kandyan War, finally ending Sri Lankan independence. Following the suppression of the Uva Rebellion the Kandyan peasantry were stripped of their lands by the Wastelands Ordinance, a modern enclosure movement, and reduced to penury. The British found that the uplands of Sri Lanka were very suitable for coffee, tea and rubber cultivation. By the mid-19th century, Ceylon tea had become a staple of the British market bringing great wealth to a small number of white tea planters. The planters imported large numbers of Tamil workers as indentured labourers from south India to work the estates, who soon made up 10% of the island's population. These workers had to work in slave-like conditions living in line rooms, not very different from cattle sheds.

Figure 2: Question Answering HIT sample above with passage on the left and input fields for answer on the right and Highlighted candidate spans of sample answers below