BiQuAD: Towards QA based on deeper text understanding

Frank Grimm **Philipp Cimiano**
CIT-EC, Universität Bielefeld
{fgrimm, cimiano}@cit-ec.uni-bielefeld.de

Abstract

Recent question answering and machine reading benchmarks frequently reduce the task to one of pinpointing spans within a certain text passage that answers the given question. Typically, these systems are not required to actually understand the text on a deeper level that allows for more complex reasoning on the information contained. We introduce a new dataset called *BiQuAD* that requires deeper comprehension in order to answer questions in both extractive and deductive fashion. The dataset consist of $4,190$ closed-domain texts and a total of $99,149$ question-answer pairs. The texts are synthetically generated soccer match reports that verbalize the main events of each match. All texts are accompanied by a structured Datalog program that represents a (logical) model of its information. We show that state-of-the-art QA models do not perform well on the challenging long form contexts and reasoning requirements posed by the dataset. In particular, transformer based state-of-the-art models achieve F_1-scores of only 39.0. We demonstrate how these synthetic datasets align structured knowledge with natural text and aid model introspection when approaching complex text understanding.

1 Introduction

Most of the recent question answering benchmarks require systems to pinpoint the span of the answer to the question in the given text. In the well-known SQuAD 2.0 dataset (Rajpurkar et al., 2018), systems are able to extract the correct answer span in the following paragraph as an answer to the question: "*What tactic did researchers employ to offset the former deficit of work surrounding the complexity of algorithmic problems?*":

"*Before the actual research explicitly devoted to the complexity of algorithmic problems started off, **numerous foundations were laid out by various researchers**. Most influential among these was the definition of Turing machines by Alan Turing in 1936, which turned out to be a very robust and flexible simplification of a computer.*" (sample from Rajpurkar et al. (2018))

Results of state-of-the-art (SOTA) systems on these datasets have reached Exact Match (EM) and F_1 performances of 90.9 and 93.2, respectively [1]. Some of these models even outperform the human baseline (which lies at $F_1 = 89.4$ for SQuAD 2.0).

It is unclear to which extent the existing benchmarks actually require systems to comprehend texts and to what extent these systems rely on surface cues signalling a match between question and answer span. Most state-of-the-art models rely on transformer models such as BERT and ALBERT (Lan et al., 2020) that are pre-trained on supplementary tasks using large amounts of textual data (Devlin et al., 2019) and employ extensive self-attention mechanisms that have been shown to learn many of the features of a classic natural language processing (NLP) pipeline (Tenney et al., 2019). It has been shown for a number of tasks that such models rely on surface cues and on artifacts of the datasets. A recent example is the Argument Reading and Comprehension (ARC) task (Habernal et al., 2018). A deeper analysis of the data and the performance of transformer models has shown that they exploit only surface cues and artefacts of the data, failing to perform beyond chance when systematic (adversarial) modifications are applied on the dataset (Niven and Kao, 2019).

Our motivation in this paper is to introduce a new question answering dataset that requires a deeper understanding of the text to answer questions beyond merely matching answer spans. In particular, in our dataset, the answers to questions can often

[1] Current performance of 'FPNet (ensemble)' at the time of writing according to the SQuAD 2.0 leaderboard https://rajpurkar.github.io/SQuAD-explorer/.

Proceedings of the 10th Conference on Lexical and Computational Semantics, pages 105–115
August 5–6, 2021, Bangkok, Thailand (online)

not be found in the original text, but can be inferred on the basis of a deeper, structural, understanding. Examples are aggregation questions such as "*'How many goals did Marco Reus score in the first half of the match?'*" but also questions such as "*'Who won the game?'*", requiring a system to understand that a balanced score leads to a tie, making the question unanswerable. Since most models cannot keep track of all the goals and intermediate scores for the whole game this provides a significant challenge. The dataset we present is called *BiQuAD* and comprises of 99,149 question answer pairs on 4,190 documents, averaging 23 questions per document. The texts describe soccer matches that have actually taken place. The texts have been generated automatically on the basis of handcrafted templates from structured reports of soccer games. A model of each game is available in the form of a Datalog program representing the meaning of the text in terms of a model consisting of predicates relevant to the description of a soccer game. The questions are paraphrases of queries that can be answered over the model of the game. Similar to extensions from SQuAD 1.0 to 2.0 (Rajpurkar et al., 2018), a percentage of the generated questions in the dataset are deemed as *unanswerable*.

In this paper we present the dataset in more detail and describe its creation (section 3). Further, we present the results of state-of-the-art QA systems on this task in section 4. We show the limits of current state-of-the-art QA in answering questions for which the answer is not in the text but requires deeper inference on the basis of the information given in the text. We show that, in spite of text being artificially generated using 335 relatively simple templates, thus being very regular, this task is not solvable by the current state-of-the-art in question answering. Arguably, the current state-of-the-art focuses on extractive QA and was not designed for deeper understanding. We posit that results on our dataset show that extractive models in particular overfit on surface cues that do not require deeper text understanding. In particular, we show that while the state-of-the-art on Squad 2.0 for example yield results of EM and F_1-scores of 90.7 and 93.0, results of these models for our task range between 38.8 and 39.0 respectively.

2 Related Work

Datasets on machine reading and question answering tasks can be characterized by broad categories: *a)* open vs. closed (specific) domain, and *b)* text-comprehension based (e.g. extractive or Cloze-style) vs. knowledge based QA. Table 1 categorizes prominent datasets along these lines and provides an overview over the number of questions/-documents, how each dataset was collected (e.g. crowdsourcing / artificially generated) as well as the current state-of-the-art (SOTA) results. This list is necessarily non-exhaustive and we only showcase datasets that are either prominent examples of the space or noteworthy because of their relation to the work presented here.

Prominent datasets focusing on open-ended extractive QA include SQuAD 1.0 (Rajpurkar et al., 2016) and 2.0 (Rajpurkar et al., 2018), which have enjoyed wide popularity in the research community. Together with NewsQA (Trischler et al., 2017), these datasets represent the largest, crowd-sourced, extractive QA datasets available. Questions here are answered by correctly identifying a span in a context paragraph, with version 2.0 introducing a subclass of unanswerable questions. More recently, SQuAD versions in languages other than English have been developed (Croce et al., 2018; Mozannar et al., 2019; d'Hoffschmidt et al., 2020; Carrino et al., 2020). The TriviaQA (Joshi et al., 2017) dataset integrates the notion of external evidence (Wikipedia articles) for trivia and open domain question answering and broadens the task to information retrieval (IR) settings.

The RACE dataset (Lai et al., 2017) relies on a multiple choice setting and leverages data used for the assessment of reading comprehension by humans. It features simple reasoning challenges such as deducing relative values from mentions of absolute ones. Similarly, the Open Book QA (Mihaylov et al., 2018) requires models to involve common sense knowledge to solve the task successfully.

There are different ways to frame the task of answering questions by machine reading, including sentence retrieval (Momtazi and Klakow, 2015), multi-hop reasoning (Khot et al., 2020), and reasoning about multiple paragraphs or documents at the same time (Dua et al., 2019; Cao et al., 2019). Recent work has considered the development of reasoning-based QA systems (Weber et al., 2019) as well as the integration of external (Banerjee and Baral, 2020) and commonsense knowledge (Clark et al., 2020) into the QA process.

Other machine reading based QA datasets focus on answering questions on the basis of struc-

Dataset	Domain	Task Type	Samples ~	Acquisition Method	SOTA	
SQuAD 1.0, 2.0	Open	Extractive QA	150,000	Crowdsourcing	EM 90.724	F_1 93.011
NewsQA	Open	Extractive QA	120,000	Crowdsourcing	F_1 73.6	
TriviaQA	Open	Extractive QA	95,000	Semi-Automatic	EM 90.38	F_1 92.96
RACE	Open	Multiple Choice	100,000	Domain Experts	Accuracy 90.9	
Open Book QA	Open	Multiple Choice	6,000	Crowdsourcing	Accuracy 87.20	
LC-QuAD 2.0	Open	Graph Retrieval	30,000	Semi-Automatic	*tbd.*	
WikiHop	Open	Extractive QA	50,000	Graph Traversal	Accuracy 81.9	
MedHop	Closed	Extractive QA	2,500	Graph Traversal	Accuracy 60.3	
QASC	Open	Multiple Choice	10,000	Crowdsourcing	Accuracy 90	
QALD-9	Open	Graph Retrieval	700 * 11	Manual Annotation	Macro F_1 QALD 5.0	
SciQA	Closed	Document Retrieval	10,000	Automated Extraction	MAP 24.36*	
DROP	Both	Extractive QA	97000	Crowdsourcing	EM 90.10	F_1 87.04

Table 1: Overview of related QA datasets. Exact match (EM), F_1, Accuracy, Mean Average Precision (MAP) scores according to their respective leaderboards at the time of writing. *Best result on BioASQ 6b test batch 3 (Nentidis et al., 2018).

tured knowledge graphs. A prominent example is the series of Question Answering over Linked Data (QALD) evaluation campaigns (Usbeck et al., 2018), now in its 9th edition and going back to 2011. Solving the QALD tasks requires mapping natural language questions in multiple languages into a corresponding SPARQL query (Cimiano et al., 2013). While QALD provided only hundreds of training samples, recent datasets such as LC-QuAD 2.0 (Trivedi et al., 2017; Dubey et al., 2019) rely on automatic generation and human post-processing to generate sufficient sample counts required for modern deep learning architectures. A similar approach is taken by QASC (Khot et al., 2020) or WikiHop and MedHop (Welbl et al., 2018) that are aimed at multi hop inference across multiple documents, finding answers directly from the KG without the need to generate a query.

Most closed domain datasets are smaller than their open domain counterparts since annotation usually requires experts that are inherently harder to source. Datasets such as the biomedical question answering corpus BiQA (Lamurias et al., 2020) make use of user generated content from other sources instead.

The recent DROP dataset (Dua et al., 2019) focuses on complex reasoning tasks in form of both, open and closed domain, questions. The task combines the challenge of extractive QA with testing a models ability to perform limited numerical reasoning, e.g. by having to calculate date differences. At the time of this writing, graph-based models presented in (Chen et al., 2020) rank at the top of the DROP leaderboard with F_1 of 90.1, and EM of 87.0.[2]

Similar to DROP, we aim to bridge gaps required for deeper text understanding while simultaneously providing sample annotations that allow for proper model introspection. The synthetic nature of generated texts in *BiQuAD* provides an extensible way to test model capabilities for reasoning about various sub-categories. We provide long form text passages alongside structured representations, in the form of Datalog rules, as a way to either explicitly combine structured and unstructured information or a further way for model introspection by aligning neural model representations with their graphical and discrete counterparts.

3 Methods

This section outlines the methodology used to generate the dataset and in what way the state-of-the-art transformer architectures can provide a first baseline for the *BiQuAD* dataset.

3.1 Dataset Generation

The *BiQuAD* dataset consists of two different views on each match: a *Datalog* program representing a model of each text in terms of the main events in the match, as well as an artificially generated *match report* in natural language. Each of these match reports is accompanied by a set of, on average, 23 question/answer pairs generated in a similar fashion.The dataset is made available as fixed 60-20-20 training/development/test splits.

The Datalog programs and natural language match reports are extracted and generated on the basis of structured match reports available as part of

[2] https://leaderboard.allenai.org/drop/submissions/public

the "European Soccer Database" (ESDB),[3] which aggregates real minute-by-minute data on 14,196 historic soccer matches between 2008 and 2016 from various sources. It is licensed under the permissive Open Database License (ODbL) v1.0. We extract data following a scheme of match objects and associated events from the database, an overview of which is available in Fig. 1.

The structure presents a complete model of the facts contained in the database and subsequently all texts generated using its information. It contains information about the following eight types of events for each match: `cross` (medium- / long-range pass), `foul`, `shot on target` (goal shot attempts), `shot off target` (shots not reaching the goal or hitting the frame), `card` (yellow or red), `goal`, `possession` (special event reporting the minutes each team is in ball possession). The available details for each event vary from sub-types, e.g. indicating the type of card, the reason on why it was given, to the individual players involved in a foul or cross pass. These details are used to create a natural sounding output sentence from each event. The Datalog programs capture all these events in addition to the final result via logical predicates and provide thus the basis for structured querying, supporting aggregation questions.

The transformation of the data to Datalog programs relies on a number of rules that extract data from the ESDB and transforms it into Datalog clauses. The full set of rules, as well as a comprehensive overview, is available for download alongside the QA dataset itself. Applied to a full object hierarchy of a match and all its events in the ESDB this generates a set of Datalog programs and documents of various sizes. While some matches only describe major events, like goals or cards, the majority averages more than one event per minute.

The following example represents information on the overall outcome of a certain match in both Datalog (shown here in a standard variant) and text:

[3] https://www.kaggle.com/hugomathien/soccer

EXAMPLE 1.

```
match(M47).
match_league(M47, "Bundesliga").
match_hometeam(M47,"Borussia Dortmund").
match_awayteam(M47, "Bayern München").
match_score(M47, "2:1").
match_hometeam_goals(M47, "2").
match_awayteam_goals(M47, "1").
```

```
The Bundesliga match ended with the
home team Borussia Dortmund beating
Bayern München 2 to 1.
```

Text Generation: The natural language match report is generated via a set of templates defined for the different types of events. An additional set of templates describes the overall match in various degrees of detail. Similar to the above Datalog transformation, we represent transformations into text as natural language with dynamic placeholders:

EXAMPLE 2.

```
The {@data.league} game of
{@data.hometeam} versus {@data.awayteam}
ended {@data.score}.
```

For text generation we defined a total of 335 rules, with least 5 different rules for each relevant event, so that some language variation is introduced. Filter functions are used to postprocess the names of players by removing parts of the full name to check the ability of systems to detect and resolve co-references. Some rules also introduce explicit co-reference markers, allowing generated systems to replace a player's name that occurs in multiple subsequent events.

EXAMPLE 3.

```
The {@data.league} match ended with the
home team {@data.hometeam} beating
{@data.awayteam} {@data.home_goals}
to {@data.away_goals}.
```

Question Templates ($QT_{category}$) are used to generate (question, answer) pairs on the original Datalog program. They are used to construct a Datalog query that answers the question given structured knowledge, as well as their textual representation. The answer is then retrieved by executing the constructed query and, where possible, annotated in text form. The templates, outlined below, have each been designed to address a specific challenge for the QA task. Similar to how most SQuAD evaluations report performance on answerable and

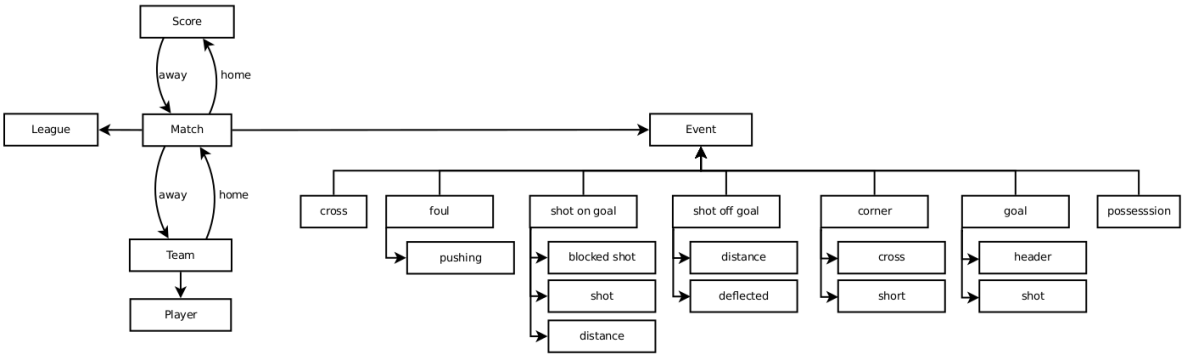

Figure 1: ESDB object model overview.

unanswerable questions separately, this allows researchers to evaluate their models along detailed axes and pinpoint potential for improvements. The dataset features the following nine types of questions:

- $QT_{Simple\ Facts}$ This template takes a fact from the Datalog program and randomly removes an entity, this generates cloze style questions answerable using a single sentence in the textual representation, e.g. "*Who won the game?*".
- $QT_{Multiple\ Facts}$ Questions that relate to multiple entities, akin to the extraction of relations with two arguments, such as "*Who did :event/player1 tackle?*".
- $QT_{Paraphrased\ Facts}$ This is an extension of $QT_{Simple\ Facts}$ that generates similar questions but is mapped to different text templates that change and omit entity labels (e.g. omitting the first name of a player or omitting a subsequent use of team names). These templates require models to learn inexact matching of labels to entities in the underlying knowledge base and introduce simple patterns of co-reference resolution.
- $QT_{Aggregation}$ (*min/max/count*): Events such as goals, cards, and fouls are discrete entries in the Datalog program. While some resulting questions might be available in text, e.g. "*How many goals did Team A score?*", some require further deduction via counting ("How many goals did Marco Reus score?"). This also includes comparisons between multiple entities, e.g. "Did :player1 score more goals than :player2?".
- $QT_{Unanswerable}$ Aligned with Rajpurkar et al. (2018), this template introduces an adversarial element to the dataset by generating questions that look valid but are in fact not answerable by the data. These questions are generated by randomly sampling another document and ensuring the resulting Datalog query does not yield a result for the current match. This leads to realistic questions that might even overlap in entities (e.g. player names) but make no sense in the context of the current document.
- $QT_{Temporal}$ This template generates questions relating to the temporal order of match events; it generates questions such as "*Who scored the first goal of the match?*" or "*Who scored the last goal in the first half of the match?*".
- $QT_{Aggregation\ Temporal}$ This template combines $QT_{Multiple\ Facts}$ and $QT_{Temporal}$ by asking questions such as "How many goals did Team A score in the second half of the match?" or "Did Team A score more goals in the first half of the match?" (comparison).

All templates contain placeholders such as `:event/player1` that are dynamically resolved via the knowledge in the Datalog program for each match and can refer to individual properties or entity names. Question templates may provide preconditions or constraints that have to be true in order for the question to be generated on any given match. This ensures that questions are answerable and compatible with the given match data.

A question about one player tackling another for example would generate the following Datalog query and text:

EXAMPLE 4.

```
Q(m, p1) :- event_match(e, m),
   event_player1(e, p1),
   event_player2(e, p2),
   event_type(e, "foulcommit"),
   event_subtype(e, "pull").
```

```
Who pulled {@data.player2|namefilter}?
```

The templates in category $QT_{Unanswerable}$ are generated in a second pass over the dataset, for each four answerable questions an unanswerable one is sampled randomly from another document in the corpus. The answer for the sampled question is dropped and the accompanying datalog query is used to validate that no answer exists for the question w.r.t. the current document. This reliably transforms templates that generate sensible and answerable questions into unanswerable ones.

Template annotations include a number of answer types (e.g. *text* or *numeric*) that are used to annotate the location of the answer within the context document. The datalog query of the question is automatically rewritten to obtain a sensible location within the text to annotate an answer. In the case of textual answers, the closest matches are annotated since the exact location might not contain the answer itself due to co-reference. For numerical answers we employ two strategies: *a)* temporal questions looking for the minute a particular event occurred are annotated at the appropriate marker and *b)* generic numeric answers are generated at the end of the document (ensuring that at least five multiple choices are presented, padded with random numbers if necessary).

All questions consist of four elements:

- Question in natural language.
- Datalog query corresponding to the question.
- Answer retrieved from the knowledge base.
- Metadata, such as the answer type (text, numeric) and question template category.

In Figure 2 we give an example excerpt from an automatically generated report for a single match. One of the questions w.r.t. this match in the dataset is the following:

- NL Question: How many goals did Fulham score in the first halftime?
- Datalog Query (excerpt):

```
Q(m, team) :- event_match(e, m),
     event_type(e, "goal"),
     event_team(e, team),
     event_minute(e) <= 45.
A :- sum(Q(m, "home")).
```

- Answer: *3 (three)*

The England Premier League match between Fulham and Norwich City ended 5:0.

2: Foul in minute 2: Handball by Johnson.
4: Anthony Pilkington takes the ball.
5: Sascha Riether pulls against Pilkingtons shirt.
6: Snodgrass takes the ball.
[...]
25: Dangerous play foul by Grant Holt on Mahamadou Diarra in minute 25.
26: Goal by Duff for Fulham.
29: Tierney shoots off.
30: In minute 30 John Arne Riise attempts a cross.
30: Incident between Fulham player Hangeland and Grant Holt results in penalty.
32: Dembele fouls Holt in minute 32 by obstructing.
[...]
87: Steve Sidwell scores for Fulham.
89: Opponent player is tackled from behind by Mahamadou Diarra.
92: Penalty against Brede Hangeland in minute 92 after incident with Morison.

Figure 2: Example match report excerpt between Fulham and Norwich City in 2012.

After generation, we generate fixed splits by shuffling all match report documents and dividing them into sets of train (60%), development (20%), and test (20%). The dataset, as well as an evaluation script and resources for generating it, are made available online. In order to provide a comparable baseline to existing state-of-the-art models the subset of QA pairs suitable for extractive QA is exported into a SQuAD-compatible data format.

3.2 Model

In order to provide first reference results for the *BiQuAD* dataset, we evaluate state-of-the-art vanilla transformer architectures on the task, in particular ALBERT (Lan et al., 2020). Based on hardware constraints and hyperparameter optimization using fine-tuned *base* models, all trainings ran for two epochs, with a learning rate of $3e{-}5$ and a batch size of 8. The model itself uses a standard ALBERT architecture for question answering tasks following (Lan et al., 2020; Rajpurkar et al., 2018). The tasks were executed in a Linux cluster environment on GTX1080 Ti GPUs (CUDA10). A single additional hidden layer stores answer span logits (start and end of a span) and treats the first token (`[CLS]`) as an indicator for unanswerable questions. Optimization is performed through Adam (with $epsilon = 1e{-}8$).

Input Sequences in both models are constrained by the maximum sequence length they can process. Similar to the size of the parameter space, the specific maximum values (512 tokens for both models used here) depends on the maximum sequence

length used during pre-training. We leverage the ability of the HuggingFace implementation (Wolf et al., 2019) to specify a *document stride*. This effectively extracts features in a sliding window approach over the full document and determines the answer by observing the maximal logit over all windows.

3.3 Evaluation

The evaluation of each model is performed in two major settings: *a)* single question-answering and *b)* document level question-answering. While the former aims to maintain compatibility with SQuAD-like datasets and optimizes on the overall ability of a model to extract individual answers from the textual description of a soccer match, the latter is designed to assess the overall ability of the model to cover lots of different questions on a particular complex text document.

Single Question Answering pairs are evaluated in accordance with the SQuAD paradigm of exact matches (EM) and F_1 scores (Rajpurkar et al., 2018):

- Exact Match (EM): Percentage of predictions completely matching the ground truth answer.
- F_1: Macro-averaged F_1 score. Here each answer and ground truth pair is treated as a bag-of-words to determine an inexact overlap and evaluate in a less strict manner. The calculation of individual F_1 scores follows the classical definition of $F_1 = 2\frac{Precision*Recall}{Precision+Recall}$ (van Rijsbergen, 1979). Unlike SQuAD our datasets only present a singular ground truth per question so it is unnecessary to search for a maximal F_1 score here.

For the purpose of error analysis, in this article we report these results on a per-category level, as well as distinguished by answerable and unanswerable questions. This not only showcases where a particular model might have problems with the dataset it is trained on, it also is of immense help when debugging errors and guiding decisions of where a model might require more training data.

Transfer Learning from open domain question answering datasets such as SQuAD 2.0 is used to assess how well the language structure itself can be used for extractive QA in the unseen data of our dataset. For this we evaluate the aforementioned experimental setup after training on the SQuAD 2.0 training split and evaluating on the test split of *BiQuAD* .

All results are reported on the development split, the test split is withheld from public release for use in an evaluation webservice. In the open source release of the *BiQuAD* dataset, evaluation scripts are provided in order to keep these evaluations consistent.[4] To aid in reproducibility, the MIT-licensed open source release also contains the scripts required to generate samples.

4 Results

This section provides an overview of the dataset and provides first results on the dataset by providing baselines relying on state-of-the-art transformer models.

4.1 Dataset

The dataset comprises 4,190 documents with play by play soccer matches and 99,149 questions ($\sim$ 23 per document) and is thus of similar size as comparable datasets. Each textual representation of a match contains an average of 82 sentences (759 words). The template based text generation yielded long form documents of rather factual and sober match descriptions. While syntactic and vocabulary variability is clearly limited due to this rule-based approach, co-reference and detailed event descriptions make texts non-trivial to reason about.

Albeit not being leveraged in the baseline models presented herein, the parallel construction of textual and structured descriptions enables the adoption of *BiQuAD* for use in further downstream tasks, such as relation extraction or knowledge base completion.

The dataset splits follow a 60-20-20 scheme, it contains a *training* split with 2,514 documents (58,807 QA pairs) and *Development and Test* splits with 838 documents (19,870 QA pairs) each.

4.2 Model Results

The following results outline the performance of SOTA models, as described in section 3, for question answering on the *BiQuAD* dataset.

Single Question Answering is evaluated on individual question answer pairs in the development dataset, table 2 shows the overall results. Individual question templates, and subsets such as answerable

[4] https://github.com/ag-sc/BiQuAD

$QT_{Answerable}$ and unanswerable $QT_{Unanswerable}$ questions. The latter distinction is important because both models regarded in this study explicitly model if a question is answerable in their architecture and are thus typically well equipped to make this decision.

Question Templates	EM	F_1
All	38.8	39.0
Answerable	25.4	25.8
Unanswerable	86.6	86.6
Simple Facts	25.0	21.6
Multiple Facts	29.7	30.1
Temporal	16.4	16.4
Aggregation	33.8	64.1
Aggregation Temporal	0.0	66.3

Table 2: Results for single QA setting, reported results on the public development split.

On the question of answerability, the models perform reasonably well. The fact that it does not achieve perfect scores here indicates that they are not able to exploit simple surface queues to make this decision and validates the approach to generate these samples as described in section 3. Other template categories indicate that the model cannot yet cope with more involved categories, such as the temporal reasoning category.

Transfer Learning results, presented in table 3, evaluate the QA pairs in the development set of *BiQuAD* on a model trained with the training data from SQuAD 2.0. The results show a strong ability to determine the answerability of questions but break down in other categories. This capability indicates that questions generated as unanswerable might often contain easily spotted and exploited surface clues, such as player names, that do not occur in the document in question.

The results presented here show that modern deep learning models such as ALBERT perform similarly well on the proposed dataset. While SQuAD-like datasets are commonly limited to the evaluation of subsets such as $QT_{Answerable}$ and $QT_{Unanswerable}$, the template based nature of *BiQuAD* allows for even further inspection.

5 Conclusion

We have introduced a challenging new QA dataset that emphasizes document-level text understanding. While most of the existing benchmark datasets

Question Templates	EM	F_1
All	24.4	25.9
Answerable	3.7	5.7
Unanswerable	99.0	99.0
Simple Facts	6.5	6.8
Multiple Facts	16.9	24.1
Temporal	0.0	0.0
Aggregation	0.0	2.5
Aggregation Temporal	0.0	0.9

Table 3: Results for the transfer learning QA setting, reported results on the public development split.

require systems to extract answer spans from text or to select an answer given multiple choices, we have attempted to provide a dataset that requires answering questions beyond the content explicitly mentioned in the text, requiring inference and aggregation on top of the information given in the text. Our methodology builds on a structured database of soccer matches. From this database, it generates natural language reports of games relying on a set of handcrafted templates in addition to a Datalog logical representation of a text. For each document, 23 Datalog queries are generated and transformed into NL relying on a further set of handcrafted templates. The artificial nature of the dataset allows for clear cut error analysis and can guide the implementation of, especially closed domain, QA systems. Downsides introduced by synthetic generation include an unnaturally factual text style and the fact that relatively simple heuristics might be able to generate the correct answer for some of the question templates. These heuristics could potentially even target the soccer domain itself, as is common with many subtasks in closed domain corpora. We aim to alleviate these concerns in future work by extending *a)* the tooling around model introspection and *b)* coverage of further domains. These steps should further improve the way reasoning tasks are reflected in the dataset and establish *BiQuAD* , or the approach of synthetic corpora, as a modeling tool to be used alongside other corpora of natural texts in comparable or open domain settings.

We have provided baseline systems and show that existing state-of-the-art systems yield very low results on the dataset, with exact match and F_1-scores of 38.8 and 39.0, respectively. Results of $EM = 24.4, F_1 = 25.9$ in a transfer learning setup further strengthen the assumption that these models cannot sufficiently answer the type of rea-

soning tasks imposed by the dataset. While state-of-the-art systems perform well on standard extractive questions ($QT_{Multiple\ Facts,Aggregation}$), we show that on questions requiring inference and temporal reasoning, the baseline systems perform at F_1 below 20 even when encoding answers in a way compatible with extractive systems.

On the basis of these results and experience with other datasets, we can see that explicit modelling is required for various complex reasoning tasks. Many recent state-of-the-art models on QA achieve this complexity only in limited scopes, such as pure numeric reasoning, not deeper general text understanding. The parallel construction of structured Datalog knowledge about a closed world model may be used to *a)* develop models that combine textual information with external semantic knowledge or *b)* decode how a model performs reasoning tasks by linking text and structural knowledge when inspecting individual components such as attention layers. *BiQuAD* allows to model and inspect reasoning on specific categories without necessarily overfitting on any particular subtask. Our dataset is freely available and, in combination with other datasets of non-synthetic nature, will hopefully contribute to push the state of the art in machine reading further.

Acknowledgments We would like to thank the anonymous reviewers for their valuable feedback.

References

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–162, Online. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2306–2317. Association for Computational Linguistics.

Casimiro Pio Carrino, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. Automatic spanish translation of squad dataset for multi-lingual question answering. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 5515–5523. European Language Resources Association.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020. Question directed graph attention network for numerical reasoning over text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6759–6768, Online. Association for Computational Linguistics.

Philipp Cimiano, Vanessa López, Christina Unger, Elena Cabrio, Axel-Cyrille Ngonga Ngomo, and Sebastian Walter. 2013. Multilingual question answering over linked data (QALD-3): lab overview. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization - 4th International Conference of the CLEF Initiative, CLEF 2013, Valencia, Spain, September 23-26, 2013. Proceedings*, pages 321–332.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3882–3890. ijcai.org.

Danilo Croce, Alexandra Zelenanska, and Roberto Basili. 2018. Neural learning for question answering in italian. In *AI*IA 2018 – Advances in Artificial Intelligence*, pages 389–402, Cham. Springer International Publishing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. Fquad: French question answering dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 1193–1208. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2368–2378. Association for Computational Linguistics.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over

wikidata and dbpedia. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779 of *Lecture Notes in Computer Science*, pages 69–78. Springer.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. SemEval-2018 task 12: The argument reasoning comprehension task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 763–772, New Orleans, Louisiana. 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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. QASC: A dataset for question answering via sentence composition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8082–8090. AAAI Press.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 785–794. Association for Computational Linguistics.

Andre Lamurias, Diana Sousa, and Francisco M. Couto. 2020. Generating biomedical question answering corpora from q&a forums. *IEEE Access*, 8:161042–161051.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics.

Saeedeh Momtazi and Dietrich Klakow. 2015. Bridging the vocabulary gap between questions and answer sentences. *Inf. Process. Manage.*, 51(5):595–615.

Hussein Mozannar, Elie Maamary, Karl El Hajal, and Hazem M. Hajj. 2019. Neural arabic question answering. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop, WANLP@ACL 2019, Florence, Italy, August 1, 2019*, pages 108–118. Association for Computational Linguistics.

Anastasios Nentidis, Anastasia Krithara, Konstantinos Bougiatiotis, Georgios Paliouras, and Ioannis Kakadiaris. 2018. Results of the sixth edition of the BioASQ challenge. In *Proceedings of the 6th BioASQ Workshop A challenge on large-scale biomedical semantic indexing and question answering*, pages 1–10, Brussels, Belgium. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. 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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

C. J. van Rijsbergen. 1979. *Information Retrieval*. Butterworth.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 191–200. Association for Computational Linguistics.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer.

Ricardo Usbeck, Ria Hari Gusmita, Axel-Cyrille Ngonga Ngomo, and Muhammad Saleem. 2018. 9th challenge on question answering over linked data (QALD-9) (invited paper). In *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018*, volume 2241 of *CEUR Workshop Proceedings*, pages 58–64. CEUR-WS.org.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. Nlprolog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6151–6161. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.