TempQuestions: A Benchmark for Temporal Question Answering

Zhen Jia*
School of Information Science
Southwest Jiaotong University
China
zjia@swjtu.edu.cn

Abdalghani Abujabal
Max Planck Institute for Informatics
Saarland Informatics Campus
Germany
abujabal@mpi-inf.mpg.de

Rishiraj Saha Roy
Max Planck Institute for Informatics
Saarland Informatics Campus
Germany
rishiraj@mpi-inf.mpg.de

Jannik Strötgen
Max Planck Institute for Informatics
Saarland Informatics Campus
Germany
jannik.stroetgen@mpi-inf.mpg.de

Gerhard Weikum
Max Planck Institute for Informatics
Saarland Informatics Campus
Germany
weikum@mpi-inf.mpg.de

ABSTRACT

Answering complex questions is one of the challenges that question-answering (QA) systems face today. While complexity has several facets, question dimensions like temporal and spatial intents necessitate specialized treatment. Methods geared towards such questions need benchmarks that reflect the desired aspects and challenges. Here, we take a key step in this direction, and release a new benchmark, *TempQuestions*, containing 1,271 questions, that are all *temporal* in nature, paired with their answers. As a key contribution that enabled the creation of this resource, we provide a crisp *definition* for temporal questions. Most questions require decomposing them into sub-questions, and the questions are of a kind that they would be best evaluated on a combination of structured data and unstructured text sources. Experiments with two QA systems demonstrate the need for further research on complex questions.

CCS CONCEPTS

• **Information systems → Test collections**;

KEYWORDS

Question answering; Temporal questions; Benchmarks

ACM Reference Format:
Zhen Jia, Abdalghani Abujabal, Rishiraj Saha Roy, Jannik Strötgen, and Gerhard Weikum. 2018. TempQuestions: A Benchmark for Temporal Question Answering. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3184558.3191536

1 INTRODUCTION

Motivation. Answering natural-language questions (QA) has been intensively researched over the last few decades. Earlier approaches, up to when IBM Watson won the Jeopardy! quiz show, have mostly tapped into textual sources (including Wikipedia articles) using passage retrieval and other techniques [14, 23]. In the last few years, the paradigm of translating questions into formal queries over structured knowledge bases (KBs) and data bases (DBs, including Linked Open Data) has become prevalent [7, 30, 33].

QA over structured data (KB-QA) translates the terms in a question into the vocabulary of the underlying KB or DB: entity names, semantic types, and predicate names for attributes and relations. State-of-the-art systems (e.g., [1, 5, 6, 34]) perform well for simple questions that involve a few predicates around a single target entity (or a qualifying entity list). A typical question is:

"Which film by Luc Besson did Bruce Willis star in?"

which can be translated into a SPARQL query, like:

```
SELECT ?x WHERE {
?x type movie.?x directedBy LucBesson.BruceWillis actedIn ?x}
```

with the answer: *'The Fifth Element'*.

However, KB-QA has limitations regarding *complex questions* that require decomposing the input into sub-questions. A typical example is (with the answer being *'Milla Jovovich'*):

"Which actress in a Besson movie married him?"

Here, a SPARQL query would require multiple query variables, and a three-way join between actresses, movies and directors. Such complex questions are too difficult for today's KB-QA systems. Decomposing the question into *"actress in a Besson movie"* and *"actress married Besson"*, and subsequently intersecting their results would be a viable execution plan, though.

The need for this kind of decomposition arises for all kinds of complex questions. In this paper, we focus on a specific kind of user input, namely, *temporal questions*. A substantial fraction of online information needs are time-dependent [4, 20]. Even when a search request does not explicitly refer to dates or events, computing answers may require testing temporal conditions. Consider the example (the answer again being *'Milla Jovovich'*):

*The work was done when the author was at the MPI for Informatics.

WWW '18 Companion, April 23–27, 2018, Lyon, France

ACM ISBN 978-1-4503-5640-4/18/04.
https://doi.org/10.1145/3184558.3191536

*"Which actress starred in Besson's **first** science fiction and **later** married him?"*

A QA system could decompose this into sub-questions like SQ1: *"science fiction movies directed by Luc Besson"*, SQ2: *"actresses starring in Luc Besson movies"*, and SQ3: *"actresses married to Luc Besson"*. Additionally, we need to filter the results of SQ1 to identify the first (i.e., earliest in time) answer, and we need to test the year of the movie against the date of the marriages for the results of SQ3 to rule out spouses who pre-dated that movie.

An ideal execution plan for this complex question needs to compute this *decomposition*, and also needs to generate the post-processing in terms of *reasoning* about time points and intervals. The latter is a new aspect that KB-QA has not considered so far. Prior work on text-oriented QA discussed this point [8, 13, 15] but did not aim for general solutions.

Contribution. The quality of QA is usually evaluated by benchmarks. As a first step towards addressing the challenge of handling complex questions, we offer a new benchmark set of temporal questions. The questions are chosen such that many of them require a combination of evaluating sub-questions and reasoning over sub-results (results of the sub-questions).

There already exists a variety of QA benchmarks. For KB-QA, the Free917 [10] and WebQuestions [7] collections are the most popular. Both are vastly dominated by simple questions and do not exercise a system's capability to decompose and process complex questions. The QALD series of evaluation tasks [31] includes both simple and complex questions. However, the number of questions per year is relatively small (50 – 250 questions). The ComplexQuestions collection of [5] contains various types of complex questions: however, temporal questions present only a small fraction. For text-oriented QA, the TREC [2, 32] and CLEF [21] conference series offer a wealth of benchmark questions, but there is no design consideration on harnessing structured data at all.

The benchmark proposed in this paper, called *TempQuestions*, consists of 1,271 temporal questions with gold-standard answers. This collection is derived by judiciously selecting time-related questions from the Free917, WebQuestions and ComplexQuestions sets, with additional curation and tagging of temporal cues.

Our benchmark supports systematic testing and evaluation of how well QA systems can handle temporal questions that require decomposition and reasoning on sub-results. We ran the benchmark with two state-of-the-art QA systems, AQQU [6] and QUINT [1], where source code is available, and found that both performed marginally. This shows that there is ample room for improvement, and emphasizes the need for research on complex questions. TempQuestions is publicly available at the following link: http://qa.mpi-inf.mpg.de/TempQuestions.zip.

2 DEFINING TEMPORAL QUESTIONS

There are diverse types of questions with a temporal aspect. Questions can contain temporal expressions or signals to express temporal relations. Furthermore, questions may ask for some kind of temporal information, e.g., a date. However, to concisely define *temporal questions*, these concepts, i.e., temporal expressions and temporal signals, need to be precisely specified as well. In this section, we first explain these concepts, which are typically used for temporal information annotation in the context of natural language processing (NLP). Then, we define *temporal questions* based on these existing concepts, which we extend according to the requirements for temporal QA, as explained below.

2.1 Temporal Expressions

In NLP, the temporal markup language TimeML [22] is frequently used for annotating temporal information in text documents. It is also the annotation standard used by most tools, which perform temporal annotation automatically, e.g., temporal taggers for temporal expressions [27].

Besides tags for events and temporal relations between two TimeML entities, TimeML contains `TIMEX3` tags for temporal expressions and `SIGNAL` tags for temporal signals (cf. Sec. 2.2). The `TIMEX3` tag is used to annotate temporal expressions of four types: date, time, duration, and set expressions. The semantics of all temporal expressions can be normalized to some value in a standard format, which allows the comparison between temporal expressions – a characteristic of temporal information, which can also be exploited for temporal QA. TimeML's most important attribute to capture the temporal information of temporal expressions is the `value` attribute. In the case of duration and set expressions, the `value` attribute captures the length of the interval, and the `value` attribute of date and time expressions contains information how to anchor the point in time on a timeline of the respective granularity.

According to TimeML's specifications, set expressions refer to the re-occurring nature of an event. Examples are *'once a week'* and *'daily'*. Duration expressions are used to specify the length of an interval. For instance, *'three weeks'* and *'several years'* are two duration expressions. Note that the temporal information might be concrete as in *'three weeks'* or vague as in *'several years'*. Date and time expressions both refer to points in time – though the points in time are of different granularities: all granularities smaller than *'day'* are considered as time expressions, for instance, expressions referring to parts of a day (e.g., *'Monday morning'* and *'yesterday night'*) and expressions referring to a specified time (e.g., *'9 pm'*, *'three o'clock'* and *'February 5, 2018 23:59:59 CET'*). In contrast, date expressions may refer to a particular day (e.g., *'last Thursday'* and *'23rd of November'*) or to any point in time of a coarser granularity (e.g., *'the 21st century'*, *'last year'* and *'September 2016'*).

Note that these examples directly show that date and time expressions can be realized in different ways: fully-specified, relatively specified, underspecified, or implicitly specified [27]. Fully-specified expressions can be normalized without any further context information (e.g., *'September 2016'* as `2016-09`). In contrast, relative expressions require a reference time (e.g., *'last Thursday'*) and underspecified expressions need a reference time and a relation to the reference time (e.g., *'(on) Thursday'*). In both cases, the reference time might be the time of the sentence or a date mentioned in the textual context. If relative and underspecified date and time expressions occur in NL questions, it is thus important that the information about when the question was formulated is also available. Otherwise, questions such as *"Who was the US president two years ago?"* cannot be answered as it is impossible to determine to which year *'two years ago'* refers.

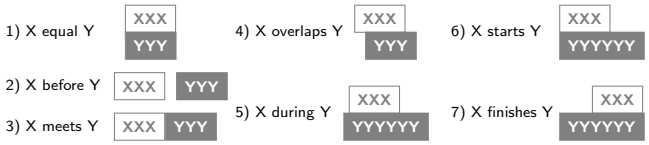

Figure 1: The 13 temporal relations (nos. 2 through 7 have inverses) between two intervals X and Y, as in Allen [3].

Finally, non-standard temporal knowledge is required for normalizing implicit expressions such as holidays (e.g., *'Columbus Day 2018'* – which is, in the US, the second Monday in October). In some works, the definition of implicit temporal expressions has been extended to further include all types of free-text temporal expressions, such as event names or other textual phrases with temporal scopes [16] (e.g., *'Obama's presidency'*, which can be normalized to an interval with a particular start and end date).

In the creation and analysis of our benchmark (Sec. 3 and 4), we will consider questions with fully-specified, underspecified, and relative temporal expressions as *explicit temporal questions*, in contrast to *implicit temporal questions*, which contain implicit temporal expressions including free-text temporal expressions.

2.2 Temporal Signals

TimeML defines *temporal signals* as textual elements that make explicit the *temporal relation* between two TimeML entities (events or temporal expressions), such as *'before'* or *'during'*. In natural language (NL) questions, signals occur, for instance, to explicitly specify a valid time interval for the searched information, as in: *"Which movies did Besson work on before his marriage to Jovovich?"*. Note that we relax the TimeML definition to consider all trigger terms as temporal signals, even if one of the entities is not mentioned explicitly, but is the answer of a question, e.g., in when-questions.

In general, any of the 13 temporal relations defined in Allen's interval algebra for temporal reasoning [3] can be the described relation, that is, the `equal` relation as well as the six relations `before`, `meets`, `overlaps`, `during`, `starts`, and `finishes` with respective inverses (see Fig. 1 for visualizations of the relations). However, due to ambiguities, it is often not possible to select a unique temporal relation for a temporal question. For example, the question *"What did Besson work on before his marriage to Jovovich?"* could be interpreted as asking for either the movie he was working on directly before his marriage or all movies which he was working on any time before his marriage.

It is crucial to point out that NL questions are often formulated with even further ambiguities. While the question *"Which movies did Besson work on before his marriage to Jovovich?"* as well as *"Which movie did Besson work on before his marriage to Jovovich?"* concisely describe the required number of answer movies (several due to plural and one due to singular, respectively), the latter requires the movie which Besson worked on directly before his marriage, i.e., the *temporal constraint* cannot be simply validated, but valid answers have to be sorted and the closest one has to be chosen. In addition, the slightly reformulated question *"What did Besson work on before his marriage to Jovovich?"* could be interpreted one way or the other (singular or plural) – a fact that also makes it sometimes difficult, even for humans, to determine the correct answer of a question.

Due to such ambiguities, in the context of temporal QA, temporal relations could be simplified as the following three types:

(i) `before` and `meet` are treated as the relation `BEFORE`
(ii) `before_inverse` and `meet_inverse` are treated as `AFTER`
(iii) all other relations are treated as `OVERLAP`

Typical *trigger words* suggesting the three temporal relations above, respectively, are the *temporal signals*:

(i) *'before'*, *'prior to'*
(ii) *'after'*, *'following'*
(iii) *'during'*, *'while'*, *'when'*, *'until'*, *'in'*, *'at the same time'*

In addition to the trigger terms defined in TimeML, we add ordinals to the class of temporal signals, as they are often used in NL questions to specify particular instances of items which can be sorted chronologically. An example is *'last'* in *"What was Besson's last movie before his marriage to Jovovich?"*.

2.3 Temporal Questions

Based on the extended concepts of temporal expressions and temporal signals, we can now concisely define a temporal question:

Definition 2.1. A *temporal question* is any question, which contains a temporal expression, a temporal signal, or whose answer is of temporal nature.

Note that this definition is purely *semantic*. In practice, these categories are detected by matching against patterns and lexicons (Sec. 3), accompanied by subsequent reasoning to remove false positives. Thus, various detection techniques (say, for temporal expressions with varying levels of implicitness considerations [16]), may have different recall in the retrieval of temporal questions from a given corpus. Also, note that a temporal question may contain multiple temporal signals and temporal expressions. In addition, any question containing any type of temporal expression is covered under the umbrella of temporal questions. In this work, we consider all temporal expressions independent of their occurrence type as long as they can be anchored on a timeline, either as points in time or as (possibly open) time intervals.

In our analysis of the benchmark in the next section, we distinguish four types of temporal questions: *explicit* and *implicit* with respective temporal expressions (Sec. 2.1), *ordinal* (containing an ordinal), and *temporal answer* which covers all questions asking for some kind of temporal information (e.g., when-questions).

3 TempQuestions: CREATION

Existing KB-QA datasets [5, 7, 10] are mixed bags with several types of questions: simple, compositional, ordinal, temporal, and spatial, among others. While we have reasonable evidence of the presence of temporal questions across these benchmarks, the fraction in each dataset individually is small: as a result, systems that ignore temporal questions can still achieve acceptable performance on these benchmarks. This motivated us to collate temporal questions from existing resources to create our temporal-questions-only benchmark. This was enabled by formulating unambiguous definitions and conventions for temporal questions (Sec. 2). We refer to our new benchmark as *TempQuestions*, and it contains 1,271 questions

with various temporal facets: explicit and implicit temporal expressions, temporal answers, and ordinal constraints. *TempQuestions* is available at: http://qa.mpi-inf.mpg.de/TempQuestions.zip.

Source datasets. Specifically, we extracted temporal questions from the following three KB-QA datasets whose answer sets are based on Freebase:

- **Free917 [10]:** It consists of 917 questions (641 training and 276 test), manually annotated by experts with their SPARQL queries. These factoid questions were provided by two native English speakers.
- **WebQuestions [7] (WQ)**: This has been one of the most popular benchmarks in KB-QA, and contains 5,810 question-answer pairs split into 3,778 training and 2,032 test instances. This dataset was constructed using a combination of Google Suggest API and crowdsourcing.
- **ComplexQuestions [5] (CQ):** It contains 2,100 questions paired with their answers; 1,300 training and 800 test. The questions are samples from query logs of a commercial search engine, together with extractions from previous benchmarks (WebQuestions and Yin et al.'s data [35]). Questions in this dataset are syntactically more complex than questions in previous datasets.

Method overview. We follow a two-stage approach to construct *TempQuestions*: (i) an automated temporal question detection on the above datasets, and (ii) a manual inspection to rule out mistakes in the first step. Additionally, all answers to the final questions were manually verified and mistakes and redundancies in the previous gold standards were corrected.

Automated detection. To identify temporal questions in accordance with the conceptual definitions proposed in Sec. 2, we use a combination of existing taggers, dictionaries, and lexico-syntactic patterns. First, we ran temporal expression taggers SUTime [11] and HeidelTime [26] over all questions. These taggers annotate explicit TIMEX3 tags, and we were thus able to identify questions with explicit temporal expressions (like *"who won the state of texas in [2008]?"*). HeidelTime's temponym tagging extension and an event dictionary created using Freebase were used to identify questions with implicit temporal expressions. SIGNAL words are tagged using a dictionary constructed as per suggestions from Setzer [25], and the list of temporal prepositions (Sec. 2.2) [17, 18] (like *"who lived in america [before] europeans arrived?"*). We tag ordinal words like *first, second,* and *last* using the Stanford CoreNLP [19] recognizer and a dictionary. After this step, we can identify temporal questions like *"who was the [first] coach of the buccaneers?"*. Finally, questions whose answers are temporal are identified using simple start patterns like *when*, *since when*, *what date*, *in what year*, *which century*, etc. We now had 1,541 potential temporal questions. Since we were focused on recall and wanted to collect as many temporal questions as possible, there were quite a few false positives.

Manual inspection. Next, a human expert went over each question to remove non-temporal questions. Some instances that were removed were: *"what is president nixon's first name?"* (wrong interpretation of the ordinal tag), and *"who does nicolas cage play in a christmas carol?"* (Christmas was wrongly tagged as an event). Moreover, the same human expert also verified whether existing gold answers were incorrect or noisy. Redundant answers were normalized to the names of the corresponding Freebase entities. As an example, for the question *"who did libya gain independence from in 1951?"*, the answer *'its independence from Italy'* was removed, and only *'Italy'* was retained. Finally, we had a total of 1,271 cleaned and verified temporal questions in our benchmark.

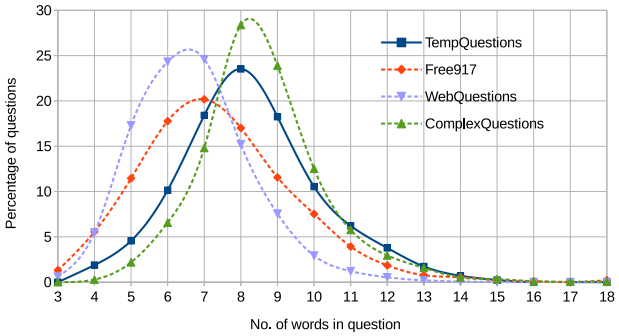

Figure 2: Length distribution in TempQuestions.

4 TempQuestions: ANALYSIS

We now present detailed qualitative and quantitative analyses of our benchmark, giving the reader glimpses into the content. We also highlight scope for research in this direction, by showing below-par performance of state-of-the-art systems on TempQuestions.

4.1 Measurement

First, in Fig. 2, we show how questions in our benchmark are distributed by length (in words), and contrast this with Free917, WQ, and CQ. Questions in our benchmark are between 4 and 15 words long, and the average question length is 8.28 words. The figure shows that a good proportion of questions in TempQuestions are relatively verbose, implying increased parsing difficulty for QA systems. Next, to give readers a feel of the questions in our resource upfront, we present sample questions in Table 1, segmented by the following three dimensions: temporal category, numbers of entities and relations, and question source.

Distribution of question types. We provide a simultaneous breakdown into the four classes of temporal questions, along with the input source, in Table 2. The two key points are: (a) TempQuestions has a good number of questions with implicit temporal expressions (209) and ordinals (155) – both these classes require *additional reasoning* and ranking on part of the QA-system, and thus add a level of difficulty; (b) the total 1,364 is higher than 1,271, showing that there are several questions that belong to more than one category, and are thus quite challenging for current QA systems (like *"who was elected the first governor of virginia in 1776?"*, with both explicit and ordinal tags).

Multiple entities and relations. Table 3 shows the way entities and relations appear in TempQuestions. Stanford NER [19] was used to tag entities, followed by a round of manual inspection (among detected entities, 36% were of type *person*, 30% of type *location*, 17% of type *organization*, and 17% were miscellaneous). Relation tagging was done manually by an expert, as current systems like Saha et al. [24] performing automated relation (fact) extraction are far from perfect. What is noteworthy here is that there are several questions with multiple entities (205) and relations (145)

Table 1: Representative examples from TempQuestions.

Property	Question
Segmentation by question type	
Explicit temporal	*"who won the state of texas in 2008?"* *"what kind of government does iran have after 1979?"*
Implicit temporal	*"who was the president after jfk died?"* *"what team did michael jordan play for after the bulls?"*
Temporal answer	*"what years did the knicks win the championship?"* *"when was the united nations founded?"*
Ordinal contraint	*"who was the first coach of the bucaneers?"* *"who was andy williams second wife?"*
Segmentation by question concepts	
Multi-entity	*"what did france lose to the british in the treaty of paris in 1763?"* *"when was the last time the oakland raiders won the super bowl?"*
Multi-relation	*"who won best supporting actor when alfred junge won best art direction?"* *"what book was written by george orwell and published in 1945?"*
Segmentation by question source	
Free917 [10]	*"when was the airspeed oxford first flown?"* *"in 1981 what award did danny devito win?"*
WQ [7]	*"what was the currency in france before euro?"* *"who is julia roberts married to 2012?"*
CQ [5]	*"who was us president when vietnam war started?"* *"who did michael jordan play for after the bulls?"*

Table 2: Distribution of question types by source. The total is greater than 1,271 as some questions have multiple tags.

Question Tag	Free917	WQ	CQ	Total
Explicit temporal	41	344	222	607
Implicit temporal	3	81	125	209
Temporal answer	88	254	51	393
Ordinal constraint	18	111	26	155
Total	150	790	424	1,364

Table 3: Distribution of entities and relations in questions.

Property	0	1	2	3	Total
#Question entities	5	1,061	201	4	1,271
#Question relations	0	1,126	145	0	1,271

in TempQuestions (examples in Table 1). Multi-relation and multi-entity questions are more difficult for *semantic parsing* [7], and reflect *semantic compositionality*. Most current KB-QA systems are designed for single-entity single-relation questions and would require new techniques to address questions in our resource. An example of a question with no named entity is *"who is the richest person 2015?"*.

Table 4: Performance of state-of-the-art KB-QA systems AQQU and QUINT on TempQuestions and WebQuestions.

Benchmark	Method	Precision	Recall	F-Score
TempQuestions	AQQU	24.6	48.0	27.2
	QUINT	27.3	52.8	30.0
WebQuestions	AQQU	49.8	60.4	49.4
	QUINT	52.1	60.3	51.0

Presence of temporal signals. Finally, we show how temporal signals are distributed: before (49 questions), after (28), overlap (435), and ordinal (156). Signal words may indicate the necessity of question decomposition, rewriting, and separate processing of individual subquestions. As discussed earlier (Sec. 1), this is yet another key challenge that needs to be overcome if QA systems are to answer complex temporal questions. Higher numbers of questions with the *overlap signal* (signifying temporal durations or intervals) point to increased difficulty levels.

4.2 Performance

We now evaluate how two state-of-the-art KB-QA systems AQQU [6] and QUINT [1] perform on TempQuestions, with Freebase as the backend KB. AQQU uses distant supervision and learning-to-rank techniques on several generated SPARQL candidates to find the best query to be executed over the KB, and relies on a set of hand-coded query templates for semantic parsing. QUINT removes this dependence on hand-coded templates for KB-QA, and automatically learns question-query templates solely from user questions paired with their answers. Results are shown in Table 4, where numbers are shown for TempQuestions, and contrasted with WebQuestions (WQ). These systems are designed for standard KB-QA, and thus perform significantly worse on our new benchmark. This is evident from F1-scores of around $27-30\%$, which are $\simeq 50.0\%$ for WQ. This raises the call for better systems tailored for handling temporal intent, while also addressing challenges raised by compositionality and reasoning constraints. Detailed results by question category are shown in Table 5. The sweeping observation is that while all categories reflect poor performance, questions with implicit temporal expressions are particularly challenging.

5 RELATED RESOURCES

Multiple datasets have been proposed for KB-QA, which differ in the underlying KB (DBpedia or Freebase), size (a couple of hundreds to a few thousands), and question phenomena they involve (simple, compositional, and/or questions with conditions, among others) [1, 5, 7, 9, 10, 29, 31]. We refer the reader to Diefenbach et al. [12] for further details.

Benchmarks with complex questions are still ad hoc, and in their infancy. QALD [29, 31] is a series of evaluation campaigns on QA over linked data, and releases datasets every year to evaluate KB-QA systems. Thus far, seven challenges have been presented.

Table 5: Detailed performance of AQQU and QUINT on TempQuestions, segmented by question type.

Type	Explicit temporal			Implicit temporal			Temporal answer			Ordinal constraint		
Method	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
AQQU	27.6	60.7	31.1	12.9	34.9	14.5	26.1	33.5	27.4	28.4	57.4	32.7
QUINT	29.3	60.9	32.6	25.6	54.4	27.0	25.2	38.2	27.3	21.3	54.9	26.1

Questions in QALD cover many interesting phenomena such as aggregation, count, and additional conditions, (for example, *"Which German cities have more than 250000 inhabitants?"*). However, the main shortcoming is the very small size (50 – 250 questions). Recently, Abujabal et al. [1] released 150 questions paired with their answers over Freebase. While all questions in this dataset contain more than one entity/relation, the underlying SPARQL query would still require joining over a single variable only. Questions were collected using a public crawl of WikiAnswers, a large, community-authored corpus of NL questions. The WebQuestions (WQ) [7] and SimpleQuestions [9] datasets contain a majority of simple factoid questions, e.g., *"what language does cuba speak?"*, with a few exceptions. While questions in WQ [7] are only paired with answers, they are improved with SPARQL queries in Bordes et al. [9]. Bao et al. [5] released a new dataset with complex questions paired with their answers over Freebase (2,100 question-answer pairs). The LC-QuAD dataset [28] contains 5,000 questions and their corresponding SPARQL queries over DBpedia. Questions in LC-QuAD exhibit high syntactic and structural variation. These were generated using a set of hand-written templates that verbalize SPARQL queries, which are then corrected and paraphrased by humans.

6 CONCLUSIONS AND FUTURE WORK

We released *TempQuestions*, a new benchmark for temporal question answering, with 1,271 question-answer pairs. With textual answers, the resource is suitable for answering over KBs, free text, or hybrid sources. The questions are accompanied by useful markup tags like question types and signals to allow for detailed system analysis. To facilitate follow-up research, we make results of two state-of-the-art systems on TempQuestions available with our release, and with thorough scrutiny, show that this benchmark is particularly challenging for current KB-QA. As an additional contribution, we provide a concrete definition of a temporal question. Finally, through this benchmark, we call upon the community to build QA-systems that can handle open challenges like temporal intent, compositionality, and constraint-based reasoning.

REFERENCES

[1] Abdalghani Abujabal, Mohamed Yahya, Mirek Riedewald, and Gerhard Weikum. 2017. Automated Template Generation for Question Answering over Knowledge Graphs. In *WWW*.

[2] Eugene Agichtein, David Carmel, Dan Pelleg, Yuval Pinter, and Donna Harman. 2015. Overview of the TREC 2015 LiveQA Track. In *TREC*.

[3] James F. Allen. 1983. Maintaining Knowledge About Temporal Intervals. *Comm. ACM* (1983).

[4] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. 2007. On the value of temporal information in information retrieval. In *ACM SIGIR Forum*.

[5] Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-Based Question Answering with Knowledge Graph. In *COLING*.

[6] Hannah Bast and Elmar Haussmann. 2015. More Accurate Question Answering on Freebase. In *CIKM*.

[7] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *EMNLP*.

[8] Branimir Boguraev, Siddharth Patwardhan, Aditya Kalyanpur, Jennifer Chu-Carroll, and Adam Lally. 2014. Parallel and nested decomposition for factoid questions. *Natural Language Engineering* (2014).

[9] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv* (2015).

[10] Qingqing Cai and Alexander Yates. 2013. Large-scale Semantic Parsing via Schema Matching and Lexicon Extension. In *ACL*.

[11] Angel X. Chang and Christopher D. Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *LREC*.

[12] Dennis Diefenbach, Vanessa Lopez, Kamal Singh, and Pierre Maret. 2017. Core techniques of question answering systems over knowledge bases: A survey. In *Knowledge and Information systems*.

[13] Aditya Kalyanpur et al. 2012. Structured data and inference in DeepQA. *IBM Journal of Research and Development* (2012).

[14] David A. Ferrucci et al. 2012. This is Watson. *IBM Journal of Research and Development* 56 (2012). Issue 3/4.

[15] Aditya Kalyanpur, Siddharth Patwardhan, BK Boguraev, Adam Lally, and Jennifer Chu-Carroll. 2012. Fact-based question decomposition in DeepQA. *IBM Journal of Research and Development* (2012).

[16] Erdal Kuzey, Vinay Setty, Jannik Strötgen, and Gerhard Weikum. 2016. As Time Goes By: Comprehensive Tagging of Textual Phrases with Temporal Scopes. In *WWW*.

[17] Ken Litkowski. 2014. Pattern Dictionary of English Prepositions. In *ACL*.

[18] Ken Litkowski and Orin Hargraves. 2006. Coverage and inheritance in the preposition project. In *SIGSEM*.

[19] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *ACL*.

[20] Donald Metzler, Rosie Jones, Fuchun Peng, and Ruiqiang Zhang. 2009. Improving Search Relevance for Implicitly Temporal Queries. In *SIGIR*.

[21] Anselmo Peñas, Christina Unger, Georgios Paliouras, and Ioannis Kakadiaris. 2015. Overview of the CLEF Question Answering Track 2015. In *CLEF*.

[22] James Pustejovsky, Robert Knippen, Jessica Littman, and Roser Saurí. 2005. Temporal and Event Information in Natural Language Text. In *LREC*.

[23] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL*.

[24] Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for Numerical Open IE. In *ACL*.

[25] Andrea Setzer. 2002. *Temporal information in newswire articles: An annotation scheme and corpus study.* Ph.D. Dissertation. University of Sheffield.

[26] Jannik Strötgen and Michael Gertz. 2015. A Baseline Temporal Tagger for all Languages. In *EMNLP*.

[27] Jannik Strötgen and Michael Gertz. 2016. *Domain-sensitive Temporal Tagging*. Morgan & Claypool Publishers.

[28] Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. LC-QuAD: A Corpus for Complex Question Answering over Knowledge Graphs. In *ISWC*.

[29] Christina Unger, Corina Forascu, Vanessa López, Axel-Cyrille Ngonga Ngomo, Elena Cabrio, Philipp Cimiano, and Sebastian Walter. 2015. Question Answering over Linked Data (QALD-5). In *CLEF*.

[30] Christina Unger, André Freitas, and Philipp Cimiano. 2014. An introduction to question answering over linked data. In *Reasoning Web*.

[31] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th Open Challenge on Question Answering over Linked Data (QALD-7). In *SemWebEval*.

[32] Ellen M. Voorhees. 2010. Reflections on TREC QA. In *CLEF*.

[33] Mohamed Yahya, Klaus Berberich, Shady Elbassuoni, Maya Ramanath, Volker Tresp, and Gerhard Weikum. 2012. Natural language questions for the Web of data. In *EMNLP*.

[34] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In *ACL*.

[35] Pengcheng Yin, Nan Duan, Ben Kao, Jun-Wei Bao, and Ming Zhou. 2015. Answering Questions with Complex Semantic Constraints on Open Knowledge Bases. In *CIKM*.