HOBBIT

Collaborative Project

Holistic Benchmarking of Big Linked Data

Project Number: 688227 **Start Date of Project:** 2015/12/01 **Duration:** 36 months

Deliverable 7.3.2
Second Challenge Results Overview

Dissemination Level	Public
Due Date of Deliverable	Month 36, 30/11/2018
Actual Submission Date	Month 36, 30/11/2018
Work Package	WP7 - Organization of Evaluation Campaigns
Task	T7.1 & T7.3
Type	Report
Approval Status	Final
Version	1.0
Number of Pages	43
Filename	D7.3.2_Second_Challenge_Results_Overview.pdf

Abstract: This deliverable comprises the overview report of the second round of the HOBBIT challenges. The best technologies developed during these challenges are presented and the main outcomes and achievements are summarized.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688227.

History

Version	Date	Reason	Revised by
0.1	05/11/2018	First draft created	Georgios Katsimpras & Vassiliki Rentoumi (NCSR-D)
0.2	11/11/2018	Peer Reviewed	Gayane Sedrakyan (IMINDS)
0.3	12/11/2018	Content Revised	Vassiliki Rentoumi & Georgios Katsimpras (NCSR-D)
1.0	13/11/2018	Final version created	Vassiliki Rentoumi & Georgios Katsimpras (NCSR-D)

Author List

Organization	Name	Contact Information
NCSR-D	Georgios Katsimpras	gkatsibras@iit.demokritos.gr
NCSR-D	Vassiliki Rentoumi	vrentoumi@iit.demokritos.gr
InfAI	René Speck	speck@informatik.uni-leipzig.de
AGT	Martin Strohbach	MStrohbach@agtinternational.com
AGT	Pavel Smirnov	PSmirnov@agtinternational.com

Executive Summary

This deliverable provides an overview report of the outcomes derived from the second round of the HOBBIT challenges[1]. HOBBIT organized five challenges over the third project year. We present the best technologies developed during the five HOBBIT challenges and summarize the main results and achievements.

In particular in 2018, HOBBIT ran the second series of the five HOBBIT challenges originally organized in 2017. The second series of the five HOBBIT challenges were hosted in the same events as their first series' counterparts. The second series of HOBBIT challenges are the following:

- the Mighty Storage Challenge (MOCHA) at the European Semantic Web Conference (ESWC 2018)
- the Open Knowledge Extraction (OKE) challenge at the European Semantic Web Conference (ESWC 2018)
- the Scalable Question Answering (SQA) challenge at the European Semantic Web Conference (ESWC 2018)
- the DEBS Grand Challenge at the ACM International Conference on Distributed and Event-Based Systems (DEBS 2018)
- the HOBBIT Link Discovery Task as part of the Ontology Alignment Evaluation Initiative (OAEI) at the International Semantic Web Conference (ISWC 2018)

The first four challenges ran in June 2018, while the last challenge took place in October 2018.

In addition, HOBBIT launched four open challenges[2] (Open MOCHA, Open OKE, Open SQA and StreaML open challenge) in the last quarter of 2017, where potential participants could submit their systems and join the challenge at virtually any point in time, with periodic cutoffs regulating the announcement of winners. The open challenges aimed at i) boosting participation through the absence of hard deadlines and ii) starting the dissemination of MOCHA, OKE and SQA @ ESWC 2018, as well as DEBS Grand Challenge @ DEBS 2018, early. These open challenges have been terminated as no systems were submitted in time for the first cutoff date and interested teams were directed to the "non-open" counterparts of the challenges.

In the Introduction (Section 1) we provide an overview of the main ideas and goals of this deliverable. In Section 2 we describe the HOBBIT benchmarks that have been used within the HOBBIT challenges. In Section 3 Open Challenges are briefly presented. In Sections 4, 5, 6, 7 and 8, we present a complete overview of the five challenges and their results. In Section 9 we summarize the achievements performed through the second series of the challenges. We also report the feedback provided from the participants.

[1] https://project-hobbit.eu/challenges/
[2] https://project-hobbit.eu/open-challenges/

Abbreviations and Acronyms

WP	Work Package
BLD	Big Linked Data
KPIs	Key Performance Indicators
IM	Instance Matching
OAEI	Ontology Alignment Evaluation Initiative
OKE	Open Knowledge Extraction
MOCHA	Mighty Storage Challenge
QALD	Question Answering over Linked Data
DEBS GC	Debs Grand Challenge
SQA	Scalable Question Answering
SML	Structured Machine Learning
SDK	Software Development Kit
StreaML	Stream Machine Learning
DSB	Data Storage Benchmark
SPBv	Versioning Benchmark
QA	Question Answering

Contents

List of Tables

List of Figures

1 Introduction

This deliverable comprises the results overview report of the second series of HOBBIT challenges. We present the best technologies developed during the second series of the challenges and summarize the main outcomes and achievements. Through the five HOBBIT challenges organized we aimed to measure the performance of technologies for the different steps of the Big Linked Data (BLD) lifecycle. In contrast to existing benchmarks, we managed to provide modular and easily extensible benchmarks for all industry-relevant BLD processing steps that allow to assess whole suites of software that cover more than one step.

In the second series of HOBBIT challenges we took into consideration the valuable feedback received from the first round and we arranged a diverse action plan to facilitate the user participation and experience. Specifically, we updated the platform's documentation[3] with more information, tutorials and articles, on how to use the HOBBIT platform. Also, we created a set of video tutorials[4] and provided extensive implementation guidelines.

Moreover, to achieve a more user-friendly technical environment, we offered several implementation alternatives, including Docker[5] and various APIs[6], as well as ready to work example systems. In addition, we introduced online leader-boards to enhance the continuous testing of participating systems but also to increase the levels of engagement and retention. Besides, significant technical updates, concerning the HOBBIT platform, have been accomplished, which ensured its stability, reliability and led to an effortless execution of challenges.

A description of the HOBBIT benchmarks which participated in the second series of the HOBBIT challenges is presented in Section 2. In section 3 we provide a short summary of the HOBBIT Open Challenges launched in the last quarter of 2017 and mainly aimed at i) boosting participation through the absence of hard deadlines and ii) starting the dissemination of MOCHA, OKE and SQA @ ESWC 2018, as well as DEBS Grand Challenge @ DEBS 2018, early. In Sections 4, 5, 6, 7, 8, we present a complete overview of the results produced from each of the five HOBBIT challenges. In these sections we also show that participating systems demonstrated comparable results to the relevant state-of-the-art systems that participated as baselines in the five HOBBIT challenges. The majority of the participating systems together with their results have been described in papers submitted to the corresponding challenges. The papers were peer-reviewed by experts and the most promising systems were invited to participate. The systems were tested against a concrete set of evaluation criteria (a.k.a Key Performance Indicators (KPIs)) for each challenge that were chosen based on the defined experimental set-up for each challenge task. The evaluation criteria defined for each challenge are described in detail on the corresponding sections of each challenge and are further reported in D 9.2.3 – Annual Public Report of the Third Year.

Additional information on the second series of HOBBIT challenges can be found on the project's website[7], as well as in related deliverables: D7.4 – Second Workshop Proceedings, D7.2.2 – Second Workshop Organization Report and D7.4.2 – Second Challenge Evaluation. D7.4 reports on the proceedings of the challenges, D7.2.2 reports on the organizational aspects of the challenges and D7.4.2 reports on the quantitative and qualitative evaluation of the challenges.

In Section 9 we summarize the results obtained through the second series of the challenges. We also report the feedback provided from the participants.

[3] https://hobbit-project.github.io/
[4] https://www.youtube.com/watch?v=QWPujpc9srM, https://www.youtube.com/watch?v=3oeEyHXVd_4
[5] https://www.docker.com/
[6] https://github.com/hobbit-project
[7] https://project-hobbit.eu/

Table 1: Overview of HOBBIT Benchmarks.

Benchmark	Short Description
Data Extraction for Sensor Data	Evaluate storage solutions that deal with the ingestion of streams of RDF data
Data Extraction for Unstructured Data	Test the performance (runtime and accuracy) of entity recognition and linking frameworks over streams of unstructured data (text)
Linking	Go beyond mere instance matching and check how well tools performs on other types of links (e.g., geospatial links) when faced with large amounts of data
Data Analytics	Study the performance of machine Learning techniques (i.e., performance and runtime) on streams of structured data (e.g., RDF)
Data Storage	Stress test storage solutions for RDF when faced with realistic scenarios such as being the backend of a social network
Versioning	Check how well storage solutions deal with storing evolving data available in several versions and performing queries on and across these different versions
Question Answering	Evaluate the performance of data access solutions that can answer questions in natural language as well as keyword queries on large amounts of data
Faceted Browsing	Test storage solutions w.r.t. their performance as backends of data browsers

2 Benchmarks

Eight benchmarks have been developed within the HOBBIT project. A summary of these benchmarks is given in Table 1. All these benchmarks have been employed in the five HOBBIT challenges which have already been successfully performed. Table 2 shows a mapping between the HOBBIT benchmarks and of the challenges in which the latter participated.

The following subsections aim to give an overview of the technical updates and enhancements ,concerning the aforementioned benchmarks, that were realized during the final year of the project.

Table 2: Mapping of HOBBIT Challenges/Benchmarks.

Event	Benchmark
MOCHA @ESWC 2018 June 3rd to June 7th, 2018	Data Extraction for Sensor Data (INFAI) Data Storage (OpenLink) Versioning (FORTH) Faceted Browsing (IAIS)
OKE @ESWC 2018 June 3rd to June 7th, 2018	Data Extraction for Unstructured Data (INFAI)
SQA @ESWC 2018 June 3rd to June 7th, 2018	Question Answering (IAIS)
Grand Challenge @DEBS 2018 June 25th to June 29th, 2018	Data Analytics (AGT)
Link Discovery Track OM@ISWC 2018 October 8th to October 12th, 2018	Linking (FORTH)

2.1 Data Extraction for Sensor Data

The final version of the benchmark presents more functionalities compared to the previous version which are listed below:

- Introduced dependencies between data resources by creating a network among them.
- Tested the ingestion performance of a storage system by deploying datasets that vary in volume (size of statements and time stamps).
- Used dilatation factors based on the real time differences between the various statement, in order to benchmark the system within a particular time interval.
- Used streaming data from multiple resources.
- Varied the queries to cover different proportions of inserted statements.

The benchmark was used in the MOCHA 2018 challenge[8] and was able to unveil the limitations of existing systems pertaining to data ingestion. The outcome of the the benchmark indicated that Virtuoso Commercial 8.0 performed better. For a detailed analysis of the results refer to section 4.

More details about the Data Acquisition benchmark can be found in D3.1.2 Data Extraction Benchmark for Sensor Data.

[8] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/

2.2 Data Extraction for Unstructured Data

In the final version of the benchmark we performed the following extensions:

- Extended the task generator.
  - Benchmark for property extraction between entities.
  - Benchmark for knowledge extraction.
- Extended the data generator for these new benchmarks.
- Integration of key performance indicators for this new benchmarks.
- Run the benchmarks and co-organized the Open Knowledge Extraction Challenge at ESWC 2018[9].

The benchmark was used in the OKE 2018 challenge[10] which was held at ESWC 2018. The benchmark successfully attracted participants from both academia and industry, and thus fostered the collaboration between the two communities. For a detailed analysis of the results refer to section 6.

More details about the Knowledge Extraction benchmark can be found in D3.2.2 Second Version of the Data Extraction Benchmark for Unstructured Data.

2.3 Linking

During the third year of the HOBBIT project we completed the implementation/extension of Spatial Benchmark Generator SPgen which can be used to test the performance of systems that deal with topological relations proposed by the state of the art DE-9IM (Dimensionally Extended nine-Intersection Model). This benchmark generator implements all topological relations of DE-9IM between *LineStrings* and *Polygons* in the two-dimensional space. SPgen follows the chokepoint-based approach for benchmark design, i.e., it focuses on the technical difficulties of existing systems and implements tests that address those difficulties and push systems to resolve them. Specifically, the chokepoints are: a) Scalability: produce datasets large enough to stress the systems under test, b)Output quality: compute precision, recall and f-measure, and c) Time performance: measure the time the systems need to return the results. This benchmark is generic in the sense that it is *schema agnostic*: it can operate with any datasets that contain trajectories, a trajectory being a set of points or a set of longitude, latitude pairs. Also, we used the TomTom datasets provided in the context of project HOBBIT.

The Linking benchmark was successfully used in the Ontology Alignment Evaluation Initiative (OAEI) that was held at ISWC 2018. More specifically we organised the *HOBBIT Link Discovery Track* with two tasks, the *Linking* and *Spatial* each running the corresponding HOBBIT benchmarks. Three systems participated and evaluated in the benchmark. The overall results indicated that there is still room for further improvement. For a detailed analysis of the results refer to section 7.

More details about the Linking benchmark can be found in D4.1.2 Second Version of the Linking Benchmark.

[9] https://project-hobbit.eu/challenges/oke2018-challenge-eswc-2018/
[10] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/

2.4 Data Analytics

The benchmark was used in the DEBS Grand Challenge 2018 [11]. In the final year, the following subtasks have been completed:

- SML benchmark was extended to V2.0 and integrated into the HOBBIT platform. The benchmark is focused on assessing the performance of various stream processing analytical systems by measuring the accuracy and the speed of their implementations.
- Evaluation of the developed benchmark was done via organization of the DEBS Grand Challenge 2018 on the HOBBIT platform[12].

The benchmark was used in the DEBS Grand Challenge 2018 [13]. Nine systems participated in the benchmark. The results indicated that reliable predictions regarding naval transportation ca be achieved. For a detailed analysis of the results refer to section 8.

More details about the SML benchmark can be found in D4.2.2 Second Version of the Data Analytics Benchmark.

2.5 Data Storage

For the final version of the benchmark we finished a considerable number of tasks, from which we emphasize the following:

- Implementation of the second version of the benchmark (DSB v2.0): Unlike the first version that was single threaded, this one is multi-threaded, allowing us to test the concurrency of the systems using a real workload that is driven by updates.
- Mutual comparisons between different scales, and with SQL implementation of the benchmark.
- Evaluation of the benchmark in MOCHA Challenge held at ESWC 2018[14]

The benchmark was used in the MOCHA 2018 challenge[15]. Five systems participated in the benchmark and only two systems managed to completed, which indicates a rather demanding task. In fact, Virtuoso v8.0 performed best. For a detailed analysis of the results refer to section 4.

More details about the Data Storage benchmark can be found in D5.1.2 Second Version of the Data Storage Benchmark.

2.6 Versioning

In the third year of the project we performed the following tasks:

- Finalized the implementation of the data generator for the **SPBv** v2.0 so that it supports addition and deletion of triples and can send data in different forms.

[11] https://project-hobbit.eu/challenges/debs2018-grand-challenge/
[12] https://project-hobbit.eu/challenges/debs2018-grand-challenge/
[13] https://project-hobbit.eu/challenges/debs2018-grand-challenge/
[14] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/
[15] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/

- Finalized the query workload so that it is based on real DBpedia query logs.
- Implemented all of necessary system adapters that are be used to integrate versioning systems in the HOBBIT platform.
- Evaluated the benchmark in MOCHA Challenge held at ESWC 2018[16]

The benchmark was used in the MOCHA 2018 challenge[17]. All systems except one, managed to complete the benchmark. The best performance was achieved by Virtuoso v8.0. For a detailed analysis of the results refer to section 4.

More details about the Versioning benchmark can be found in D5.2.2 Second Version of the Versioning Benchmark.

2.7 Question Answering

The benchmark was used as a task in the SQA challenge held at ESWC 2018[18].

For the final version of the benchmark we performed the following actions:

- Refreshed the benchmark datasets and improved user and technical documentation.
- Improved the multilingual benchmark, by leveraging localized versions of DBpedia.
- Improved the large scale benchmark, by including complex and more varied questions.
- Participated in the SQA challenge in ESWC2018[19].

The benchmark was used as a task in the SQA challenge held at ESWC 2018[20]. Three systems participated in the benchmark and were evaluated against four measures. For a detailed analysis of the results refer to section 5.

More details about the Question Answering benchmark can be found in D6.1.2 Second Version of the Question Answering Benchmark.

2.8 Faceted browsing

For the final version of the Faceted Browsing benchmark we performed some extensions which are stated below:

- Analyzed existing faceted browsing engines for the types of SPARQL queries they generate and incorporated selected ones into the benchmark.
- Added support for composing faceted browsing benchmarks from high-level user interaction events, such as panning a map or enabling/disabling certain constraints. This increased the flexibility of the system by enabling benchmarking of different strategies that convert interactions to a (set of) SPARQL queries.

[16] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/
[17] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/
[18] https://project-hobbit.eu/challenges/sqa-challenge-eswc-2018/
[19] https://project-hobbit.eu/challenges/sqa-challenge-eswc-2018/
[20] https://project-hobbit.eu/challenges/sqa-challenge-eswc-2018/

- For all newly introduced queries / query types, we classified them according to the identified choke-points.
- Integrated the revised benchmark into the HOBBIT platform.
- Participated in the MOCHA challenge held at ESWC 2018[21].

The benchmark was used in the MOCHA 2018 challenge[22]. All systems successfully participated in the benchmark and presented similar performance. However, Virtuoso Open Source was the winner. For a detailed analysis of the results refer to section 4.

More details about the Faceted Browsing benchmark can be found in D6.2.2 Second Version of the Faceted Browsing Benchmark.

3 Open Challenges

Apart from the second series of HOBBIT challenges, HOBBIT launched four open challenges[23] (Open MOCHA, Open OKE, Open SQA and StreaML open challenge) in the last quarter of 2017, where potential participants could submit their systems and join the challenge at virtually any point in time, with periodic cutoffs regulating the announcement of winners. The open challenges aimed at i) boosting participation through the absence of hard deadlines and ii) starting the dissemination of MOCHA, OKE and SQA @ ESWC 2018, as well as DEBS Grand Challenge @ DEBS 2018, early.

Open MOCHA[24] consisted of the same tasks described in Section 4. Open OKE[25] consisted of three tasks; i) Focused Named Entity Identification and Linking, ii) Broader Named Entity Identification and Linking and iii) Focused Musical Named Entity Recognition and Linking. These tasks were part of the OKE challenge organized at ESWC 2017. Open SQA[26] challenge was also launched as an open challenge in December 2017 and consisted of the same tasks described in Section 5.1. The StreaML (Stream Machine Learning) Open Challenge was launched as an open challenge in February 2018. StreaML[27] Open Challenge comprised the offspring of DEBS GC 2017 as it reused the data set and the task description of DEBS GC 2017.

For all four Open Challenges we also introduced the concept of challenge leaderboards, where a leaderboard of the participating systems was implemented as part of the HOBBIT platform so that participants could continuously monitor their systems' performance against other competing systems Despite the dissemination efforts made to promote the Open Challenges as described in D7.1.3 and in D7.2.2 the open challenges have been terminated as no systems were submitted in time for the first cutoff date and interested teams were directed to the "non-open" counterparts of the challenges.

[21] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/
[22] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/
[23] https://project-hobbit.eu/open-challenges/
[24] https://project-hobbit.eu/open-challenges/mocha-open-challenge/
[25] https://project-hobbit.eu/open-challenges/oke-open-challenge/
[26] https://project-hobbit.eu/open-challenges/sqa-open-challenge
[27] https://project-hobbit.eu/open-challenges/streaml-open-challenge/

4 MOCHA Challenge Results Overview

4.1 Definition of Tasks

The aim of the Mighty Storage Challenge was to test the performance of solutions for SPARQL processing in aspects that are relevant for modern applications. These include ingesting data, answering queries on large datasets and serving as backend for applications driven by Linked Data. The proposed challenge tests the systems against data derived from real applications and with realistic loads. An emphasis was put on dealing with changing data in form of streams or updates. We designed the challenge to encompass the following tasks:

- Task 1: Sensor Streams Benchmark, that measured how well systems can ingest streams of RDF data.
- Task 2: Data Storage Benchmark, that measured how data stores perform with different types of queries.
- Task 3: Versioning RDF Data Benchmark, that measured how well versioning and archiving systems for Linked Data perform when they store multiple versions of large data sets. This task is introduced for the first time this year.
- Task 4: Faceted Browsing Benchmark, that measured for how well solutions support applications that need browsing through large data sets.

MOCHA was successfully held at ESWC 2018[28]. Systems participating in MOCHA as well as their results were presented to the public in a dedicated workshop session. The papers describing the systems of the MOCHA challenge were peer-reviewed by experts and the most promising systems were invited to participate in the challenge. After the reviewing process two papers were accepted for publication.

The MOCHA challenge papers have been published by Springer on the proceedings volume *Buscaldi, Davide, Gangemi, Aldo, Reforgiato Recupero, Diego (Eds.), Semantic Web Challenges, Communications in Computer and Information Science, vol. 927, 2018*[29]. This volume contains the papers of all challenges that were organized at the ESWC 2018 conference. Specifically, in [9] the challenge organizers presented an overview of the challenge results and baseline systems, while in [25] and [11] the challenge participants described their systems.

4.2 Participating Systems

The MOCHA2018 challenge ran on 18th May, 2018 and its results were presented during the ESWC 2018 closing ceremony. In total, seven systems participated in MOCHA challenge, five out of which participated in all tasks, while two additional systems participated only in Task 3. The latter additional systems are OSTRICH developed by IDLab, Department of Electronics and Information Systems, Ghent University - imec and R43ples.

- Virtuoso v8.0, developed by OpenLink Software.[30] It is a modern enterprise-grade solution for data access, integration, and relational database management.

[28] https://project-hobbit.eu/challenges/mighty-storage-challenge2018/
[29] https://doi.org/10.1007/978-3-030-00072-1
[30] https://virtuoso.openlinksw.com/

- Baseline Virtuoso Open Source, developed also by OpenLink Software and is the open source version of Virtuoso. [31]
- Blazegraph [32] that is an ultra-scalable, high-performance graph database with support for the Blueprints and RDF/SPARQL APIs.
- Graph DB Free 8.5 is a family of highly-efficient, robust and scalable RDF databases. It streamlines the load and use of linked data cloud datasets GraphDB Free implements the RDF4J [33]
- Apache Jena Fuseki 3.6.0 is a SPARQL server. It provides the SPARQL 1.1 protocols for query and update as well as the SPARQL Graph Store protocol. [34]

4.3 Results & Achievements

4.3.1 Task 1

4.3.1.1 KPIs for Task 1

KPIs are divided into two categories: Correctness and Efficiency. Correctness is measured by calculating Recall, Precision and F-Measure. First, the INSERT queries created by each data generator are sent into a triple store by bulk load. After a stream of INSERT queries is performed against the triple store, a SELECT query is conducted by the corresponding data generator. In Information Retrieval, Recall and Precision were used as relevance measurements and were defined in terms of retrieved results and relevant results for a single query. For our set of experiments, the relevant results for each SELECT query were created prior to the system benchmarking by inserting and querying an instance of the Jena TDB storage solution. Additionally, we compute Macro and Micro Average Precision, Recall and F-measure to measure the overall performance of the system. Efficiency is measured by using the following metrics: (1) Triples-Per-Second that measures the triples per second as a fraction of the total number of triples that were inserted during a stream. This is divided by the total time needed for those triples to be inserted (begin point of SELECT query - begin point of the first INSERT query of the stream). We provided the maximum value of the triples per second of the whole benchmark. The maximum triples per second value was calculated as the triples per second value of the last stream with Recall value equal to 1. (2) Average Delay of Tasks as the task delay between the time stamp that the SELECT query (task) was sent to the system and the time stamp that the results were send to HOBBIT's storage for evaluation. We report both the average delay of each task and the average delay of task for the whole experiment.

4.3.1.2 Experiment set-up

For the MOCHA2018 experiments, all parameters were set to their default values, except from the following:

- **Population of generated data** = 10,000
- **Number of data generators - agents** = 4
- **Number of insert queries per stream** = 20

[31] https://github.com/openlink/virtuoso-opensource
[32] https://www.blazegraph.com/
[33] http://rdf4j.org/about/
[34] https://jena.apache.org/documentation/fuseki2/

4.3.1.3 Results for Task 1

Figure 1 illustrates the correctness KPIs for our baseline systems and Virtuoso Commercial 8.0 under the MOCHA2018 configuration. Beginning with the system that had the worse overall performance, Blazegraph, we observe that all KPIs, apart from Micro-Average-Precision, receive the lowest values compared to the other systems. The high value obtained in Micro-Average-Precision indicates that Blazegraph is able to retrieve correct results for the set of SELECT queries that it answered, but its low Macro-Average-Precision value shows that the set of SELECT queries that it managed to retrieve results for was exceptionally low. To continue with, we observe a very similar behavior between Apache Jena Fuseki 3.6.0 and GraphDB Free 8.5. Both systems achieve a high performance in Micro and Macro-Average-Recall, showing that they are able to retrieve all expected results in most SELECT queries. In contrast, their low values in Micro and Macro-Average-Precision indicate that both systems tend to include large sets of irrelevant answers to the query. Finally the two remaining systems, OpenLink Virtuoso and Virtuoso Commercial 8.0 share a similar behavior among the results: they both achieve high Mirco and Macro-Average-Recall and Precision, which shows their superior ability to ingest and retrieve triples with high accuracy.

Regarding their maximum Triples-Per-Second KPI, Figure 3b shows that Blazegraph achieves the highest value among the other systems. This observation shows that Blazegraph is able to retrieve correct results for a large amount of triples that was inserted in a very short period of time. However, based on Figure 1, since both Micro and Macro-Average-Recall values are low, we can assume that this situation does not occur very often while using Blazegraph as a triple storage solution.

Regarding the efficiency KPIs, Figure 3a shows that both OpenLink Virtuoso and Virtuoso Commercial 8.0 require on average a minute amount of time to process the SELECT queries and retrieve correct results. For Apache Jena Fuseki 3.6.0 and GraphDB Free 8.5, the ability to retrieve high Recall performance comes at the cost of efficiency, since both systems have quite high average delay of tasks. Finally, Blazegraph has a low response time compared to the last two systems, but not insignificant as OpenLink Virtuoso and Virtuoso Commercial 8.0.

To conclude, both OpenLink Virtuoso and Virtuoso Commercial 8.0 had received the same Macro-Average-F-Measure value (approx. 0.86), so in order to announce the winner for Task 1 of MOCHA2018, we consider the second KPI in order: Macro-Average-Recall. OpenLink Virtuoso and Virtuoso Commercial 8.0 received 0.88 and 0.92 resp. This clearly indicates that Virtuoso Commercial 8.0 is the winner of Task 1 of MOCHA2018.

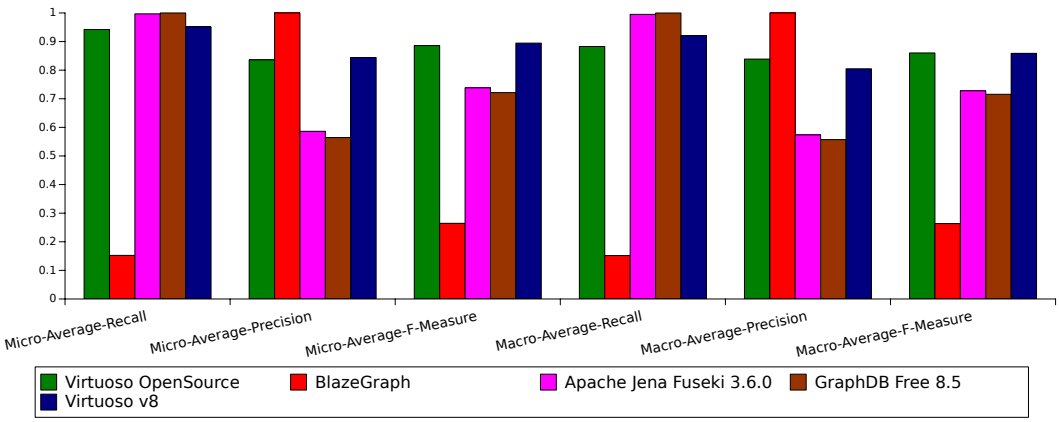
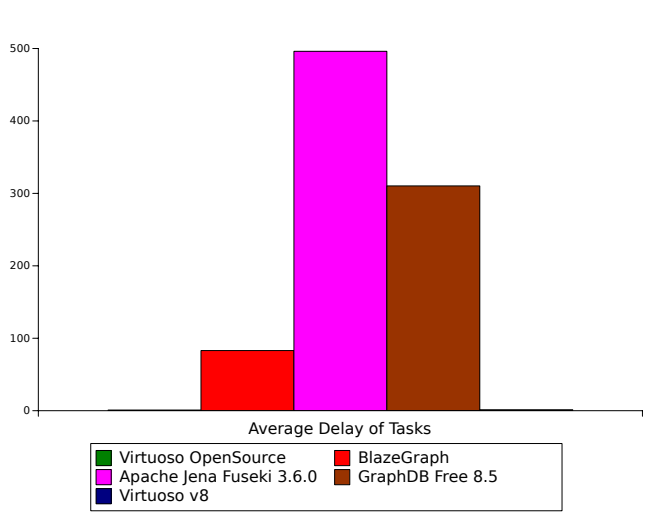
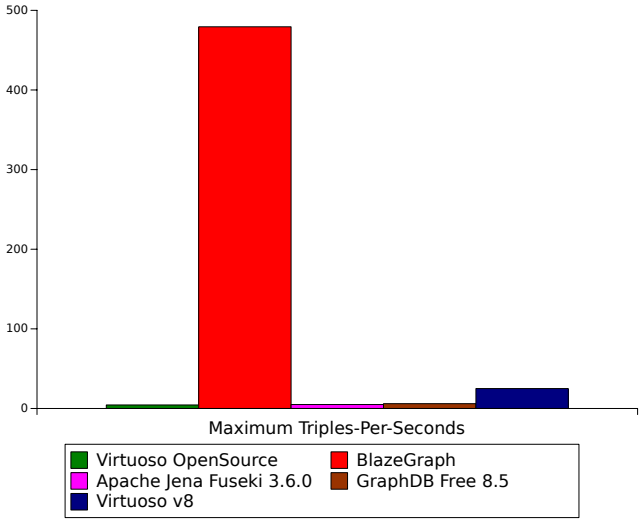


Figure 1: Micro-Average-Recall, Micro-Average-Precision, Micro-Average-F-Measure, Macro-Average-Recall, Macro-Average-Precision, Macro-Average-F-Measure of all systems for Task 1



(a) Average Delay of tasks of all systems for Task 1



(b) Maximum Triples-per-Second of all systems for Task 1

4.3.2 Task 2

4.3.2.1 KPIs for Task 2

The key performance indicators for the Data Storage benchmark are rather simple and cover both efficiency and correctness:

- **Bulk Loading Time:** The total time in milliseconds needed for the initial bulk loading of the dataset.
- **Average Task Execution Time:** The average SPARQL query execution time in milliseconds.
- **Average Task Execution Time Per Query Type:** The average SPARQL query execution time per query type in milliseconds.
- **Query failures:** The number of SPARQL SELECT queries whose result set is different (in any aspect) from the result set obtained from the triple store used as a gold standard.
- **Throughput:** The average number of tasks (queries) executed per second.

4.3.2.2 Experiment set-up.

The benchmark had a defined maximum time for the experiment of 3 hours. The DSB parameters used in the challenge were the following:

- **Number of operations** = 15000
- **Scale factor** = 30
- **Seed** = 100
- **Time compression ratio (TCR)** = 0.5

- **Sequential Tasks** = false
- **Warm-up Percent** = 20

The scale factor parameter defines the size of dataset. Its value of 30 means 1.4 billion triple dataset. After bulk loading it, there were 15000 SPARQL queries (INSERT and SELECT) executed against the system under test. One fifth of them was used for warm-up, while the rest of the queries were evaluated. The value 0.5 of TCR parameter implies about 17 queries per second.

4.3.2.3 Results for Task 2

Unfortunately, out of the five systems which participated in the task, only two managed to complete the experiment in the requested time. Blazegraph, GraphDB and Jena Fuseki exhibited a timeout during the bulk loading phase, while the achieved KPIs for Virtuoso v8.0 and Virtuoso Open Source (VOS) are given below.

Based on the results from the main KPIs (Figure 2), the winning system for the task was Virtuoso 8.0 Commercial Edition by OpenLink Software. In the domain of efficiency, it is 32% faster than VOS regarding average query execution times, and also 6% faster in data loading. In the domain of correctness, Virtuoso v8.0 made 17 query failures compared to the 4 made by VOS; however, having in mind the fact that VOS was used as a golden standard for calculating the expected query results, this KPI is biased, and should not be considered as a weakness of the commercial edition of Virtuoso.

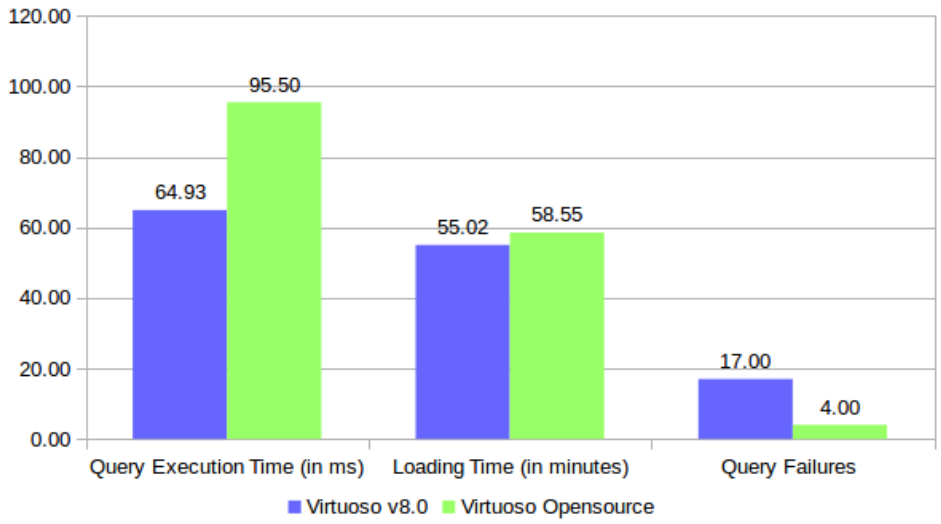


Figure 2: Task 2: Main KPIs

The rest of the KPIs (Figure 3) show where the most important advantage comes from. For the complex query types, Virtuoso v8.0 is much faster than its open source counterpart, while the differences in the short look-ups and updates (SPARQL INSERT queries) are negligible.

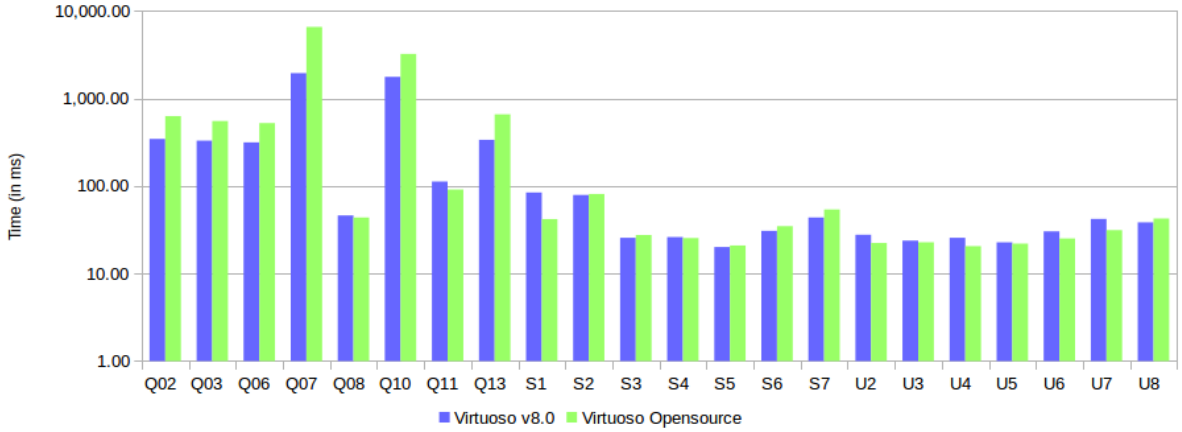


Figure 3: Task 2: Average Query Execution Time per Query Type

4.3.3 Task 3

4.3.3.1 KPIs for Task 3

Fevaluates the correctness and performance of the system under test through the following *Key Performance Indicators (KPIs)*:

- **Query failures**: The number of queries that failed to execute. Failure refers to the fact that the system under test return a result set (RS_{sys}) that is not equal to the expected one (RS_{exp}). This means that i) RS_{sys} has equal size to RS_{exp} and ii) every row in RS_{sys} has one matching row in RS_{exp}, and vice versa (a row is only matched once). If the size of the result set is larger than 50.000 rows, for time saving, only condition i) is checked.
- **Initial version ingestion speed** (triples/second): the total triples of the initial version that can be loaded per second. We distinguish this from the ingestion speed of the other versions because the loading of the initial version greatly differs in relation to the loading of the following ones, where underlying processes such as, computing deltas, reconstructing versions, storing duplicate information between versions etc., may take place.
- **Applied changes speed** (changes/second): tries to quantify the overhead of such underlying processes that take place when a set of changes is applied to a previous version. To do so, this KPI measures the average number of changes that could be stored by the benchmarked systems per second after the loading of all new versions.
- **Storage space cost** (MB): This KPI measures the total storage space required to store all versions measured in MB.
- **Average Query Execution Time** (ms): The average execution time, in milliseconds for each one of the eight versioning query types.
- **Throughput** (queries/second): The execution rate per second for all queries.

4.3.3.2 Experiment set-up

Since spvb gives the ability to the systems under test to retrieve the data of each version as Independent Copies (IC), Change-Sets (CS) or both as IC and CS, three sub-tasks defined in the context of the challenge which only differed in terms of the "generated data form" configuration parameter. So, each participant was able to submit his/her system in the correct sub-task according to the implemented versioning strategy. In particular all the systems benchmarked using the following common parameters:

- **A seed for data generation**: 100
- **Initial version size** (in triples): 200000
- **Number of versions**: 5
- **Version Deletion Ratio** (%): 10
- **Version Insertion Ratio** (%): 15

The experiment timeout was set to 1 hour for all systems.

4.3.3.3 Results for Task 3

All the systems except of the R43ples one, were managed to be tested. The latest, did not manage to load the data and answer all the queries in the defined timeout of 1 hour.

In order to be able to decide who is the winner for the Task 3 of the challenge, we combined the results of the four most important KPIs and calculated a *final score* that ranges from 0 to 1. Note here that due to reliability issues mentioned earlier, the "Storage space cost" KPI excluded from the final score formula. So, to compute the final score, we assigned weights to those KPIs, whose sum equals to 1. The four KPIs (in order of importance) along with the assigned weights are shown in Table 4.

Order	KPI	Weight
1	Throughput	0.4
2	Queries failed	0.3
3	Initial version ingestion speed	0.15
4	Applied changes speed	0.15

Table 4: Weights for the four most important KPIs

Next, we applied feature scaling[35] to normalize the results of the *Throughput*, *Initial version ingestion speed* and *Applied changes speed* by using the following formula:

$x' = \frac{x - min(x)}{max(x) - min(x)}$, where x is the original value and x' is the normalized value.

[35] Bring all results into the range of [0,1]

Regarding the *Queries failed* KPI, since the lower is the result, the better the system performs, the aforementioned formula applied on the percentage of succeeded queries and not to the number of queries that failed to be executed.

Having all the results normalized in the range of $[0-1]$ and the weights of each KPI, we computed the final scores as the sum of the weighted normalized results. As shown in Figure 4 VIRTUOSO v8.0 was the system that performed better.

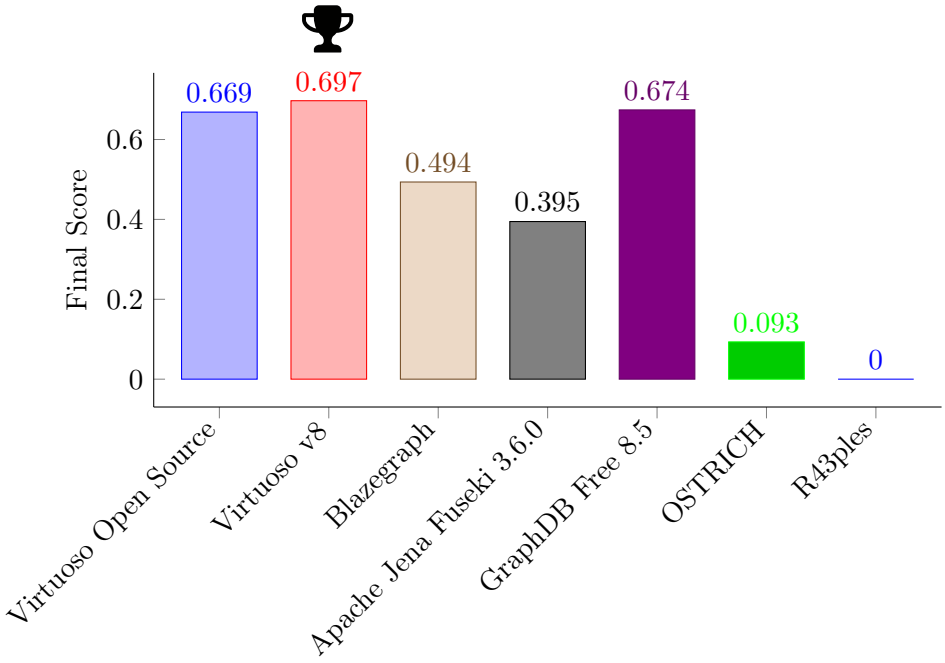


Figure 4: Final scores for all systems

4.3.4 Task 4

4.3.4.1 KPIs for Task 4

The benchmark tracks the conventional indicators for correctness and efficiency, namely precision, recall, F-Measure and query-per-second rate. These measurements are recorded for the individual choke-points as well as the overall benchmark run. For ranking the systems, we were only interested in the latter:

- **Correctness** The conformance of SPARQL query result sets with pre-computed reference results.
- **Performance** The average number of faceted browsing queries per second.

4.3.4.2 Experiment set-up

The dataset and SPARQL queries generated by the faceted browsing benchmark for the challenge are fully determined by the following settings:

- **Random Seed** = 111

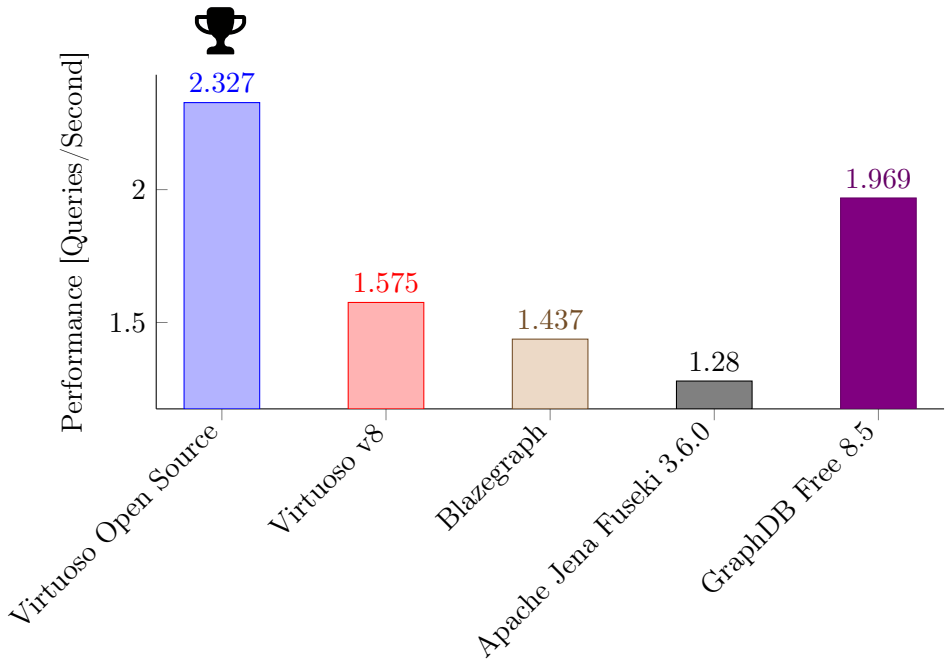


Figure 5: Task 4 Faceted Browsing Benchmark Results

- **Time range** = 0 - 977616000000
- **Number of stops / routes / connections** = 3000 / 3000 / 230000
- **Route length range** = 10 - 50
- **Region size / Cell Density** = 2000 x 2000 / 200
- **Delay Change / Route Choice Power** = 0.02 / 1.3

4.3.4.3 Results for Task 4

Figure 5 shows the performance results of the faceted browsing benchmark. All participating systems were capable of successfully loading the data and executing the queries. Error values are obtained as follows: For COUNT-based queries, it is the difference in the counts of the actual and reference result. For all other (SELECT) queries, the size of the symmetric difference of the RDF terms mentioned in the reference and actual result sets is taken. While no differences were observer during testing the benchmark with Virtuoso Open Source and Apache Jena Fuseki 3.6.0, surprisingly, very minor ones were observed in the challenge runs. Yet, as all systems nonetheless achieved an f-measure score of >99% we did not discriminate by correctness and ranked them by performance only, depicted in Figure 5. The winner of the faceted browsing challenge is Virtuoso Open Source with an average rate of 2.3 query executions per second.

4.4 Conclusions

The goal of MOCHA2018 was to test the performance of storage solutions by measuring the systems performance in four different aspects. We benchmarked and evaluated six triple stores and

presented a detailed overview and analysis of our experimental set-up, KPIs and results. Overall, our results suggest that the clear winner of MOCHA2018 is Virtuoso v8.0. As part of our future work, we will benchmark more triple storage solutions by scaling over the volume and velocity of the RDF data and use a diverse number of datasets to test the scalability of our approaches.

5 SQA Challenge Results Overview

5.1 Definition of Tasks

The Scalable Question Answering (SQA) challenge stems from the long-standing Question Answering over Linked Data (QALD)[36] challenge series, aiming at providing an up-to-date benchmark for assessing and comparing state-of-the-art-systems that mediate between a large volume of users, expressing their information needs in natural language, and RDF data. It thus targets all researchers and practitioners working on querying Linked Data, natural language processing for question answering, information retrieval and related topics. The main goal is to gain insights into the strengths and shortcomings of different approaches and into possible solutions for coping with the increasing volume of requests that QA systems have to process, as well as with the large, heterogeneous and distributed nature of Semantic Web data.

The key difficulty for Scalable Question Answering over Linked Data is in the need to translate a user's information request into such a form that it can be efficiently evaluated using standard Semantic Web query processing and inferencing techniques. Therefore, the main task of the SQA challenge was the following:

> Given an RDF dataset and and a large volume of natural language questions, return the correct answers (or SPARQL queries that retrieves those answers).

Successful approaches to Question Answering are able to scale up to big data volumes, handle a vast amount of questions and accelerate the question answering process (e.g. by parallelization), so that the highest possible number of questions can be answered as accurately as possible in the shortest time. The focus of this task is to withstand the confrontation of the large data volume while returning correct answers for as many questions as possible.

SQA was organized in conjunction with the ESWC 2018 conference, where systems participating in the challenge and their results were presented to the public in a dedicated workshop session. The papers describing the systems of the SQA challenge were peer-reviewed by experts and the most promising systems were invited to participate in the challenge.

The SQA challenge papers have been published by Springer on the proceedings volume *Buscaldi, Davide, Gangemi, Aldo, Reforgiato Recupero, Diego (Eds.), Semantic Web Challenges, Communications in Computer and Information Science, vol. 927, 2018*[37]. Specifically, in [16] the challenge organizers presented an overview of the challenge results and baseline systems, while in [7], [27] and [18] the challenge participants described their systems.

[36] http://www.sc.cit-ec.uni-bielefeld.de/qald/
[37] https://doi.org/10.1007/978-3-030-00072-1

5.2 Participating Systems

Three systems participated in the SQA challenge. We provide here brief descriptions, please refer to the respective full papers (where they exist) for more detailed explanations.

WDAqua-core1 [7] is built on a rule-based system using a combinatorial approach to generate SPARQL queries from natural language questions. In particular, the system abstracts from the specific syntax of the question and relies on the semantics encoded in the underlying knowledge base. It can answer questions on a number of Knowledge Bases, in different languages, and does not require training.

LAMA [18] was originally developed for QA in French. It was extended for the English language and modified to decompose complex queries, with the aim of improving performance on such queries and reduce response times. The question type (e.g. *Boolean* or *Entity*) is classified by pattern matching and processes by the relevant component to extract entities and properties. Complex questions are decomposed in simple queries by keyword matching.

GQA [27], the Grammatical Question Answering system, is built around a functional programming language with categorial grammar formalism. The question is parsed according to the grammar and the best parse is selected. Finally, this is decomposed into its elements, starting from the innermost, while requests are sent to DBpedia to find the corresponding values and the final answers.

5.3 Evaluation Metrics

The SQA challenge provides an automatic evaluation tool (based on GERBIL QA [26] and integrated into the HOBBIT platform)[38,39] that is open source and available for everyone to re-use. The HOBBIT platform also incorporates a leaderboard feature to facilitate comparable evaluation and result display of systems participating in challenges. The tool is also accessible online, so that participants were able to upload their systems as Docker images and check their (and others') performance via a webservice. The ranking of the systems was based on the usual KPIs (precision, recall and F measure) plus a "response power" measure, which is also taking into account the ability of the systems to cope with high volume demand without failure. The response power is the harmonic mean of three measures: precision, recall and the ratio between processed questions (an empty answer is considered as processed, a missing answer is considered as unprocessed) and total number of questions sent to the system. The final ranking was on

1. response power
2. precision
3. recall
4. F measure

[38] http://gerbil-qa.aksw.org/gerbil/
[39] http://master.project-hobbit.eu/

in that order. For each system q, precision, recall and response power are computed as follows:

$$\text{precision}(q) = \frac{\text{number of correct system answers for } q}{\text{number of system answers for } q}$$

$$\text{recall}(q) = \frac{\text{number of correct system answers for } q}{\text{number of gold standard answers for } q}$$

$$\text{response power}(q) = \frac{3}{\frac{1}{\text{precision}(q)} + \frac{1}{\text{recall}(q)} + \frac{\text{processed}}{\text{submitted}}}$$

The benchmark sends to the QA system one question at the start, two more questions after one minute and continues to send *n+1* new questions after *n* minutes. One minute after the last set of questions is dispatched, the benchmarks closes and the evaluation is generated as explained above. The 1830 questions in the dataset allow the running of the benchmark for one hour but for the SQA challenge we limited to 30 sets of questions.

5.4 Results & Achievements

The experimental data for the SQA challenge over the test dataset can be found at the following URLs:

- WDAqua: https://master.project-hobbit.eu/experiments/1527792517766,
- LAMA: https://master.project-hobbit.eu/experiments/1528210879939,
- GQA): https://master.project-hobbit.eu/experiments/1528283915360.

By providing human- and machine-readable experimental URIs, we provide deeper insights and repeatable experiment setups.

Note also that the numbers reported here may differ from the publications of the participants, as these figures were not available at the time of participant paper submission.

Test	WDAqua	LAMA	GQA
Response Power	0.472	0.019	0.028
Micro Precision	0.237	0.054	0.216
Micro Recall	0.055	0.001	0.002
Micro F1-measure	0.089	0.001	0.004
Macro Precision	0.367	0.010	0.018
Macro Recall	0.380	0.016	0.019
Macro F1-measure	0.361	0.011	0.019

Table 5: Overview of SQA results.

5.5 Conclusions

The Scalable Question Answering over Linked Data challenge introduced a new metric (Response Power) to evaluate the capability of a QA system to perform under increasing stress. For the first time, it also partially employed complex and non-well-formed natural language questions, to make the challenge even closer to real use scenarios. In this challenge, we also kept last year underlying evaluation platform (HOBBIT) based on docker, to account for the need for comparable experiments via webservices. This introduces an entrance threshold for participating teams but ensures a long term comparability of the system performance and a fair and open challenge. Finally, we offered leader boards prior to the actual challenge in order to allow participants to see their performance in comparison with the others. Overall, we are confident that the HOBBIT platform will be able to provide QA challenge support for a long time, making comparable and repeatable question answering research possible.

6 OKE Challenge Results Overview

6.1 Definition of Tasks

The Open Knowledge Extraction (OKE) challenge has the ambition to provide a reference framework for research on Knowledge Extraction from text for the semantic web by re-defining a number of tasks (typically from information and knowledge extraction), taking into account specific semantic web requirements. It thus invites researchers and practitioners from academia as well as industry to compete to the aim of pushing further the state of the art of knowledge extraction from text for the semantic web. OKE in 2018 focused on the following four tasks:

- Task 1: Focused Named Entity Identification and Linking
- Task 2: Broader Named Entity Identification and Linking
- Task 3: Relation Extraction
- Task 4: Knowledge Extraction

In more detail, *Task 1* aims at the identification and linking of entities of a given, limited set of entity types. It is a two-step process, including the identification of named entities (**Recognition step**) and the linking of those entities to resources in DBpedia (**D2KB step**). The task is limited to a subset of three DBpedia ontology types; `Person`, `Place` and `Organisation`. *Task 2* extends Task 1 to more DBpedia ontology types. Besides the three types mentioned above, a competing system might have to identify other types of entities and to link these entities as well. Table 6 provides a complete list of types that are considered in this task. Example subtypes of the corresponding class, if they exist, as well as example instances are also shown. *Task 3* consists of two subtasks; (a) focused musical named entity identification and classification and (b) linking to the MBL knowledge base that is based on MusicBrainz. The first subtask consists of the identification (**Recognition step**) and classification (**Typing step**) of named entities. The task is limited to a subset of three MBL ontology types; `Artist`, `Album` and `Song`. For the second subtask, the entities recognized in the first subtask must be linked to the corresponding resources in MBL if existing or to generate a URI for the emerging entity (**D2KB step**). A system has to fulfill both subtasks in order to participate in Task 3. *Task 4* aims at extracting knowledge from a given text and to formalize the knowledge in RDF triples. DBpedia is considered as the knowledge base in this task.

Table 6: Types and subtype and instance examples for Task 2 of the OKE challenge. Table extracted from [24].

Type	Subtypes	Instances
Activity	Game, Sport	Baseball,Chess
Agent	Organisation, Person	Leipzig_University
Award	Decoration, NobelPrize	Humanitas_Prize
Disease		Diabetes_mellitus
EthnicGroup		Javanese_people
Event	Competition, PersonalEvent	Battle_of_Leipzig
Language	ProgrammingLanguage	English_language
MeanOfTransportation	Aircraft, Train	Airbus_A300
PersonFunction	PoliticalFunction	PoliticalFunction
Place	Monument, WineRegion	Beaujolais, Leipzig
Species	Animal, Bacteria	Cat, Cucumibacter
Work	Artwork, Film	Actrius, Debian

OKE was organized in conjunction with the ESWC 2018 conference, where systems participating in the challenge and their results were presented to the public in a dedicated workshop session. The papers describing the systems submitted to the OKE challenge were peer-reviewed by experts and the most promising systems were invited to participate in the challenge.

The OKE challenge papers have been published by Springer on the proceedings volume *Buscaldi, Davide, Gangemi, Aldo, Reforgiato Recupero, Diego (Eds.), Semantic Web Challenges, Communications in Computer and Information Science, vol. 927, 2018*[40]. In particular, in [23] the challenge organizers presented an overview of the challenge results and baseline systems, while in [5] the challenge participants described their system.

6.2 Participating Systems

The challenge attracted five research groups this year. Four groups from universities and one from industry. Two groups finally participated in the OKE challenge and competed in task three. Both systems are briefly described below.

6.3 RelExt

RelExt is an approach based on a deep learning classifier that uses self attention. The classifier was trained on sentences from Wikipedia pages chosen in a distance supervised fashion with the DBPedia knowledge base. RelExt uses a filtering step to find words in sentences that might express specific

[40] https://doi.org/10.1007/978-3-030-00072-1

relations. These words are manually filtered by the authors and were used to refine the sentences to obtain training data.

RelExt participated in task three of the OKE challenge.

6.4 Baseline

The baseline system for task three simply used the annotated documents in the evaluation phase without a learning or training step on the training dataset. The input documents of task three consisted of annotated entities with entity linkings to the DBPedia knowledge base. Thus, the baseline chose pairwise the given URIs of the linked entities from the input documents to create a SPARQL query to request all predicates that hold between two URIs in DBPedia. In case two entities had a statement in the knowledge base with a predicate included in the task, the baseline chose this statement in the response document.

6.5 Evaluation Metrics

The systems participating in the challenge were evaluated using recall, precision and F1-measure. Equation 1, 2 and 3 formalize Precision p_d, Recall r_d and F1-measure used to evaluate the quality of the systems responses for each document $d \in D$. They consist of the number of true positives TP_d, false positives FP_d and false negatives FN_d. We micro and macro average the performances over the documents.[41]

$$p_d = \frac{TP_d}{TP_d + FP_d} \tag{1}$$

$$r_d = \frac{TP_d}{TP_d + FN_d} \tag{2}$$

$$f_d = 2 \cdot \frac{p_d \cdot r_d}{p_d + r_d} \tag{3}$$

6.6 Results

Table 7 depicts the results of the OKE 2018 on task three. The results are available in Hobbit for both participants, RelExt[42] and the Baseline[43]. RelExt won this task with 54.30% Macro F1-Score and 48.01% Micro F1-Score.

6.7 Conclusions

The OKE attracted five research groups from academia and industry coming from the knowledge extraction as well as the SW communities. Indeed, the challenge proposal was aimed at attracting groups from these two communities in order to further investigate existing overlaps between both. Additionally, one of the goals of the challenge was to foster the collaboration between the two communities, to the aim of growing further the SW community. To achieve this goal we defined a SW reference

[41] The macro averages for the performance measures can be retrieved from the official Hobbit SPARQL endpoint at http://db.project-hobbit.eu/sparql.

[42] https://master.project-hobbit.eu/experiments/1529075533385

[43] https://master.project-hobbit.eu/experiments/1527777181515

Table 7: RelExt and Baseline.

KPI	RelExt	Baseline
Avg. ms per Doc	836.26	513.42
Error Count	1	0
Macro F1-Score	**54.30**	8.00
Macro Precision	53.98	10.00
Macro Recall	64.17	7.18
Micro F1-Score	**48.01**	8.66
Micro Precision	39.62	68.75
Micro Recall	60.92	4.62

evaluation framework, which is composed of a) four tasks, b) a training and evaluation dataset for each task, and c) an evaluation framework to measure the performance of the systems. Although the participation in terms of number of competing systems remained quite limited with two, we believe that the challenge is a success in the hybridisation of Semantic Web technologies with knowledge extraction methods.

As a matter of fact, the evaluation framework is available online and can be reused by the community and for next editions of the challenge.

7 Link Discovery Challenge Results Overview

7.1 Definition of Tasks

In the Link Discovery Track two benchmark generators were proposed to deal with *link discovery* for spatial data represented as *trajectories* i.e., sequences of longitude, latitude pairs. The aim of the Link Discovery Track is to test the performance of Link Discovery tools that implement string-based as well as topological approaches for identifying matching spatial entities. The different frameworks were evaluated for both accuracy (precision, recall and f-measure) and time performance.

Two datasets were used. TomTom[44] Data have been employed for the creation of the appropriate benchmarks. Moreover, TomTom datasets contain representations of traces (GPS fixes). Each trace comprises a number of points. Each point has time stamp, longitude, latitude and speed (value and metric). The points are sorted by time stamp of the corresponding GPS fix (ascending). The second dataset is acquired from Spaten [8]. Spaten is an open-source configurable spatio-temporal and textual dataset generator, that can produce large volumes of data based on realistic user behavior.

The Link Discovery Track consists of the following tasks:

- Task 1 (Linking): The first task measures how well the systems can match traces that have been modified using string-based approaches along with addition and deletion of intermediate points.

[44] https://www.tomtom.com/

Since TomTom datasets only contain coordinates, in order to apply string-based modifications implemented in LANCE [20] we have replaced a number of those points with labels retrieved from Linked Data spatial datasets using the Google Maps[45], Foursquare[46] and Nominatim Openstreetmap[47] APIs. This task also contains modifications on date and coordinate formats.

- Task 2 (Spatial): The second task measures how well the systems can identify the DE-9IM (Dimensionally Extended nine-Intersection Model) topological relations. The supported spatial relations are the following: *Equals, Disjoint, Touches, Contains/Within, Covers/CoveredBy, Intersects, Crosses, Overlaps*. The traces are represented in the Well-known text (WKT) format. For each relation, a different pair of source and target datasets is given to the participants.

  For the Linking task TomTom dataset was used while for the Spatial task both datasets were used.

The Link Discovery track was accepted at OAEI OM 2018 Workshop and was organized in conjunction with the ISWC 2018 conference. OM workshop conducted an extensive and rigorous evaluation of ontology matching and instance matching (link discovery) approaches through the OAEI (Ontology Alignment Evaluation Initiative) 2018 campaign.

7.2 Participating Systems

The participating system of the Linking task was the AgreementMakerLight (AML). In essence the AgreementMakerLight (AML) is an automated ontology matching system. AML system is further detailed in [6].

Furthermore, the participating systems to the Spatial task were three: AgreementMakerLight (AML), Rapid Discovery of Topological Relations (RADON) and Silk systems. RADON and Silk systems had already been described in [21] and [22].

In essence RADON performs the discovery of topological relations between geospatial resources according to the DE9-IM standard. Furthermore, Silk is a framework for Spatial and Temporal Link Discovery.

7.3 Results & Achievements

7.3.1 Task 1 (Linking)

The test cases implemented in the benchmark focus on string-based transformations with different (a) levels (b) types of spatial object representations and (c) types of date representations. Furthermore, the benchmark supports addition and deletion of ontology (schema) properties, known also as schema transformations. The datasets that implement those test cases can be used by Instance Matching tools to identify matching entities. In a nutshell, the benchmark can be used to check whether two traces with their points annotated with place names designate the same trajectory.

In order to evaluate the systems we built a ground truth containing the set of expected links where an instance s_1 in the source dataset is associated with an instance t_1 in the target dataset that has been generated as a modified description of s_1. The only participant was AgreementMakerLight (AML) which

[45] https://developers.google.com/maps/
[46] https://developer.foursquare.com/
[47] http://nominatim.openstreetmap.org/

was judged on the basis of *precision*, *recall*, *F-measure* and *runtime* results that are shown in Tables 8 and 9. AML returned high precision and recall capturing all the correct links. AML completes the task with perfect results.

Table 8: HOBBIT Link Discovery Linking Task (Sandbox 100 instances)

Sandbox task

	Precision	**Recall**	**F-measure**	**Run Time**
AML	1.000	1.000	1.000	6838

Table 9: HOBBIT Link Discovery Linking Task (Mainbox 5000 instances)

Mainbox task

	Precision	**Recall**	**F-measure**	**Run Time**
AML	1.000	1.000	1.000	313039

Datasets, reference alignments, and task results are available on the HOBBIT website: https://project-hobbit.eu/challenges/om2018/.

7.3.2 Task 2 (Spatial)

Regarding Task 2, the test cases were divided into four tasks. In the first two tasks (SLL and LLL), the systems were asked to match LineStrings to LineStrings considering a given relation for 200 and 2K instances for the TomTom and Spaten datasets. In the last two second tasks (SLP, LLP), the systems were asked to match LineStrings to Polygons (or Polygons to LineStrings depending on the relation) again for both datasets.

This benchmark generator implements all topological relations of DE-9IM between trajectories in the two dimensional space. To the best of our knowledge such a generic benchmark, that takes as input trajectories and checks the performance of linking systems for spatial data does not exist. For the design, we focused on (a) on the correct implementation of all the topological relations of the DE-9IM topological model and (b) on producing large datasets large enough to stress the systems under test. The supported relations are: *E*quals, Disjoint, Touches, Contains/Within, Covers/CoveredBy, Intersects, Crosses, Overlaps.

Three systems participated in this task, specifically, AgreementMakerLight (AML), Silk and RADON.The systems were judged on the basis of time performance as precision, recall and F-measure were equal to 1.0 for all systems. The results are shown in Figures 6, 7, 8 and 9.

Taking into account the executed test cases we can identify the capabilities of the tested systems as well as suggest some improvements. All the systems participated in most of the test cases. RADON is the only system that addressed all the tasks, while it can be improved for the *Touches* and *Intersects* relations for the Tasks SLL and LLL and it also has the best performance for the SLP and LLP tasks. AML performs extremely well in most cases.

In general, all systems needed more time to match the TomTom dataset than the Spaten one, due to the smaller number of points per instance in the latter. Comparing the LineString/LineString to the LineString/Polygon Tasks we can say that all the systems needed less time for the first in *Contains*,

Within, *Covers* and *Covered by* relations, more time for the *Touches*, *Instersects* and *Crosses* relations, and approximately the same time for the *Disjoint* relation.

Datasets, reference alignments, and task results are available on the HOBBIT website: https://project-hobbit.eu/challenges/om2018/.

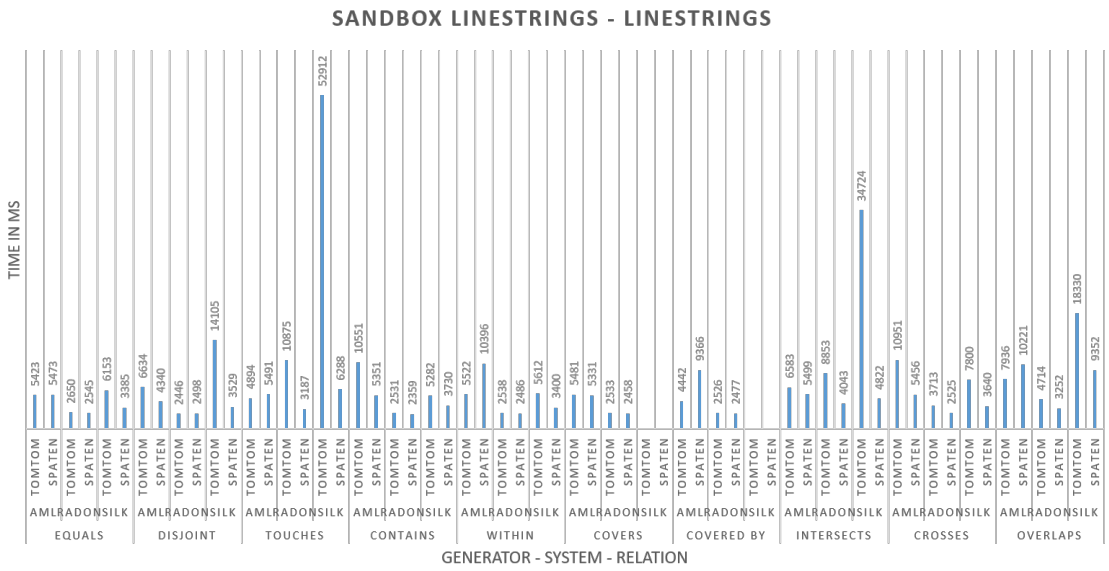


Figure 6: HOBBIT Link Discovery Spatial Task (Sandbox Linestrings - Linestrings)

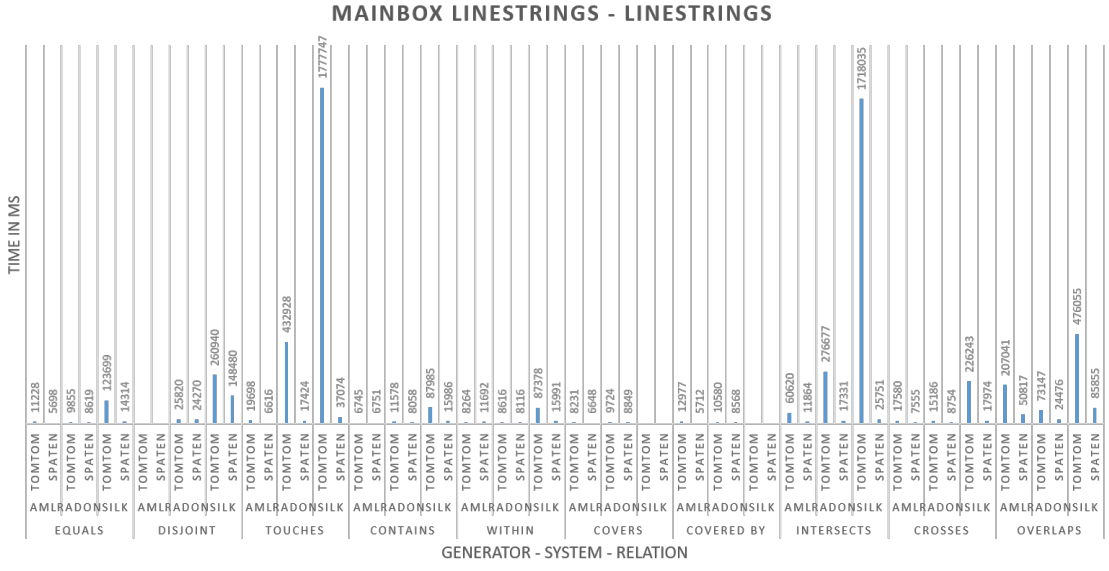


Figure 7: HOBBIT Link Discovery Spatial Task (Mainbox Linestrings - Linestrings)

7.4 Conclusions

The goal of the Link Discovery Track was to test the performance of Link Discovery tools that implement string-based as well as topological approaches for identifying matching spatial entities. We

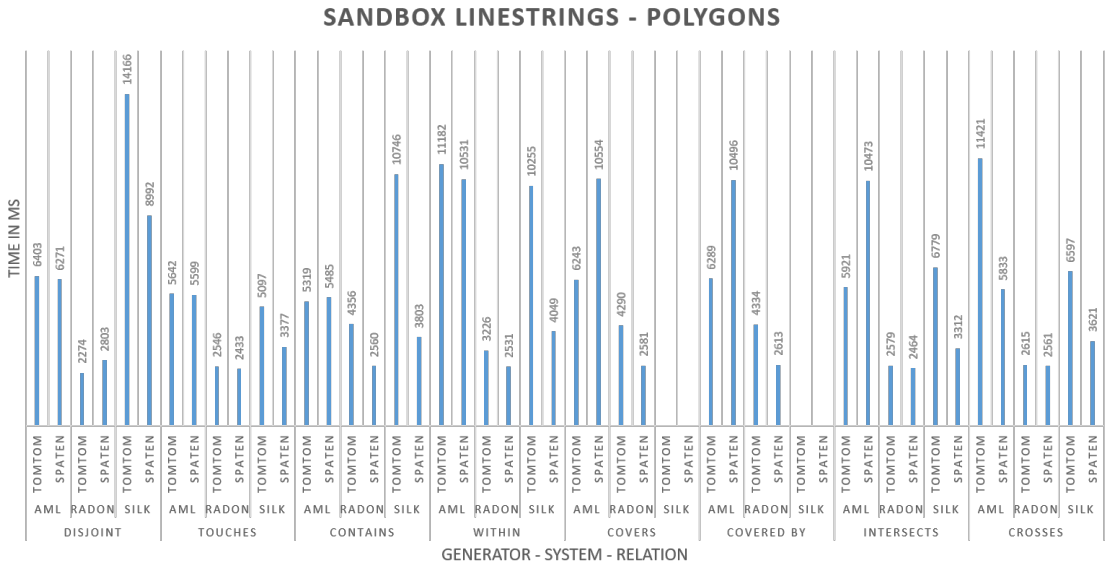


Figure 8: HOBBIT Link Discovery Spatial Task (Sandbox Linestrings - Polygons)

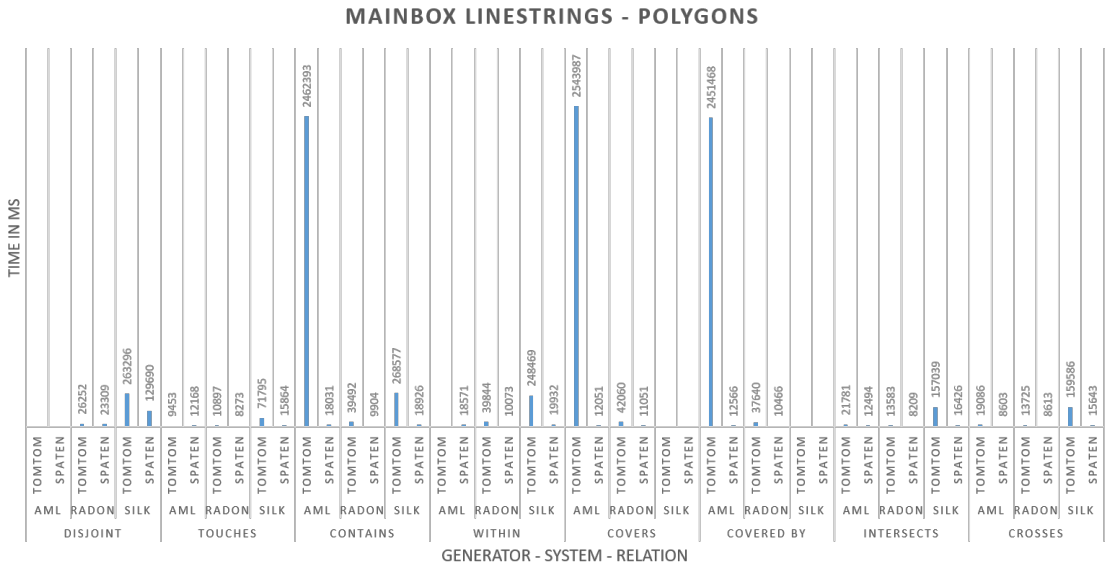


Figure 9: HOBBIT Link Discovery Spatial Task (Mainbox Linestrings - Polygons)

benchmarked and evaluated three systems and the overall results indicated that despite the fact that the systems present noteworthy performance, there is still room for further improvement.

8 DEBS Grand Challenge Results Overview

8.1 Definition of Tasks

The DEBS Grand Challenge (GC) is a series of annual challenges joint to the annual ACM International Conference on Distributed and Event-Based Systems (DEBS) which targets the evaluation of

event-based systems for real-time analytics in different domains.

The 2018 DEBS GC [48] focused on the application of machine learning to spatio-temporal streaming data. The goal of the challenge was to make the naval transportation industry more reliable by providing predictions for vessels' destinations and arrival times. Predicting both correct destinations and arrival times of vessels are relevant problems, that once solved, can boost the efficiency of the overall supply chain management.

The 2018 DEBS Grand Challenge has been co-organized by Marine Traffic, the Big Data Ocean project and the HOBBIT (https://project-hobbit.eu/) project represented by AGT International (http://www.agtinternational.com/). The Grand Challenge data was provided by the Marine Traffic and hosted by the Big Data Ocean while the automated evaluation platform was provided by the HOBBIT project.

Two queries were considered by the DEBS 2018 GC for the prediction of vessel's destination and arrival time. In more detail, query (1) "Predicting the correct destination of a vessel" is a relevant problem for a wide range of stakeholders including port authorities, vessel operators and many more. The prediction problem is to generate a continuous stream of predictions for the destination port of any vessel given the following information: i) unique ID of the ship, ii) actual position of the ship, iii) name of the port of departure, iv) time stamp, and v) vessel's draught. The above data was provided as a continuous stream of tuples and the goal of the system was to provide for every input tuple one output tuple containing the name of the destination port. A solution is considered correct at time stamp T if for a tuple with this timestamp as well as for all subsequent tuples the predicted destination port matches the actual destination port. The goal of any solution was not only to predict a correct destination port but also to predict it as soon as possible counting from the moment when a new port of origin appears for a given vessel. After port departure and until arrival, the solution had to emit one prediction per position update.

With regards to query (2) "Predicting arrival times of ships", there was a set of ports defined by respective bounding boxes of coordinates. Once a ship leaves a port (i.e. the respective bounding box), the task was to predict the arrival time at its destination port (i.e. when the next defined bounding boxes is entered). Also for this query, after port departure and until arrival, the solution had to emit one prediction per position update.

The project HOBBIT provided the platform for the evaluation of the systems submitted to DEBS GC 2018. The challenge papers have been published by ACM as part of the DEBS 2018 conference proceedings volume, *DEBS '18: Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems, ACM, New York, NY, USA, 2018*[49]. Specifically, in [1] the challenge organizers presented an overview of the challenge, while in [2, 3, 4, 12, 13, 14, 15, 17, 19] the challenge participants described their systems.

8.2 Participating Systems

Nine systems participated in DEBS Grand Challenge, addressing both scenarios.

The team from **Hubert Curien Laboratory (Jean Monnet University, France)** presented a solution, which employs an efficient multi-dimensional index to store the training and historical dataset [13]. For upcoming events the indexing structure determines the closest points of interests. Authors describe pros and cons of traditional techniques like All-Korder-Markov, Probabilistic Suffix

[48] https://project-hobbit.eu/challenges/debs2018-grand-challenge/
[49] https://dl.acm.org/citation.cfm?id=3210284

Tree, Naive Bayes, Support Vector Machines and conclude that such techniques cannot be generalized and could not be used for prediction (because of forgetting older events). To tackle these problems the authors designed the optimized storage framework, which employs space-fitting curve in order to reduce the number of dimensions and use linear data structure (Ph-tree). The described solution got the second place in the DEBS 2018 challenge.

The team from **Chungnam National University (Daejeon, Republic of Korea)** presented a solution based on a "likelihood" measure calculated using Bayesian inference and grid-based heuristics [12]. Authors decided not to use deep-learning technologies like neural networks, which require a huge amounts of data. Another idea of the solution is to use equal-sized square grid cells to divide the problem area based in minimum bounding rectangles (from longitudes and latitudes) obtained from the training data.

The team from **Alexandru Ioan Cuza University of Iasi (Romania)** presented two different solutions. First solution employs a cell grid architecture using a sequence of use-case-specific hash tables [2]. Authors decided split the data according to a cell grid and perform training/prediction based on a per cell principle. Sequences of hash tables were maintained per each cell, where training data of this particular cell have been stored. Each new incoming event was matched against the structures in the cell and the best candidate should be returned as a prediction. Then optional robustness test was performed to check whether the best candidate affects the robustness of the prediction stream. The solution demonstrated the most accurate prediction results and won the first place.

Second solution from **Alexandru Ioan Cuza University of Iasi (Romania)** team was based on nearest neighbour search principle [19]. For training the solution divides dataset into routes between ports, data-points of each route then partitioned by an arrival port. Then for each arrival port the solution constructs a Ball Tree structure. Authors choose a Ball Trees because they can work with any number of dimensions. The exact amount of dimensions for the particular use-case was empirically found by authors, and some values have been optimized using genetic algorithm. For all the points in the route a bearing angle between current and previous point is calculated. For all the Ball Trees the closest Route Point is searched and the similarity function (different for each type of query) was applied to find out the best (closest) candidate. Authors compared the precision of Ball Trees with KD-trees and got 41% better results for Ball Trees. Adding multithreading and parallel streams the runtime performance was improved to 40%.

The team from **University of Illinois Urbana-Champaign (USA)** proposed a MtDetector - a solution based on Deep Neural Network (DNN) [14]. For each port a solution builds a DNN and infers the arrival port. The authors observed that ships departing from the same port arrive only at a specific subset of ports and then decided to reduce solution space for each ship. The authors proposed an incremental majority filter algorithm to reduce noise, which lead to increased accuracy from 30 to 90%. As a features for DNN the authors used standard ones (timestamp, ship type, speed, longitude, latitude, course, heading) as well as custom ones: bearing, cumulative distance, cumulative time. Last two ones are total moving distance and total travel time from departure to its current position/timestamp respectively. The authors state that the last two features are contributing a lot to a final accuracy value. The solution is implemented on top of DtCraft distributed execution engine with task-based parallelism.

The team from **Dresden University of Technology (Germany)** proposed their solution based on ensemble of models [4]. Tasks of the challenge are considered to be a classification task and regression tasks. To capture complex non-linearity of trajectory, keeping the state of high-level behavior authors applied an ensemble of methods: Random Forests, Gradient Boosting Decision Trees (GBDT) and Extremely Randomized Trees (ERT), deep learning model with Long Short-Term Memory (LSTM) cells. The authors conclude that LSTM model works only for short trips, but without complex tra-

jectories as well as it requires large preprocessing and splitting of the initially provided data set. The authors conclude that only time features can capture an exact behavior on each segment of the trip. The authors applied feature engineering on time-based data to obtain travel times needed to finish trip from a given timestamp: trips were splitted on weekly, daily and hourly intervals. The authors conclude that the most useful feature was a pre-computed time delta for each ship from its current position to reach its final destination. The feature was included into the final prediction model.

The team from **University of Carthage (Tunisia)** proposed a solution based on GeoHashing, two indexes and pairwise sequence alignment to score similarity of two geohash sequences with queen-spatial neighborhood [15]. Using the training data trip patterns have been calculated using 5-uplet (vessel-id, ship-type, departure-port-name, arrival-port, arrival-timestamp) as a key and 6-uplet: (timestamp, latitude, longitude, course, heading, draught) as a value. The authors observed that there might be different trip patterns for the same pair (departure port, arrival port) as well as zigzag patterns between neighboring geohashes. To calculate the similarity between geohash sequences the authors check if elements are within a matching distance from each other and their neighbors. The solution is implemented using Apache Spark - a scalable open source engine for data processing.

The team from **Israel Institute of Technology (Haifa, Israel)** proposes a Venelia system which applies several machine learning techniques, including predictive models such as Markov chains and regression trees, as well as queuing theory [3]. The authors state that different type of ships have different behavioural models and different prediction models should be applied based on a vessel subtypes (introduced by the authors). For prediction Venelia uses three types of Markov chains: the port Markov chain, the way-point Markov chain, the path Markov chain. Venelia trains multiple instances of this Markov chains triplet: a global instance, per vessel type instances, and per vessel sub-types instances. To select the destination port among the three predictions a weighted voting algorithm is applied, where prediction confidence and model relevance are considered. To predict arrival times Venelia uses snapshot principle and gradient tree boosting regression model, which features vessel type, sub-type and speeds. Technically Venelia is implemented on AKKA framework and implements a number of well-know stream-balancing algorithms.

The team from **Insight Centre for Data Analytics (Ireland)** proposed a solution based on sequence-to-sequence model with 2 Long Short-Term Memory (LSTM) networks inside [17]. Authors applied a spacial grid principle and transformed geo-bound moving of ships to movings between encoded cells. The solution is more focused on the transition of a vessel between cells in a spatial grid rather than time intervals between observations of vessel coordinates. The authors train neural networks (encoder and decoder) using Stochastic gradient descent (SGD) optimization algorithm. The authors state that they also tried a bunch of approaches (Gated recurrent unit, Google's NMT, bidirectional (BRNN), deep bidirectional (DBRNN), pyramidal deep bidirectional (PDBRNN), convolutional (CNN)) and conclude that seq2seq method with LSTM neural network is an effective solution for predicting vessel trajectory. Authors made a youtube screencast demonstrating their solution[50].

8.3 Results & Achievements

The DEBS 2018 Grand Challenge consists of 2 independent tasks, each task running the second version Structured Machine Learning (DEBS GC 2018 benchmark) using the test dataset. Both tasks use the same dataset (330,000 datapoint), but executed with different measurable criteria.

For the task 1, the query is to predict a destination port names and main measurable criteria is earlyness rate, i.e. how early a benchmarked system would start to predict correct port names until

[50]Youtube screencast:https://www.youtube.com/watch?v=JrSIU_Y9SOo

Table 10: Results for Query 1 (prediction of arrival port names).

Team	Earlyness rate	A	Working time (sec)	B	Total Q1
University of Iasi	0.685	1	99	2	1.25
University of Illinois	0.672	2	86	1	1.75
Jean Monnet University	0.668	3	149	5	3.5
Chungnam National University	0.653	4	102	3	3.75
University of Iasi (2nd)	0.647	5	157	6	5.25
Israel Institute of Technology	0.5	6	129	4	5.5
Dresden University of Technology	-	-	-	-	-
Insight Centre	-	-	-	-	-
University of Carthage	-	-	-	-	-

the end of the trip. The higher value gives the higher rank A for the system. For the task 2, the query is to predict arrival time and measurable criteria is mean absolute error (in minutes). The lower value gives the higher rank A for the system.

The performance results of the tasks execution within the DEBS 2018 Grand Challenge are presented in the tables 10, 11. Participating teams are sorted by total score of their solutions, which had a major weight in the evaluation criteria. The score (place) of the system in each task was calculated by the formula $0.75 * A + 0.25 * B$. Missing results means that system was not able to finish the final benchmark in a given timeout (2400 seconds).

The final leaderboard of the teams are presented in Table 12. Rank (place) of systems is calculated as a summary of ranks (places), which solution in both tasks. From the tables it can be seen, that

Table 11: Scores for Query 2 (prediction of arrival times).

Team	Mean Absolute Error (min.)	A	Working time (sec)	B	Total Q2
University of Iasi	959.839	1	100	2	1.25
Jean Monnet University	1099	2	145	4	2.5
Chungnam National University	1251.15	3	100	2	2.75
Israel Institute of Technology	1493.18	4	133	3	3.75
University of Illinois	5425.53	5	86	1	4.75
University of Iasi (2nd)	1705.35	6	164	5	5
Dresden University of Technology	-	-	-	-	-
Insight Centre	-	-	-	-	-
University of Carthage	-	-	-	-	-

Table 12: Overall scores.

Team	Q1	Q2	Total Score
University of Iasi	1.25	1.25	2.5
Jean Monnet University	3.5	2.5	6
Chungnam National University	3.75	2.75	6.5
University of Illinois	1.75	4.75	6.5
Israel Institute of Technology	5.5	3.75	9.25
University of Iasi (2nd)	5.25	5	10.25
Dresden University of Technology	-	-	-
Insight Centre	-	-	-
University of Carthage	-	-	-

solution provided by the University of Iasi demonstrated best results in both tasks. The team won the main award of the DEBS 2018 Grand Challenge [10].

9 Conclusions

The HOBBIT project has successfully organized five challenges. The MOCHA, OKE and SQA challenges were organized in conjunction with the ESWC 2018 conference. Also, HOBBIT was responsible for the 2018 DEBS Grand Challenge that annually runs as part of the DEBS conference, as well as the Link Discovery Task at the 2018 OAEI campaign which was held under the Ontology Matching workshop at the ISWC 2018 conference. Participating systems were evaluated using the eight HOBBIT benchmarks and platform.

In the second series of HOBBIT challenges we took into consideration the feedback received from the first round and we improved the procedures and technologies needed to integrate and execute a system on the platform. Moreover, we also updated the platform's documentation with more information and the corresponding instructions provided on the challenges' websites on how to submit and test systems. Therefore, the challenges were an extensive test of the benchmarking platform, and the fair evaluation of Big (Linked) Data processing technologies was received very positively by the community.

References

[1] *DEBS '18: Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, New York, NY, USA, 2018. ACM.

[2] Ciprian Amariei, Paul Diac, Emanuel Onica, and Valentin Roşca. Cell grid architecture for maritime route prediction on ais data streams. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 202–204, New York, NY, USA, 2018. ACM.

[3] Moti Bachar, Gal Elimelech, Itai Gat, Gil Sobol, Nicolo Rivetti, and Avigdor Gal. Venilia, online learning and prediction of vessel destination. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 209–212, New York, NY, USA, 2018. ACM.

[4] Oleh Bodunov, Florian Schmidt, André Martin, Andrey Brito, and Christof Fetzer. Real-time destination and eta prediction for maritime traffic. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 198–201, New York, NY, USA, 2018. ACM.

[5] Héctor Cerezo-Costas and Manuela Martín-Vicente. Relation extraction for knowledge base completion: A supervised approach. In *Semantic Web Evaluation Challenge*, pages 52–66. Springer, 2018.

[6] Vivek Shivaprabhu Isabela Mott Catia Pesquita Francisco Couto Isabel Cruz Daniel Faria, Booma S. Balasubramani. Results of AML in OAEI 2017. http://www.dit.unitn.it/~pavel/om2017/papers/oaei17_paper2.pdf, 2017.

[7] Dennis Diefenbach, Kamal Singh, and Pierre Maret. On the scalability of the qa system wdaqua-core1. In *Semantic Web Evaluation Challenge*, pages 76–81. Springer, 2018.

[8] Thaleia Dimitra Doudali, Ioannis Konstantinou, and Nectarios Koziris. Spaten: A spatio-temporal and textual big data generator. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017*, pages 3416–3421, 2017.

[9] Kleanthi Georgala, Mirko Spasić, Milos Jovanovik, Vassilis Papakonstantinou, Claus Stadler, Michael Röder, and Axel-Cyrille Ngonga Ngomo. *MOCHA2018: The Mighty Storage Challenge at ESWC 2018*. Springer International Publishing, 2018.

[10] Vincenzo Gulisano, Zbigniew Jerzak, Pavel Smirnov, Martin Strohbach, Holger Ziekow, and Dimitris Zissis. The debs 2018 grand challenge. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 191–194, New York, NY, USA, 2018. ACM.

[11] Milos Jovanovik and Mirko Spasić. Benchmarking virtuoso 8 at the mighty storage challenge 2018: Challenge results. In *Semantic Web Evaluation Challenge*, pages 24–35. Springer, 2018.

[12] Hyungkun Jung, Kang-Woo Lee, Joong-Hyun Choi, and Eun-Sun Cho. Bayesian estimation of vessel destination and arrival times. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 195–197, New York, NY, USA, 2018. ACM.

[13] Abderrahmen Kammoun, Tanguy Raynaud, Syed Gillani, Kamal Singh, Jacques Fayolle, and Frederique Laforest. A scalable framework for accelerating situation prediction over spatio-temporal event streams. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 221–223, New York, NY, USA, 2018. ACM.

[14] Chun-Xun Lin, Tsung-Wei Huang, Guannan Guo, and Martin D. F. Wong. Mtdetector: A high-performance marine traffic detector at stream scale. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 205–208, New York, NY, USA, 2018. ACM.

[15] Rim Moussa. Scalable maritime traffic map inference and real-time prediction of vessels' future locations on apache spark. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 213–216, New York, NY, USA, 2018. ACM.

[16] Giulio Napolitano, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. The scalable question answering over linked data (sqa) challenge 2018. In *Semantic Web Evaluation Challenge*, pages 69–75. Springer, 2018.

[17] Duc-Duy Nguyen, Chan Le Van, and Muhammad Intizar Ali. Vessel destination and arrival time prediction with sequence-to-sequence models over spatial grid. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 217–220, New York, NY, USA, 2018. ACM.

[18] Nikolay Radoev, Mathieu Tremblay, Amal Zouaq, and Michel Gagnon. Lama: a language adaptive method for question answering. *Scalable Question Answering over Linked Data Challenge (SQA2018), Heraklion, Greece*, 2018.

[19] Valentin Roşca, Emanuel Onica, Paul Diac, and Ciprian Amariei. Predicting destinations by nearest neighbor search on training vessel routes. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, DEBS '18, pages 224–225, New York, NY, USA, 2018. ACM.

[20] Tzanina Saveta, Evangelia Daskalaki, Giorgos Flouris, Irini Fundulaki, and Axel-Cyrille Ngonga Ngomo. Lance: A generic benchmark generator for linked data. In *International Semantic Web Conference (Posters & Demos)*, 2015.

[21] Mohamed Ahmed Sherif, Kevin Dreßler, Panayiotis Smeros, and Axel-Cyrille Ngonga Ngomo. Radon-rapid discovery of topological relations. In *AAAI*, pages 175–181, 2017.

[22] Panayiotis Smeros and Manolis Koubarakis. Discovering spatial and temporal links among rdf data. In *LDOW@ WWW*, 2016.

[23] René Speck, Michael Röder, Felix Conrads, Hyndavi Rebba, Catherine Camilla Romiyo, Gurudevi Salakki, Rutuja Suryawanshi, Danish Ahmed, Nikit Srivastava, Mohit Mahajan, et al. Open knowledge extraction challenge. *Semantic Web Challenges: 5th SemWebEval Challenge at ESWC 2018, Heraklion, Greece, June 3–7, 2018, Revised Selected Papers*, page 39.

[24] René Speck, Michael Röder, Sergio Oramas, Luis Espinosa-Anke, and Axel-Cyrille Ngonga Ngomo. *Open Knowledge Extraction Challenge 2017*, pages 35–48. Springer International Publishing, Cham, 2017.

[25] Ruben Taelman, Miel Vander Sande, and Ruben Verborgh. Versioned querying with ostrich and comunica in mocha 2018. In *Semantic Web Evaluation Challenge*, pages 17–23. Springer, 2018.

[26] Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, Paolo Ferragina, Christiane Lemke, Andrea Moro, Roberto Navigli, Francesco Piccinno, Giuseppe Rizzo, Harald Sack, René Speck, Raphaël Troncy, Jörg Waitelonis, and Lars Wesemann. GERBIL – General entity annotation benchmark framework. In *24th International Conference on World Wide Web*, pages 1133–1143, 2015.

[27] Elizaveta Zimina, Jyrki Nummenmaa, Kalervo Järvelin, Jaakko Peltonen, Kostas Stefanidis, and Heikki Hyyrö. Gqa: grammatical question answering for rdf data. In *Semantic Web Evaluation Challenge*, pages 82–97. Springer, 2018.